

### EE-432

Computer Networks LAB MANUAL

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 1 Wireshark: A Network Protocol Analyzer

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
| Hafiz Muhammad Bilal | 2022\_EE\_014 |  |  |  |

Checked on:

Signature:

### Introduction to Wireshark

 **Objectives**

This lab will enable the students to achieve the following:

* Familiarize themselves with the Wireshark environment
* Learn how to capture packets of network tranffic
* Browse interactively the traffic running on a computer network

 **Instructions**

1. Read manual carefully before starting lab.
2. All exercises are individual exercises
3. You are supposed to provide the answers to the questions listed at the end of this manual in text. Paste screenshots/images in the textboxes where required. You will be required to submit your complete manual on Google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. You must aim to complete the lab half an hour before the lab time ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

 **Background**

A protocol analyzer is a tool that can be used to inspect what exactly is happening on a network with respect to traffic flow. For example, if your TCP/IP sessions are "hanging", a protocol analyzer can show which system sent the last packet, and which system failed to respond. If you are experiencing slow screen updates, a protocol analyzer can display delta time stamps and show which system is waiting for packets, and which system is slow to respond.

A protocol analyzer can show runaway traffic (broadcast or multicast storms) and its origin, system errors and retries, and whether a station is sending, trying to send, or only seeming to communicate. You will get information that is otherwise unavailable, which results in more efficient troubleshooting and better LAN health.

* + 1. **Computer network**

A computer network, often simply referred to as a network, is a collection of hardware components and computers interconnected by communication channels that allow sharing of resources and information. In the world of computers, networking is the practice of linking two or more computing devices together for the purpose of sharing data. In networking, the communication language used by computer devices is called the protocol. Yet another way to classify computer networks is by the set of protocols they support. Networks often implement multiple protocols to support specific applications.

* + 1. **What is a protocol analyzer?**

Protocol analyzers capture conversations between two or more systems or devices. A protocol analyzer not only captures the traffic, it also decodes (interprets) the traffic. Decoding allows you to view the conversation in English, as opposed to binary language. A sophisticated protocol analyzer will also provide statistics and trend information on the captured traffic. Protocol analyzers provide information about the traffic flow on your local area network (LAN), from which you can view device-specific information.

* + 1. **Introduction to Wireshark**

Wireshark is a free and open-source packet analyzer, used for network troubleshooting, analysis, software and communications protocol development, and education.

The basic tool for observing the messages exchanged between executing protocol entities is called a packet sniffer. As the name suggests, a packet sniffer captures (“sniffs”) messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/ received from/by application and protocols executing on your machine.

[Figure 1.1](#_zhzr1llor40j) shows the structure of a packet sniffer. At the right of [Figure 1.1](#_zhzr1llor40j) are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in [Figure 1.1](#_zhzr1llor40j) is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. Recall from the discussion from Section 1.5 in the textbook (Figure 1.20) that messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In [Figure 1.1,](#_zhzr1llor40j) the assumed physical media is an Ethernet, and so all upper layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.

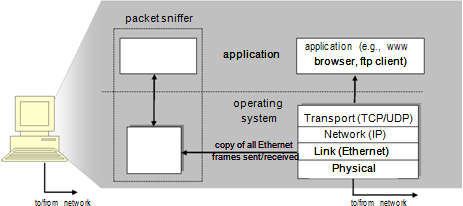


Figure 1.1: Packet sniffer structure

The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must “understand” the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in [Figure 1.1.](#_zhzr1llor40j) The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string “GET,” “POST,” or “HEAD,” as shown in Figure 2.8 in the textbook.

We will be using the Wireshark packet sniffer [<http://www.wireshark.org/>] for these labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers. It’s an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support that includes a user-guide (<http://www.wireshark.org/docs/wsug_html_chunked/>), man pages (<http://www.wireshark.org/docs/man-pages/>), and a detailed FAQ (<http://www.wireshark.org/faq.html>), rich functionality that includes the capability to analyze more than 500 protocols, and a well-designed user interface. It operates in computers using Ethernet, Token-Ring, FDDI, serial (PPP and SLIP), 802.11 wireless LANs and ATM connections (if the OS on which it’s running allows Wireshark to do so).

* + - 1. **Getting Wireshark**

In order to run Wireshark, you will need to have access to a computer that supports both Wireshark and the libpcap or WinPCap packet capture library. The libpcap software will be installed for you alongside Wireshark automatically. See <http://www.wireshark.org/download.html> for a list of supported operating systems and download sites

Download and install the Wireshark software:

* + - * + Go to <http://www.wireshark.org/download.html> and download and install the stable release Wireshark 3.0.3 binary for your computer. Wireshark can be installed on both Windows and Linux. See the documentation page of Wireshark for more details.
        + Download the Wireshark user guide.

The Wireshark FAQ has a number of helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.

* + - 1. **Running on Windows**

On Windows, you should be able be able to find the link by clicking on the Start option of the Windows taskbar and thereby finding the wireshark program in All Programs.

On Linux machines, wireshark can be run by typing “wireshark” at the command prompt (in case there is a problem with your path, type /usr/bin/wireshark which is where wireshark is typically installed).

When you run the Wireshark program, the Wireshark graphical user interface shown in [Figure 1.2](#_tnntc3wvm3y5) will be displayed. Initially, no data will be displayed in the various windows.

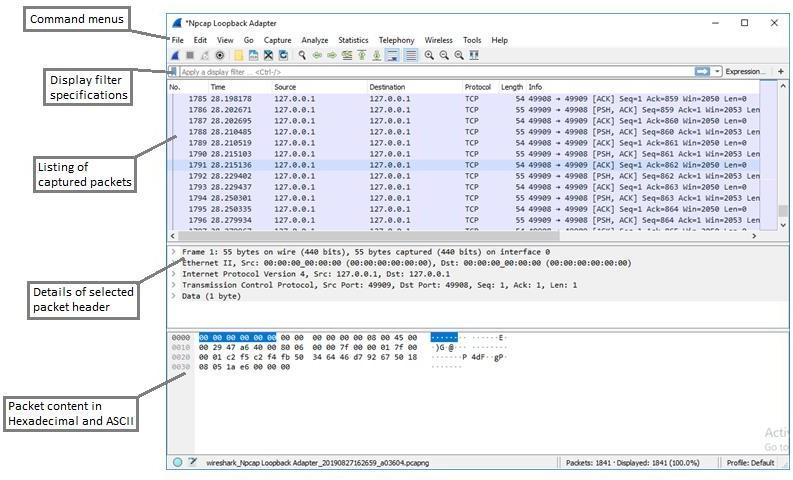


Figure 1.2: Wireshark GUI

The Wireshark interface has five major components:

1. The **command menus** are standard pull down menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.
2. The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol’s header), the time at which the packet was captured, the packet’s source and destination addresses, the protocol type, and protocol-specific information

contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

1. The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet’s one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right-pointing or down-pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.in Wireshark
2. The **packet-contents** window displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
3. Towards the top of the Wireshark graphical user interface, is the **packet display filter field**, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we’ll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

 **Lab Procedure**

The best way to learn about any new piece of software is to try it out! We’ll assume that your computer is connected to the Internet via a wired Ethernet interface. Do the following:

1. **Start up your favorite web browser**, which will display your selected homepage.
2. **Start up the Wireshark software**. You will initially see a window similar to that shown in [Figure 1.2](#_tnntc3wvm3y5) except that no packet data will be displayed in the packet-listing, packet-header, or packet-contents window, since Wireshark has not yet begun capturing packets..
3. **To begin packet capture**, select the Capture pull down menu and select Options. This will cause the “Wireshark·Capture Interfaces” window to be displayed, as shown in [Figure 1.3.](#_dqnyyhw92v0)

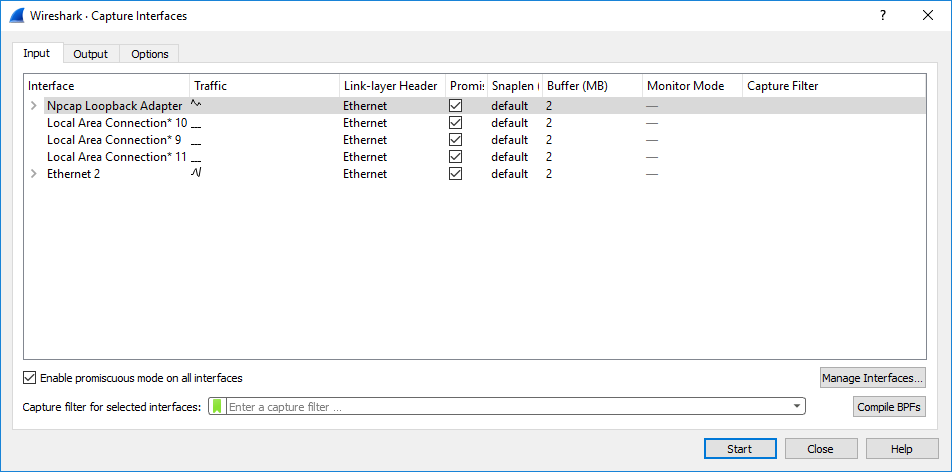


Figure 1.3: Wireshark capture interfaces

1. **Selecting the network interface on which packets would be captured:** Under the “Options” tab, you can use most of the default values, but check “Show capture information during live capture” under Display Options. The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the “Input” tab inside the “Interface” panel. In case your computer has more than one active network interface (e.g., if you have both a

wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets (mostly likely the wired interface). After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin – all packets being sent/received from/by your computer are now being captured by Wireshark!

1. Once you begin packet capture, a packet capture summary window will appear, as shown in [Figure 1.4.](#_gky33wjbhfsh) This window summarizes the number of packets of various types that are being captured, and (importantly!) contains the Stop button that will allow you to stop packet capture. Don’t stop packet capture yet.

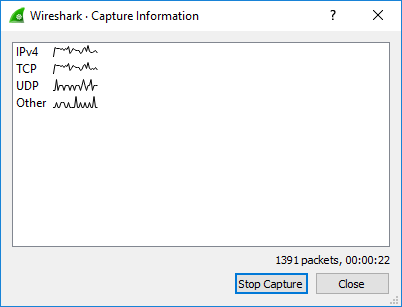


Figure 1.4: Wireshark capture information

1. **Capturing an HTTP interaction on Wireshark:** While Wireshark is running, enter the URL: <http://uet.edu.pk/> and have that page displayed in your browser. In order to display this page, your browser will contact the HTTP server at [http://uet.edu.pk,](http://uet.edu.pk/) and exchange HTTP messages with the server in order to download this page, as discussed in section

2.2 of the text. The Ethernet frames containing these HTTP messages will be captured by Wireshark.

1. **Stopping the capture and inspecting captured packets:** After your browser has displayed the page, stop Wireshark packet capture by clicking “Stop Capture” in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. The main Wireshark window should now look similar to [Figure 1.2.](#_tnntc3wvm3y5) You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the uet.edu.pk web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the Protocol column in [Figure](#_tnntc3wvm3y5) [1.2](#_tnntc3wvm3y5)). Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user. We’ll learn much more about these protocols as we progress through the text! For now, you should just be aware that there is often much more going on than “meets the eye”. *Note: You can answer question 1 of the “Questions” section now.*
2. **Filtering:** Type in “http” (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select Apply (to the right of where you entered “http”). This will cause only HTTP message to be displayed in the packet-listing window.
3. **Details of a packet:** Select the first http message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer to the uet.edu.pk HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window. By clicking on right-pointing and down-pointing arrows heads to the left side of the packet details window, minimize the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. Maximize the amount information displayed about the HTTP protocol. Your Wireshark display should now look roughly as shown in [Figure 1.5.](#_net1v2abm6ha) (Note, in particular, the minimized amount of protocol information for all protocols except HTTP, and the maximized amount of protocol information for HTTP in the packet-header window).

*Note: You can answer questions 2 and 3 of the “Questions” section now.*

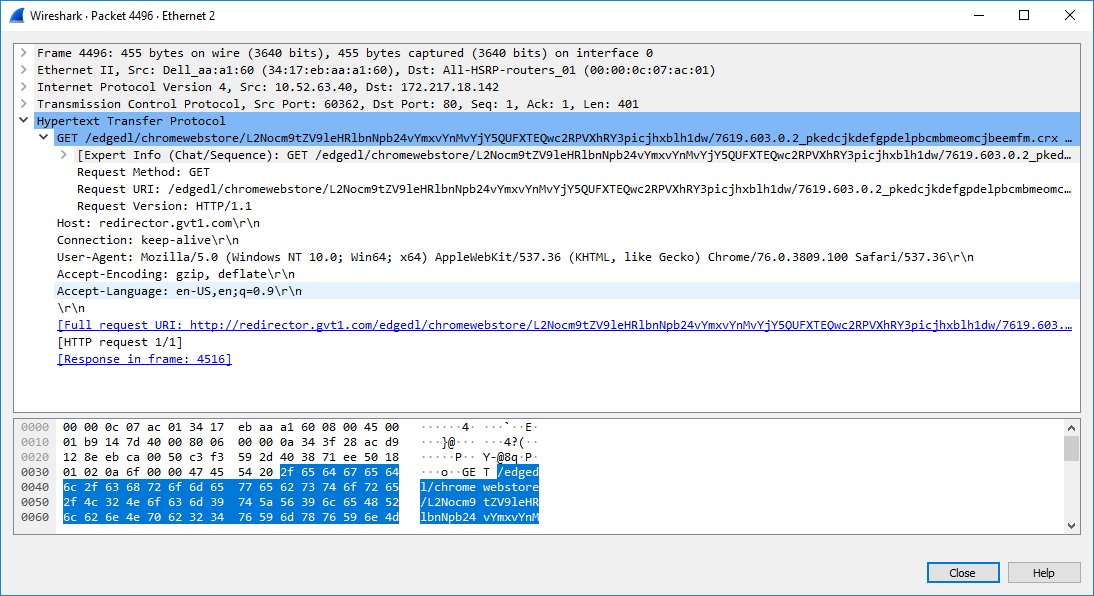


Figure 1.5: Wireshark display after step 9

1. **Statistics of packet captured:** Click on the “Statistics” option on the upper toolbar of Wireshark to explore the various ways in which statistics may be obtained about network traffic.

Explore specifically the ‘Conversation’ options in ‘Statistics’ option on the upper toolbar of Wireshark. We shall be using it to track a conversation of an HTTP flow in future labs.

*Note: You can answer question 5 of the “Questions” section now.*

1. **Obtaining credit for this lab:** Now, please proceed to the questions section to answer the questions. You must note down your answers in this file itself. Please note that every student must upload this file (after duly filling in the answers) to classroom. Please clarify with your instructor/ lab engineer if you have any queries.

 **Questions**

1. List the different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above.

**[3 marks]**

1. **Finding IP address of your machine in Wireshark:** What is the Internet address (or the IP address) of the ‘uet.edu.pk’? What is the Internet address of your computer? How did you find in Wireshark?

**[2 marks]**

1. **Finding IP address of your machine without Wireshark:** Note the IP address of your machine manually by typing ipconfig on the DOS prompt or by typing the command ifconfig on linux machines. Is the IP address of your machine the same as noted in question 2?

**[2 marks]**

1. What is the port number used by the HTTP server “uet.edu.pk”. How did you note in Wireshark?

**[2 marks]**

1. **Delay between request and reply:** How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

**[2 marks]**

1. **Capturing conversations:** Document your interaction with the Conversations option of the Statistics tab on the upper toolbar on Wireshark. Were you able to capture the network conversation you had with UET’s HTTP server?

**[3 marks]**

### Assessment Rubrics for EE432: Computer Networks Lab 1

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 2 Application Layer Protocol: HyperText Transfer Protocol (HTTP)

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

### Application Layer Protocol: HyperText Transfer Protocol (HTTP)

 **Objectives**

At the end of this lab, students will have achieved the following goals:

* Explore several aspects of the HTTP protocol
* Observe the basic GET/response interaction, HTTP message formats
* Retrieve large HTML files
* Install and configure HTTP server using the Apache webserver software

 **Instructions**

1. Read manual carefully before starting lab.
2. All exercises are individual exercises.
3. You are supposed to provide the answers to the questions listed at the end of this manual in text. Paste screenshots/images in the textboxes where required. You will be required to submit your complete manual on Google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. You must aim to complete the lab half an hour before the lab time ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

 **Background**

Having introduced the Wireshark packet analyzer in the introductory lab, we’re now ready to use Wireshark to investigate protocols in operation, like HTTP, which is a common language of the modern global Internet. The world’s web browsers, servers and related web applications all talk to each other through HTTP, the Hypertext Transfer Protocol. Before proceeding to the experiments, read introductions to some general terms used in this lab, to avoid any confusion.

* + 1. **What is a Web page?**

A Web page (also called a document) consists of objects. An object is a simple file – such as an HTML file, a JPEG image, a GIF image, a Java applet, an audio clip, etc. – that is addressable by a single URL. Most Web pages consist of a base HTML file and several referenced objects. For example, if a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images. The base HTML file references the other objects in the page with the objects' URLs. Each URL has two components: the host name of the server that houses the object and the object's path name. For example, the URL [www.someSchool.edu/someDepartment/picture.gif](http://www.someschool.edu/someDepartment/picture.gif) has [www.someSchool.edu](http://www.someschool.edu/) for a host name and /someDepartment/picture.gif for a path name.

* + 1. **What is a Web browser?**

A browser is a user agent for the Web; it displays to the user the requested Web page and provides numerous navigational and configuration features. Web browsers also implement the client side of HTTP. Thus, in the context of the Web, we will interchangeably use the words "browser" and "client". Popular Web browsers include Google Chrome, Netscape Communicator and Microsoft Explorer.

* + 1. **What is a Web server?**

A Web server houses Web objects, each addressable by a URL. Web servers also implement the server side of HTTP. Popular Web servers include Apache, Microsoft Internet Information Server, and the Netscape Enterprise Server. (Netcraft provides a nice survey of Web server penetration [Netcraft].)

* + 1. **Introduction to HTTP**

The Hypertext Transfer Protocol (HTTP), the Web's application-layer protocol, is at the heart of the Web. HTTP is implemented in two programs: a client program and server program. The client program and server programs, executing on different end systems, talk to each other by exchanging HTTP messages. HTTP defines the structure of these messages and how the client and server exchange the messages. HTTP defines how Web clients (i.e., browsers) request Web pages from servers (i.e., Web servers) and how servers transfer Web pages to clients. When a user requests a Web page (e.g., clicks on a hyperlink), the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects.

* + 1. **Introduction to HTTP server (by studying the Apache webserver software)**
       1. **Apache webserver**

Apache webserver is a very popular open-source freeware webserver software that can be used to run an HTTP server on a machine.

Apache implements an HTTP daemon process: In Unix-terminology, a daemon process refers to a server process that runs in the background and waits for connections/requests to be made to it so that it can service those connections/requests. Webserver software implements an HTTP daemon process that runs and waits in the background for a connection to be made to it by an HTTP client. When the HTTP client initiates a connection by sending a request, the HTTP daemon process is activated and it processes the requests and sends the appropriate response.

* + - 1. **Downloading Apache webserver**

Apache webserver can be downloaded from the site: [http://httpd.apache.org/download.cgi.](http://httpd.apache.org/download.cgi) The current stable version is Apache 2.4.41. Versions for both Unix and Windows are available. The Unix version is more functional and stable and most webservers run on Linux/Unix machines.

However, for our lab, we shall use Apache on Windows since we are only interested in running a webserver and see some basic configuration options of the Apache server.

* + - 1. **Installing Apache webserver**

For the purpose of our lab, Apache webserver (Windows version) will already be installed on the machines. For installing Apache on your home machine, you can follow the instructions at: [http://httpd.apache.org/docs/2.2/install.html.](http://httpd.apache.org/docs/2.2/install.html) Please note that Apache can be installed from: 1) source files where the source files are downloaded and compiled at your machine with flexible options, and 2) binary files where executable files are downloaded and installed; such binary files are precompiled for different OS with common features selected.

 **Lab procedure**

*For all the experiments, we will use Wireshark packet analyzer that we used in the lab 1.*

* + 1. **The basic HTTP GET/response interaction**
       1. **Aim of this exercise**

We will now learn about what packets are exchanged during an HTTP conversation – we will learn about the HTTP GET message that is sent from the HTTP client to the HTTP server and the HTTP message that is sent as response to this message.

* + - 1. **Procedure**

Follow the steps below to complete this exercise and to provide answers to the questions below:

* + - * + Start up your web browser.
        + Start up the Wireshark packet sniffer, as described in lab 1 (but don’t yet begin packet capture). Enter “http” (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We’re only interested in the HTTP protocol here, and don’t want to see the clutter of all captured packets).
        + Wait a bit more than one minute (we’ll see why shortly), and then begin Wireshark packet capture.
        + Enter the following to your browser [http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html.](http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html)Your browser should display the very simple, one-line HTML file.
        + Stop Wireshark packet capture.

The example in [Figure 2.1](#_u5iib77i479p) shows in the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP GET message, which is highlighted in the packet- listing window). Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well.

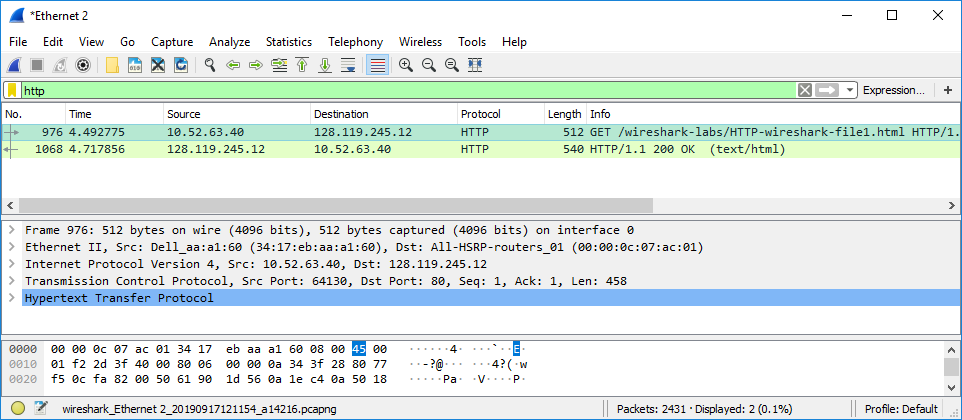


Figure 2.1: Wireshark display after <http://gaia.cs.umass.edu/wireshark-labs/> HTTP-wireshark-file1.html has been retrieved by your browser

By looking at the information in the HTTP GET and response messages, answer the following questions.

* + - 1. **Questions**

1. Which version of HTTP is the browser running 1.0 or 1.1? Which HTTP version is the server running? *Paste screenshots and accompanying text to answer this question.*

**[3 marks]**

1. What languages (if any) does the browser indicate that it can accept to the server? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

1. What is the IP address of your computer and of the gaia.cs.umass.edu server? *Describe how you determined these IP addresses.*

[**[2 marks]**](#_zfn1rg22tgtk)

1. What is the status code returned from the server to your browser? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

1. When was the HTML file that you are retrieving last modified at the server? *Describe how you determined this.*

**[2 marks]**

1. How many bytes of content are being returned to your browser? *Describe how you determined this*

**[2 marks]**

1. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one. *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

* + 1. **Installing and configuring the Apache webserver**
       1. **Aim of this exercise**

We will now do some basic configuration and start the webserver and observe the interaction between HTTP server and HTTP client. Follow the steps below to complete this exercise:

* + - 1. **Procedure: configure webserver at port 8080**

1. Firstly, you will have to configure the webserver to run at port 8080 (in other words, you have to tell the Apache software to start a ‘daemon’ process that will run in the background and wait for connections to be made to it by clients on port 8080).
2. To configure the webserver, you’d have to edit the httpd.conf (that exists in ‘C:\Apache24\conf’).
3. In the configuration file, all lines starting with # are commented out. Benefit from the instructions given in the conf file (these instructions would be in the form of comments – lines starting with # in front).
4. Look for the Listen option in the httpd.conf file and uncomment this (if it is commented) and give the option of 8080 to it (so that the webserver starts to run at port 8080).
5. If you face permissions issue (e.g., you cannot save the httpd.conf file), then you would have to start Wordpad (or the editor you’re using) as Administrator. For this you can search Wordpad, and right click on it and ‘Run it as Administrator’. *Paste a screenshot of the modified httpd.conf file below.*

**[2 marks]**

1. Now, you will run your server at port 8080. For this you shall have to use the httpd command in the bin folder of the Apache installation (which is: C:\Apache24\bin).
2. The command to start the HTTP server is ‘httpd –k start’. If the HTTP server is already running, then you will need to stop the HTTP server using the ‘httpd –k stop’ and then start the server using the command given earlier.
3. You can also restart the HTTP server directly given the command ‘httpd –k restart’. Please note that whenever, you edit the httpd.conf file, you will have to restart the HTTPD server for the configuration changes to take effect. *Paste a screenshot of the command terminal in which you run the commands described above. Capture also the output of these commands and describe what this output says.*

**[2 marks]**

1. Connect to this webserver through your browser and capture the packets on Wireshark.
2. You must disable the proxy server (if configured) on your machine so that your HTTP requests goes directly to your own machine at the configured port. *Paste a screenshot of the packets captured using Wireshark.*

**[2 marks]**

1. Now, you will customize the page displayed by [“http://localhost:8080](http://localhost:8080/)” found in C:\Apache24\htdocs. Do so by editing the HTML code in Wordpad and edit to add a line that says: “*BSEE160xx* has completed first task of Lab-2”. Try to write in blue color and underlined form. You may search on internet on how to edit html.
2. Copy/paste here the modified HTML file reflecting the change that you made. *Provide a screenshot of the modified localhost website with the address bar clearly showing the IP address of your machine and also the port you’re running your webserver on.*

**[2 marks]**

1. Now, you will ask your friend to open a connection to your website and capture the packets they send. Your friend has to specify: [http://IpAddressofYourMachine:8080](http://ipaddressofyourmachine:8080/) in the address bar of their browser. Replace IpAddressofYourMachine with your machine’s actual IP address. Note that the URL is suffixed with ‘:8080’ to demonstrate the fact that your webserver is listening on port 8080 of your machine (which itself is identified by the IP address IpAddressofYourMachine).
2. You will need to open Wireshark and look for the packets in which the source IP belongs to the machine of your friend and the destination IP belongs to your own machine. (You can check the IP of your machine using the ipconfig command).
3. Capture the packets that you see from your friend’s machine using Wireshark to webserver on your machine. *Paste a screenshot here.*

**[2 marks]**



1. Now you will open a connection to your friend’s webserver by typing the following URL in the address bar of your browser to connect to your friend’s machine: http://IpAddressofYourFriendsMachine:PortNumber from your machine. In this address, IpAddressofYourMachine should be replaced with your friend’s machine’s real IP address and

:PortNumber should be replaced with the port the HTTP server on that machine is running. *Paste a screenshot of your browser as it receives the response from your friend’s webserver and use text to explain the screenshot.*

**[2 marks]**

1. What happens when you make the connection when that server is not yet running? What happens when you specify some other port than the port on which that server is running in the URL? *Paste screenshots and use text to explain.*

**[2 marks]**

1. Open a connection to your friend’s webserver with proxy enabled and with proxy disabled. What differences do you observe? *Paste screenshots and use text to explain.*

**[2 marks]**

1. Describe what kind of an http packet was sent from your machine to your friend’s machine. Capture the sent packet using Wireshark. Your machine’s IP address should be the source IP and your friend’s machine IP address should be the destination. *Paste screenshots and use text to explain.*

**[2 marks]**

1. Describe the HTTP response received. What’s the status code in this message? What’s the English description that accompanied this status code? Capture the sent packet using Wireshark. Your machine’s IP address should be the destination IP and your friend’s machine IP address should be the source. *Paste screenshots and use text to explain.*

**[2 marks]**

1. Change the port on which you’re running the server to 8090

*Hint: You will need to edit the httpd.conf file (edit the Listen option) and restart the HTTPD server for the configuration change to take effect.*

*Paste screenshots and use text to explain how you got your server to run at this port.*

**[2 marks]**

* + 1. **The HTTP CONDITIONAL GET/response interaction**
       1. **Aim of this exercise**

We will now learn about a variant of the HTTP GET request message that we’ve seen earlier. We will note how the HTTP CONDITIONAL GET request and the reply to such a request differs from a simple HTTP GET request (which we talked about in exercise [2.4.2](#_49emc69u5xj5)).

* + - 1. **Procedure**

The following are the steps for this exercise:

1. Start up your web browser, and make sure your browser’s cache is cleared, as discussed above.
2. Start up the Wireshark packet sniffer.
3. Enter the following URL into your browser [http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html.](http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html) Your browser should display a very simple five-line HTML file.
4. Quickly enter the same URL into your browser again (or simply select the refresh button on your browser).
5. Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.
6. Filter out all the non-HTTP packets and focus on the HTTP header information in the packet-header detail window.
7. By looking at the information in the HTTP GET and response messages (the first two messages), answer the following questions.
   * + 1. **Questions**
8. Inspect the contents of the first HTTP GET request from the browser to the server. Do you see an “IF-MODIFIED- SINCE” line in the HTTP GET? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

1. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

*Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[3 marks]**

1. Does the response indicate the last time that the requested file was modified? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

1. Now inspect the contents of the second HTTP GET request from the browser to the server. Do you see an “IF- MODIFIED-SINCE:” line in the HTTP GET? If so, what information is contained in the “IF-MODIFIED-SINCE:” header? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

1. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain. *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[3 marks]**

* + 1. **Retrieving long documents**
       1. **Aim of this exercise**

In our examples thus far, the documents retrieved have been simple and short HTML files. Let’s next see what happens when we download a long HTML file.

* + - 1. **Procedure**

Do the following:

1. Start up your web browser, and make sure your browser’s cache is cleared, as discussed above.
2. Start up the Wireshark packet sniffer.
3. Enter the following URL into your browser [http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html.](http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html) Your browser should display the rather lengthy US Bill of Rights.
4. Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed
5. In the packet-listing window, you should see your HTTP GET message, followed by a multiple-packet response to your HTTP GET request. This multiple-packet response deserves a bit of explanation. In our case here, the HTML file is long, and as 4500 bytes is too large to fit in one TCP packet, the single HTTP response message is thus broken into several pieces by TCP, with each piece being contained within a separate TCP segment. Each TCP segment is recorded as a separate packet by Wireshark, and the fact that the single HTTP response was fragmented across multiple TCP packets is indicated by the “Continuation” phrase displayed by Wireshark. We stress here that there is no “Continuation” message in HTTP! Answer the following questions:
   * + 1. **Questions**
6. How many HTTP GET request messages were sent? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**



1. How many data-containing TCP segments were needed to carry the single HTTP response? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

1. What is the status code and phrase associated with the response to the HTTP GET request? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[marks]**

1. Are there any HTTP status lines in the transmitted data associated with a TCP-induced “Continuation”? *Paste screenshot (containing referenced item) and accompanying text to answer this question.*

**[2 marks]**

### Assessment Rubrics for EE432: Computer Networks Lab 2

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 3 Application Layer Protocol: Domain Name System (DNS)

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

24

 **Objectives**

As described in Section 2.5 of the textbook, the Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this lab, we’ll take a closer look at the client side of DNS. Recall that the client’s role in the DNS is relatively simple – a client sends a query to its local DNS server, and receives a response back. Much can go on “under the covers,” invisible to the DNS clients, as the hierarchical DNS servers communicate with each other to either recursively or iteratively resolve the client’s DNS query. From the DNS client’s standpoint, however, the protocol is quite simple – a query is formulated to the local DNS server and a response is received from that server.

Before beginning this lab, you’ll probably want to review DNS by reading Section 2.5 of the textbook. In particular, you may want to review the material on local DNS servers, DNS caching, DNS records and messages, and the TYPE field in the DNS record.

 **Instructions**

* Read carefully before starting the lab.
* These exercises are to be done individually.
* You are supposed to provide the answers to the questions listed at the end of this document and upload the lab report.
* Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/text but you must paraphrase the original work. Your submitted work should be written by yourself.
* Complete the lab half an hour before the lab ends.
* At the end of the lab, a viva will be conducted to evaluate your understanding.

 **Background**

* + 1. **Introduction to DNS**

There are two ways to identify a host – a hostname and an IP address. People prefer the more mnemonic hostname identifier, while routers prefer fixed-length, hierarchically-structured IP addresses. In order to reconcile these different preferences, we need a directory service that translates hostnames to IP addresses. This is the main task of the Internet's Domain Name System (DNS). The DNS is (i) a distributed database implemented in a hierarchy of name servers and (ii) an application-layer protocol that allows hosts and name servers to communicate in order to provide the translation service.

* + 1. **Introduction to NSLOOKUP**

In this lab, we’ll make extensive use of the nslookup tool, which is available in most Linux/Unix and Microsoft platforms today. To run nslookup in Linux/Unix, you just type the nslookup command on the command line. To run it in Windows, open the Command Prompt and run nslookup on the command line.

In its most basic operation, nslookup tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top -level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms). To accomplish this task, nslookup sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.

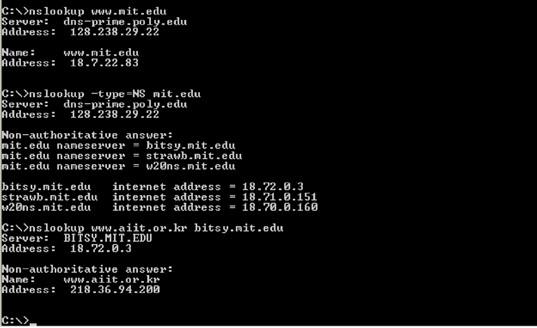


Figure 3.1: “nslookup” commands

The above screenshot shows the results of three independent nslookup commands (displayed in the Windows Command Prompt). In this example, the client host is located on the campus of Polytechnic University in Brooklyn, where the default local DNS server is dns-prime.poly.edu. When running nslookup, if no DNS server is specified, then nslookup sends the query to the default DNS server, which in this case is dns-prime.poly.edu. Consider the first command:

nslookup [www.mit.edu](http://www.mit.edu/)

In words, this command is saying “please send me the IP address for the host [www.mit.edu](http://www.mit.edu/)”. As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of [www.mit.edu.](http://www.mit.edu/) Although the response came from the local DNS server at Polytechnic University, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in Section 2.5 of the textbook.

Now consider the second command:

nslookup –type=NS mit.edu

In this example, we have provided the option “-type=NS” and the domain “mit.edu”. This causes nslookup to send a query for a type-NS record to the default local DNS server. In words, the query is saying, “please send me the host names of the authoritative DNS for mit.edu”. (When the –type option is not used, nslookup uses the default, which is to query for type A records.) The answer, displayed in the above screenshot, first indicates the DNS server that is providing the answer (which is the default local DNS server) along with three MIT nameservers. Each of these servers is indeed an authoritative DNS server for the hosts on the MIT campus. However, nslookup also indicates that the answer is “non -authoritative,” meaning that this answer came from the cache of some server rather than from an authoritative MIT DNS server. Finally, the answer also includes the IP addresses of the authoritative DNS servers at MIT. (Even though the type-NS query generated by nslookup did not explicitly ask for the IP addresses, the local DNS server returned these “for free” and nslookup displays the result.)

Now finally consider the third command:

nslookup [www.aiit.or.kr](http://www.aiit.or.kr/) bitsy.mit.edu

In this example, we indicate that we want the query sent to the DNS server bitsy.mit.edu rather than to the default DNS server (dns-prime.poly.edu). Thus, the query and reply transaction takes place directly between our querying host and bitsy.mit.edu. In this example, the DNS server bitsy.mit.edu provides the IP address of the host [www.aiit.or.kr,](http://www.aiit.or.kr/) which is a web server at the Advanced InstUETte of Information Technology (in Korea).

nslookup commands. The syntax is:

nslookup –option1 –option2 host-to-find dns-server

In general, nslookup can be run with zero, one, two or more options. And as we have seen in the above examples, the dns- server is optional as well; if it is not supplied, the query is sent to the default local DNS server.

* + 1. **Introduction to IPCONFIG**

ipconfig (for Windows) and ifconfig (for Linux/Unix) are among the most useful little utilities in your host, especially for debugging network issues. Here we’ll only describe ipconfig, although the Linux/Unix ifconfig is very similar. ipconfig can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you all this information about your host simply by entering:

Ipconfig/all

into the Command Prompt, as shown in the following screenshot.

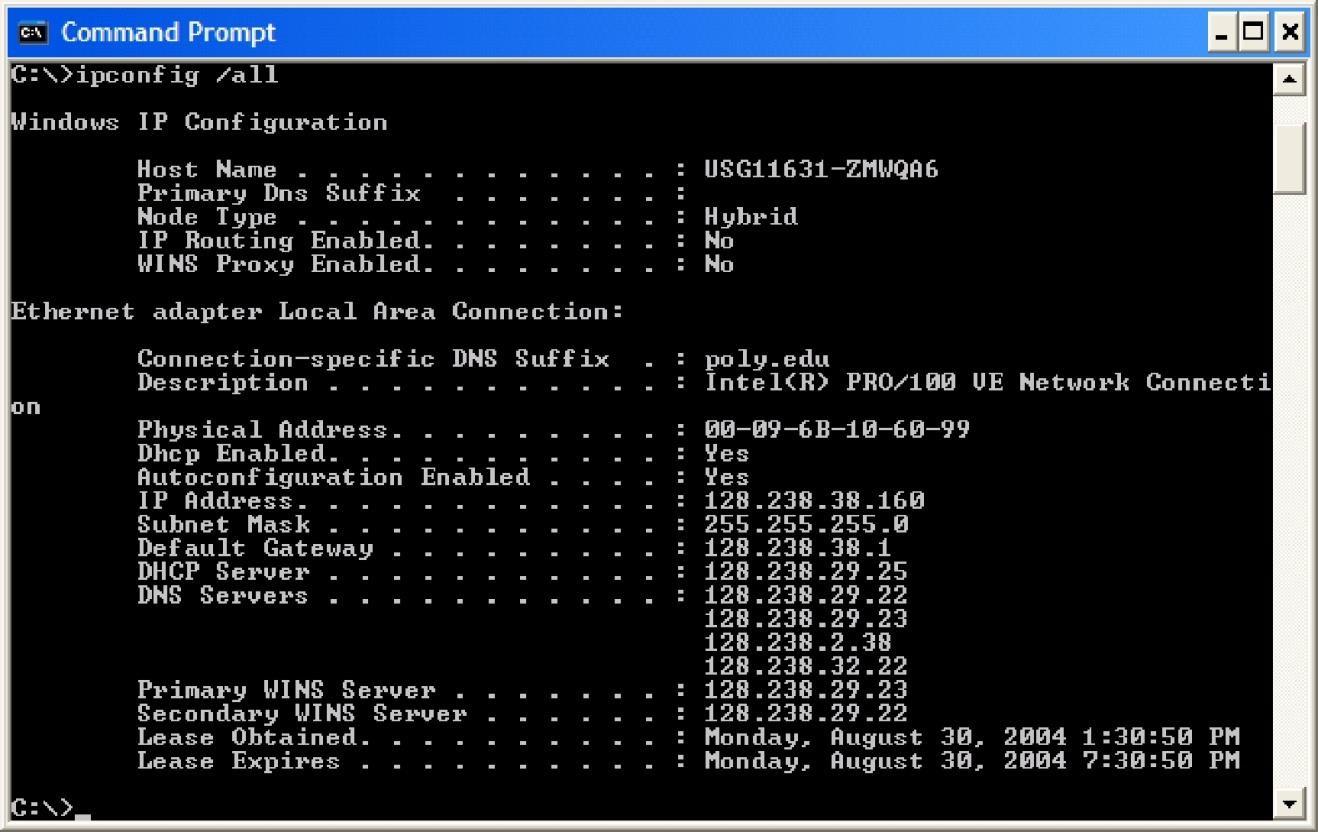


Figure 3.2: “ipconfig”

ipconfig is also very useful for managing the DNS information stored in your host. In Section 2.5 we learned that a host can cache DNS records it recently obtained. To see these cached records, after the prompt C:\> provide the following command:

ipconfig /displaydns

Each entry shows the remaining Time to Live (TTL) in seconds. To clear the cache, enter:

ipconfig /flushdns

Flushing the DNS cache clears all entries and reloads the entries from the hosts file.

 **Lab Procedure**

* + 1. **NSLOOKUP**

Now that we have provided an overview of nslookup, it is time for you to test drive it yourself. Do the following (and write down the results):

1. Run nslookup to obtain the IP address of the Web server hosting [www.uet.edu.pk.](http://www.itu.edu.pk/) *Paste a screenshot of the output of this command here.*

**[2 marks]**

1. Run nslookup to determine the authoritative DNS servers for [www.uet.edu.pk.](http://www.uet.edu.pk/) Provide both the names of these DNS servers and also the IP addresses of these DNS servers. *Paste a screenshot of your work clearly showing both the names of the DNS servers and also the IP addresses. This may require you to use multiple commands.*

**[2 marks]**

1. Run nslookup to determine the mail servers for [www.uet.edu.pk.](http://www.uet.edu.pk/) Provide both the names of these Mail servers and also the IP addresses of these Mail servers. *Paste a screenshot of your work clearly showing both the names of the Mail servers and also the IP addresses. This may require you to use multiple commands.*

**[2 marks]**

1. Query the public DNS service provided by Google at 8.8.8.8 to query for the IPv6 address of ipv6.google.com. Provide the IPv6 address. Note how this address is different from the IPv4 addresses that you were getting for the previous questions. *Paste a screenshot here to show how you got the IPv6 address. Also, describe how the IPv6 address is different from IPv4 address (you may search on the Internet to look up details on the differences in the address and presentation format).*

**[2 marks]**

* + 1. **Tracing DNS with Wireshark (while using browser)**

Now that we are familiar with nslookup and ipconfig, we’re ready to get down to some serious business. Let’s first capture the DNS packets that are generated by ordinary Web-surfing activity.

1. Use ipconfig to empty the DNS cache in your host.
2. Open your browser and empty your browser cache. (With In[ternet Explorer, go to Tools menu and select Internet](#_n8zmhk98yi6s) Options; then in the General tab select Delete Files.)
3. Open Wireshark and enter “ip.addr == your\_IP\_address” into the filter, where you obtain your\_IP\_address with ipconfig. This filter removes all packets that neither originate nor are destined to your host.
4. Start packet capture in Wireshark.
5. With your browser, visit the Web page: [http://www.ietf.org](http://www.ietf.org/)
6. Stop packet capture.
7. If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author’s (of the textbook [K&R]) computers.
8. Download the traces for DNS (downloadable from CCN classroom attached with “Lab-03” assignment).
9. Load the trace file dns-ethereal-trace-1 by using the File pull down menu, choosing Open and selecting the appropriate trace file.

**3.4.2.1. Questions**

1. Locate the DNS query and response messages. Are they sent over UDP or TCP (i.e., what transport layer protocol is being used)? *Paste a screenshot and use accompanying text to answer this question.*

**[2 marks]**

1. What is the destination port for the DNS query message? What is the source port of DNS response message? *Paste a screenshot and use accompanying text to answer this question.*

**[2 marks]**

1. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same? *Paste screenshots and use supporting text to support your answer.*

**[2 marks]**

1. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

*Paste a screenshot and use accompanying text to answer this question.*

**[2 marks]**

1. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?

*Paste a screenshot and use accompanying text to answer this question.*

**[2 marks]**

* + 1. **Tracing DNS with Wireshark (using the nslookup command)**

Now let’s play with nslookup again.

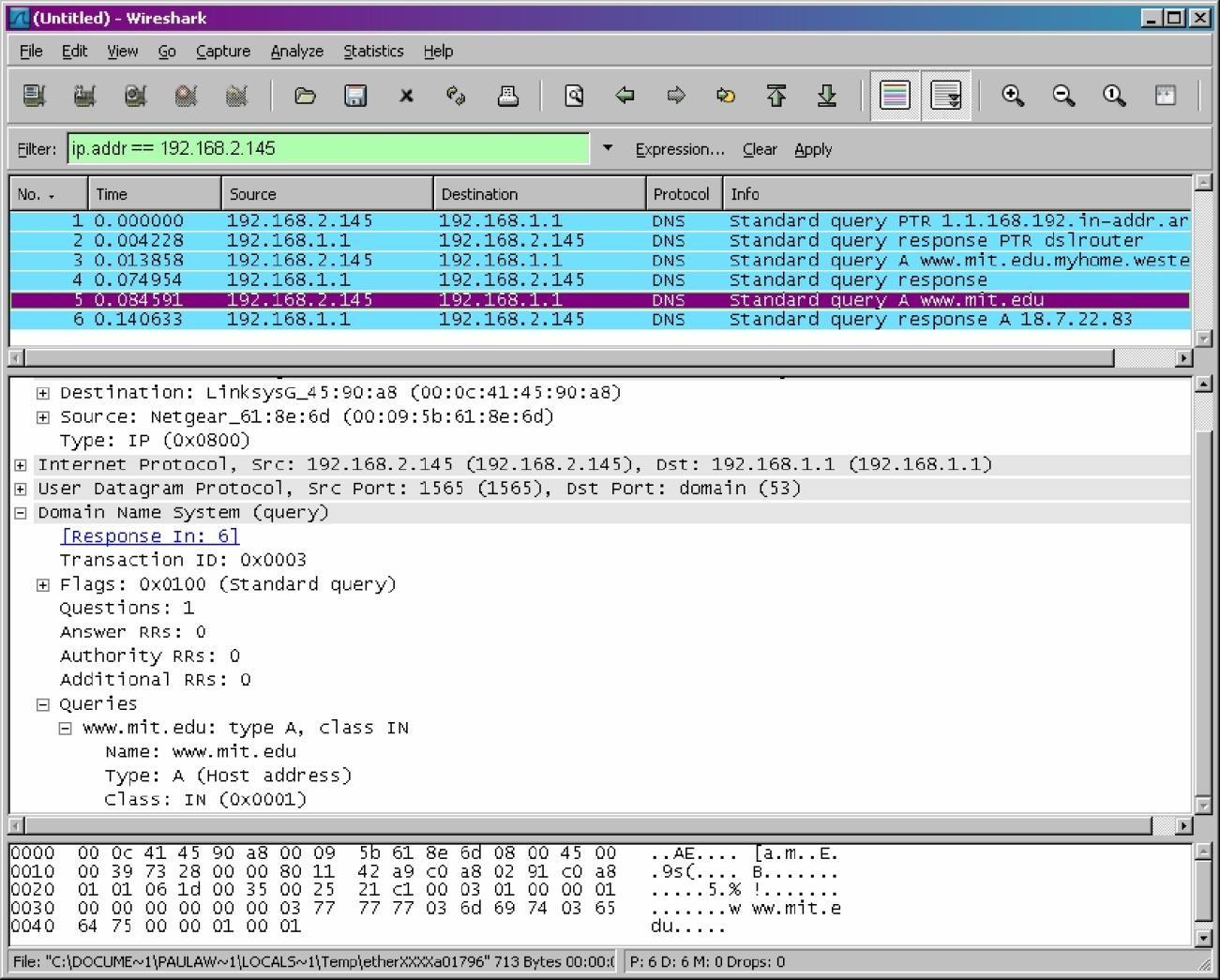
1. Start packet capture.
2. Do an nslookup on [www.mit.edu.](http://www.mit.edu/)
3. Stop packet capture.
4. You should get a trace that looks something like the following:

Figure 3.3: NSLOOKUP trace for “mit.edu.pk”

We see from the above screenshot that nslookup actually sent three DNS queries and received three DNS responses. For the purpose of this assignment, in answering the following questions, ignore the first two sets of queries/responses, as they are specific to nslookup and are not normally generated by standard Internet applications. You should instead focus on the last query and response messages.

If you are unable to run Wireshark on a live network connection, load the trace file dns-ethereal-trace-2 (downloadable from CCN classroom attachment under “Lab-03”).

**3.4.3.1. Questions**

1. What is the destination port for the DNS query message? What is the source port of DNS response message? *Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

1. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? *Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

1. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

*Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

1. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?

*Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

Now repeat the previous experiment, but instead issue the command:

nslookup –type=NS mit.edu

If you are unable to run Wireshark on a live network connection, load the trace file dns-ethereal-trace-3 (downloadable from CCN classroom attached with “Lab-03” assignment).

1. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? *Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

1. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

*Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

1. Examine the DNS response message. What MIT nameservers does the response message provide? Does this response message also provide the IP addresses of the MIT nameservers? *Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

Now repeat the previous experiment, but instead issue the command:

nslookup [www.aiit.or.kr](http://www.aiit.or.kr/) bitsy.mit.edu

If you are unable to run Wireshark on a live network connection, load the trace file dns-ethereal-trace-4 (downloadable from CCN classroom attached with “Lab-03” assignment).

1. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to? *Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

1. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

*Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

1. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?

*Paste screenshots and use supporting text to support your answer.*

**[3 marks]**

### Assessment Rubrics for EE432: Computer Networks Lab 3

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 4 Transport Layer Protocol: User Datagram Protocol (UDP)

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

### Transport Layer Protocol: User Datagram Protocol (UDP)

 **Objectives**

In performing this lab, students will achieve the following goals:

* Analyze the behavior of UDP in detail
* Determine the number of fields in UDP header, the value in the UDP header fields, and maximum number of bytes in UDP payload, source & destination port numbers

 **Instructions**

1. Read carefully before starting the lab.
2. These exercises are to be done individually.
3. You are supposed to provide the answers to the questions listed at the end of this manual by hand or in text and submit the hardcopy to lab instructor at the end of lab. In the case that you are required to submit images/screenshots, you will be required to submit the files on Google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. Complete the lab half an hour before the lab ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

 **Background**

* + 1. **Introduction to UDP**

UDP (User Datagram Protocol) is a simple transport layer protocol for client/server network applications based on Internet Protocol (IP). UDP is the main alternative to TCP and one of the oldest network protocols in existence, introduced in 1980. UDP is often used in videoconferencing applications or computer games specially tuned for real-time performance. To achieve higher performance, the protocol allows individual packets to be dropped (with no retries) and UDP packets to be received in a different order than they were sent as dictated by the application.

* + 1. **UDP Datagrams**

UDP network traffic is organized in the form of datagrams. A datagram comprises one message unit. The first eight (8) bytes of a datagram contain header information and the remaining bytes contain message data.

A UDP datagram header consists of four (4) fields of two bytes each: Source port number, Destination port number, Datagram size and checksum.

* + - 1. **UDP port number**

UDP port numbers allow different applications to maintain their own channels for data similar to TCP. UDP port headers are two bytes long; therefore, valid UDP port numbers range from 0 to 65535.

* + - 1. **Datagram size**

The UDP datagram size is a count of the total number of bytes contained in header and data sections. As the header length is a fixed size, this field effectively tracks the length of the variable-sized data portion (sometimes called payload). The size of datagrams varies depending on the operating environment but has a maximum of 65535 bytes.

* + - 1. **Checksum**

UDP checksums protect message data from tampering. The checksum value represents an encoding of the datagram data calculated first by the sender and later by the receiver. Should an individual datagram be tampered with or get corrupted during transmission, the UDP protocol detects a checksum calculation mismatch. In UDP, checksumming is optional as opposed to TCP where checksums are mandatory.

 **Procedure**

Do the following:

1. Start up the Wireshark software.
2. Begin packet capture, select the Capture pull down menu and select Options.
3. Selecting the network interface on which packets would be captured: You can use most of the default values in this window. The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the Interface pull down menu at the top of the Capture Options window. Click Start. Packet capture will now begin.
4. Start up your favorite web browser, and type any site which uses the UDP packets for traffic flow in the packet listing window.
5. Stopping the capture and inspecting captured packets: After your browser has displayed the page, stop Wireshark packet capture
6. Filtering: Filter the UDP packets.
7. Details of a packet: Select the UDP messages shown in the packet-listing window and analyze by looking into the detail of packets pane and answer the questions given at the end of this document.
8. Obtaining credit for this lab: Now, please proceed to the questions section to answer the questions. You must note down your answers in this file itself. Please note that every group must upload this file (after duly filling in the answers) through the appropriate link at your LMS course site for the specific date of your lab to obtain credit. Please clarify with your instructor/ lab engineer if you have any queries.

 **Questions**

1. Select one packet and determine the source MAC address of that UDP packet.
2. Select one packet and determine the destination MAC address of that UDP packet.



1. Select one packet and determine how many fields are there in the UDP header.
2. List the name of these fields.



1. From the packet content field, determine the length (in bytes) of each of the UDP header fields.
2. What is the source and the destination port number of UDP packet.



1. Analyze the udp packet and answer that the value in the Length field is the length of what? Verify your claim with your captured UDP packet.



1. What is the maximum number of bytes that can be included in a UDP payload.
2. What is the largest possible source port number?
3. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. (To answer this question, you’ll need to look into the IP header.)



1. Examine a pair of UDP packets in which the first packet is sent by your host and the second packet is a reply to the first packet. Describe the relationship between the port numbers in the two packets.



1. Show the data attached to the UDP packet in both hexadecimal and decimal notation.

### Assessment Rubrics for EE432: Computer Networks Lab 4

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 5 Transport Layer Protocol: Transmission Control Protocol (TCP)

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

### Transfer Layer Protocol: Transmission Control Protocol (TCP)

 **Objectives**

In this lab, we’ll investigate the behavior of TCP in detail. We’ll do so by analyzing a trace of the TCP segments sent and received in transferring a file from your computer to a remote server. We’ll study TCP’s use of sequence and acknowledgement numbers for providing reliable data transfer.

 **Instructions**

1. Read carefully before starting the lab.
2. These exercises are to be done individually.
3. You are supposed to provide the answers to the questions listed at the end of this document and upload the completed report to your google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. Complete the lab half an hour before the lab ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

 **Background**

* + 1. **The Transmission Control Protocol (TCP) Introduction**

TCP provides connections between clients and servers. A TCP client establishes a connection with a given server, exchanges data with that server across the connection, and then terminates the connection.

TCP also provides reliability. When TCP sends data to the other end, it requires an acknowledgment in return. If an acknowledgment is not received, TCP automatically retransmits the data and waits a longer amount of time. After some number of retransmissions, TCP will give up, with the total amount of time spent trying to send data typically between 4 and 10 minutes (depending on the implementation).

TCP contains algorithms to estimate the round-trip time (RTT) between a client and server dynamically so that it knows how long to wait for an acknowledgment. For example, the RTT on a LAN can be milliseconds while across a WAN, it can be seconds. Furthermore, TCP continuously estimates the RTT of a given connection, because the RTT is affected by variations in the network traffic.

TCP also sequences the data by associating a sequence number with every byte that it sends. For example, assume an application writes 2,048 bytes to a TCP socket, causing TCP to send two segments, the first containing the data with sequence numbers 1–1,024 and the second containing the data with sequence numbers 1,025–2,048. (A segment is the unit of data that TCP passes to IP.) If the segments arrive out of order, the receiving TCP will reorder the two segments based on their sequence numbers before passing the data to the receiving application. If TCP receives duplicate data from its peer (say the peer thought a segment was lost and retransmitted it, when it wasn't really lost, the network was just overloaded), it can detect that the data has been duplicated (from the sequence numbers), and discard the duplicate data.

There is no reliability provided by UDP. UDP itself does not provide anything like acknowledgments, sequence numbers, RTT estimation, timeouts, or retransmissions. If a UDP datagram is duplicated in the network, two copies can be delivered to the receiving host. Also, if a UDP client sends two datagrams to the same destination, they can be reordered by the network and arrive out of order.

TCP provides flow control. TCP always tells its peer exactly how many bytes of data it is willing to accept from the peer at any one time. This is called the advertised window. At any time, the window is the amount of room currently available in the receive buffer, guaranteeing that the sender cannot overflow the receive buffer. The window changes dynamically over time: As data is received from the sender, the window size decreases, but as the receiving application reads data from the buffer, the window size increases. It is possible for the window to reach 0: when TCP's receive buffer for a socket is full and it must wait for the application to read data from the buffer before it can take any more data from the peer.

given connection at any time. This means that TCP must keep track of state information such as sequence numbers and window sizes for each direction of data flow: sending and receiving. After a full-duplex connection is established, it can be turned into a simplex connection if desired.

* + 1. **TCP Connection Establishment and Termination**
       1. **Three-Way Handshake**

Following scenario occurs when a TCP connection is established:

1. The server must be prepared to accept an incoming connection. This is normally done by calling socket, bind, and listen and is called a passive open.
2. The client issues an active open by calling connect. This causes the client TCP to send a "synchronize" (SYN) segment, which tells the server the client's initial sequence number for the data that the client will send on the connection. Normally, there is no data sent with the SYN; it just contains an IP header, a TCP header, and possible TCP options (which we will talk about shortly).
3. The server must acknowledge (ACK) the client's SYN and the server must also send its own SYN containing the initial sequence number for the data that the server will send on the connection. The server sends its SYN and the ACK of the client's SYN in a single segment.
4. The client must acknowledge the server's SYN.

The minimum number of packets required for this exchange is three; hence, this is called TCP's three-way handshake. We show the three segments in [Figure 5.1.](#_jumhc4b5w52h)

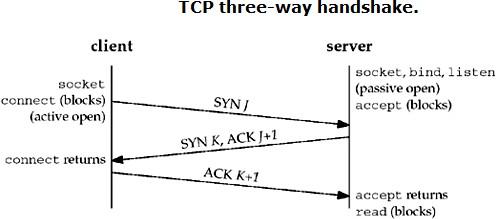


Figure 5.1: TCP three-way handshake.

We show the client's initial sequence number as J and the server's initial sequence number as K. The acknowledgment number in an ACK is the next expected sequence number for the end sending the ACK. Since a SYN occupies one byte of the sequence number space, the acknowledgment number in the ACK of each SYN is the initial sequence number plus one. Similarly, the ACK of each FIN is the sequence number of the FIN plus one.

An everyday analogy for establishing a TCP connection is the telephone system. The socket function is the equivalent of having a telephone to use. bind is telling other people your telephone number so that they can call you. listen is turning on the ringer so that you will hear when an incoming call arrives. connect requires that we know the other person's phone number and dial it. accept is when the person being called answers the phone. Having the client's identity returned by accept (where the identify is the client's IP address and port number) is similar to having the caller ID feature show the caller's phone number. One difference, however, is that accept returns the client's identity only after the connection has been established, whereas the caller ID feature shows the caller's phone number before we choose whether to answer the phone or not.

* + - 1. **TCP Connection Termination**

While it takes three segments to establish a connection, it takes four to terminate a connection:

1. One application calls close first, and we say that this end performs the active close. This end's TCP sends a FIN segment, which means it is finished sending data.
2. The other end that receives the FIN performs the passive close. The received FIN is acknowledged by TCP. The receipt of the FIN is also passed to the application as an end-of-file (after any data that may have already been queued for the application to receive), since the receipt of the FIN means the application will not receive any additional data on the connection.
3. Sometime later, the application that received the end-of-file will close its socket. This causes its TCP to send a FIN.
4. The TCP on the system that receives this final FIN (the end that did the active close) acknowledges the FIN.

Since a FIN and an ACK are required in each direction, four segments are normally required. We use the qualifier "normally" because in some scenarios, the FIN in Step 1 is sent with data. Also, the segments in Steps 2 and 3 are both from the end performing the passive close and could be combined into one segment. We show these packets in [Figure 5.2.](#_yw43apycgnbm)

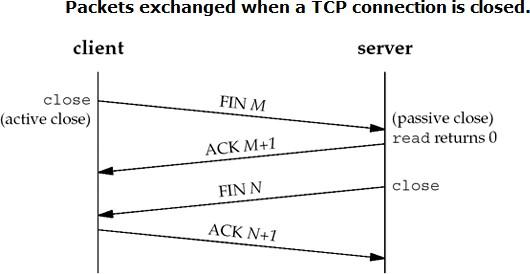


Figure 5.2: Packets exchanged when a TCP connection is closed.

A FIN occupies one byte of sequence number space just like a SYN. Therefore, the ACK of each FIN is the sequence number of the FIN plus one.

Between Steps 2 and 3 it is possible for data to flow from the end doing the passive close to the end doing the active close. This is called a half.

The sending of each FIN occurs when a socket is closed. We indicated that the application calls close for this to happen, but realize that when a Unix process terminates, either voluntarily (calling exit or having the main function return) or involuntarily (receiving a signal that terminates the process), all open descriptors are closed, which will also cause a FIN to be sent on any TCP connection that is still open.

Although we show the client in [Figure 5.2](#_yw43apycgnbm) performing the active close, either end – the client or the server – can perform the active close. Often the client performs the active close, but with some protocols (notably HTTP), the server performs the active close.

 **Procedure**

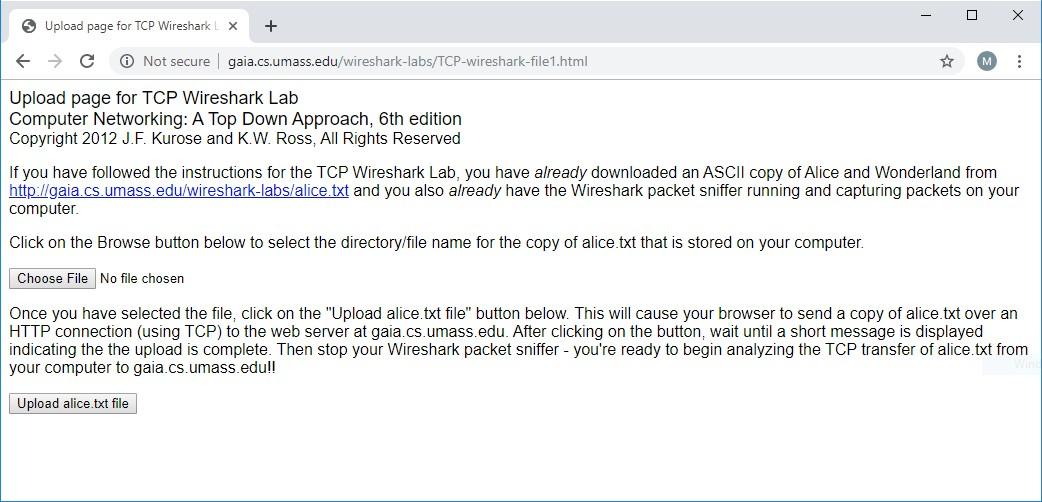
1. Capturing a bulk TCP transfer from your computer to a remote server: Before beginning our exploration of TCP, we’ll need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You’ll do so by accessing a Web page that will allow you to enter the name of a file stored on your computer and then transfer the file to a Web server using the HTTP POST method. We’re using the POST method rather than the GET method as we’d like to transfer a large amount of data from your computer to another computer. Of course, we’ll be running Wireshark during this time to obtain the trace of the TCP segments sent and received from your computer.
2. Start up your web browser. Go the <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> [and retrieve an ASCII copy of](#_w6zd4z632zq7) Alice in Wonderland. Store this file somewhere on your computer.
3. Next go to [http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.](http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html) You should see a screen that looks like:

Figure 5.3: Screenshot of “[http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html”](http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html)

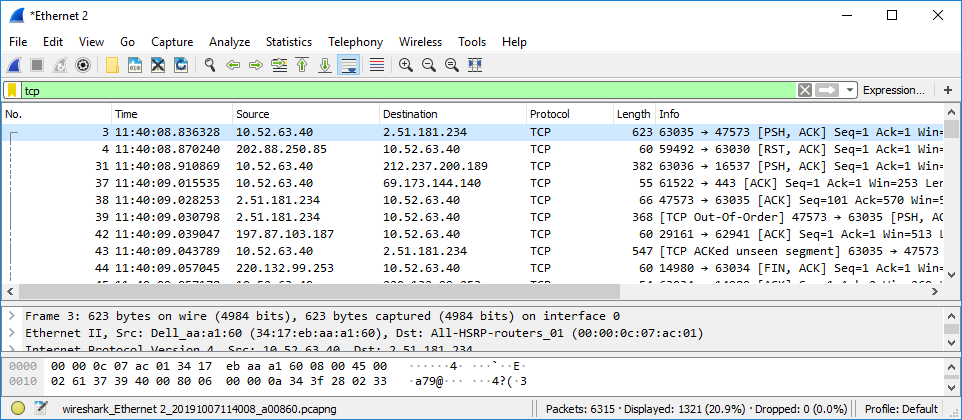
1. Use the Choose File button in this form to enter the name of the file (full path name) on your computer containing Alice in Wonderland (or do so manually). Don’t yet press the “Upload alice.txt file” button.
2. Now start up Wireshark and begin packet capture (Capture->Start) and then press OK on the Wireshark Packet Capture Options screen.
3. Returning to your browser, press the “Upload alice.txt file” button to upload the file to the gaia.cs.umass.edu server. Once the file has been uploaded, a short congratulations message will be displayed in your browser window.
4. Stop Wireshark packet capture. Your Wireshark window should look similar to the window shown below.

Figure 5.4: Wireshark window after packet capture

1. First, filter the packets displayed in the Wireshark window by entering “tcp” into the display filter specification window towards the top of the Wireshark window. What you should see is series of TCP and HTTP messages between your computer and gaia.cs.umass.edu. You should see the initial three-way handshake containing a SYN message. You should see an HTTP POST message and a series of “HTTP Continuation” messages being sent from your computer to gaia.cs.umass.edu. Recall from our discussion in the earlier HTTP Wireshark lab, that is no such thing as an HTTP Continuation message – this is Wireshark’s way of indicating that there are multiple TCP segments being

used to carry a single HTTP message. You should also see TCP ACK segments being returned from gaia.cs.umass.edu to your computer.

1. **Obtaining credit for this lab:** Now, please proceed to the questions section to answer the questions. You must note down your answers in this file itself. Please note that every student must upload this file (after duly filling in the answers) on Google Classroom to obtain credit. Please clarify with your instructor/lab engineer if you have any queries.

**2.4.1. Questions**

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window”.



1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?



1. Recall the TCP lecture studied in class and explain the content of the TCP header like sequence number, acknowledgement number and checksum etc.



1. What type of http packets used in transferring file to gaia.cs.umass.edu?
2. By looking into which field in TCP segment you can identify that a given segment is a SYN segment? What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu?
3. What is the header length of TCP verify it with Wire-shark?
4. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the ACKnowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value?



1. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to look into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.



1. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first 4 segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received?



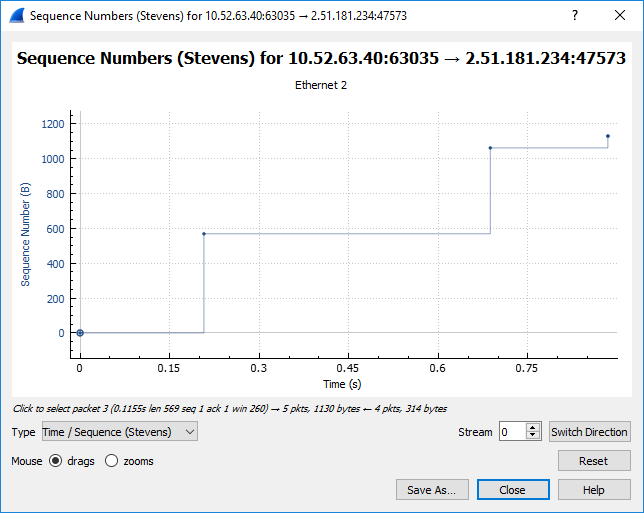
1. What is the length of each of the first four TCP segments?
2. What is the minimum amount of available buffer space advertised at the receiver?



1. Calculate the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated it.
2. How can you find that a packet is retransmitted?



Let’s now examine the amount of data sent per unit time from the client to the server. Rather than calculating this from the raw data in the Wireshark window, we’ll use one of Wireshark’s TCP graphing utilities - Time-Sequence-Graph(Stevens) - to plot out data. Select a TCP segment in the Wireshark’s “listing of captured-packets” window. Then select the menu : Statistics-> TCP Stream Graph-> Time-Sequence-Graph(Stevens).

Figure 5.5: Time sequence graph (Stevens)

Here, each dot represents a TCP segment sent, plotting the sequence number of the segment versus the time at which it was sent. Note that a set of dots stacked above each other represents a series of packets that were sent back-to-back by the sender.

1. Use the Time-Sequence-Graph(Stevens) plotting tool and show the plot you obtained for the TCP segment. Also explain that graph in few lines.



1. Use the TCP stream graph and plot the throughput graph also explain it in few lines.

### Assessment Rubrics for EE432: Computer Networks Lab 5

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 6 Socket Programming

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

### Socket Programming

 **Objectives**

At the end of this lab, the students should be able to:

* Explain the concepts of client server communication
* Setup client/server communication
* Use the sockets interface of Python and C programming language
* Implement Client /Server (Daytime and echo client server) model using TCP Socket

 **Instructions**

1. Read manual carefully before starting lab.
2. All exercises are individual exercises
3. You are supposed to provide the answers to the questions listed at the end of this manual by hand or in text and submit the hardcopy to lab instructor at the end of lab. In the case that you are required to submit images/screenshots, you will be required to submit the files on Google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. You must aim to complete the lab half an hour before the lab time ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

 **Background**

* + 1. **Application programming interface**

An application programming interface (API) is a specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables.

* + 1. **Network Application Programming Interface**

The place to start when implementing a network application is the 'interface exported by network'. Generally all operating systems provide an interface to its networking sub system. This interface is called as the 'Network Application Programming Interface' (Network API) or socket interface.

* + 1. **Network Sockets**

A network socket is an endpoint of an inter-process communication flow across a computer network. Today, most communication between computers is based on the Internet Protocol; therefore most network sockets are Internet sockets.

The socket is a special file in UNIX. The socket interface defines various operations for creating a socket, attaching the socket to the network, sending/receiving messages through the socket and so on. Any application uses a socket primitive to established a connection between client and server.

A socket address is the combination of an IP address and a port number, much like one end of a telephone connection is the combination of a phone number and a particular extension. Based on this address, internet sockets deliver incoming data packets to the appropriate application process or thread.

* + 1. **Socket API**

A socket API is an application programming interface (API), usually provided by the operating system, that allows application programs to control and use network sockets. Internet socket APIs are usually based on the Berkeley sockets standard.

* + 1. **Berkeley Socket**

The following are some general information about Berkeley Sockets

1. Developed in the early 1980s at the University of California at Berkeley. There are no longer any major alternatives. Other major alternative was TLI (Transport Layer Interface). There are communications tools that are built on tool of Berkeley sockets.
2. It is an API.
3. Its implementation usually requires kernel code.
4. It is the standard for communications programming.
5. Can use the UNIX read, write, close, select, etc. system calls.
6. Supports broadcast. This is where the same message may be delivered to multiple systems on a network without additional overhead.
7. Available on every UNIX system that I know of and somewhat available in WIN32.
8. Build for client/server development. That is having one system provide a service to other systems.
   * 1. **Client/Server Communication**

At a basic level, network-based systems consist of a server, client, and a media for communication as shown in [Figure 6.1.](#_hbfmccgoju4m) A computer running a program that makes a request for services is called client machine. A computer running a program that offers requested services from one or more clients is called server machine. The media for communication can be wired or wireless network.

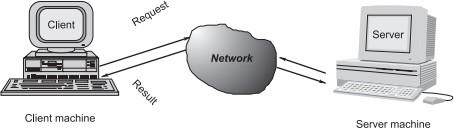


Figure 6.1: Client/server communication

Generally, programs running on client machines make requests to a program (often called as server program) running on a server machine. They involve networking services provided by the transport layer, which is part of the Internet software stack, often called TCP/IP (Transport Control Protocol/Internet Protocol) stack, the transport layer comprises two types of protocols, TCP (Transport Control Protocol) and UDP (User Datagram Protocol)

* + 1. **Port Numbers**

At any given time, multiple processes can be using any given transport: UDP, SCTP, or TCP. All three transport layers use 16-bit integer port numbers to differentiate between these processes. When a client wants to contact a server, the client must identify the server with which it wants to communicate. The TCP and UDP protocols use ports to map incoming data to a particular process running on a computer. Some ports have been reserved to support common/well known services:

ftp 21/tcp telnet 23/tcp smtp 25/tcp login 513/tcp http 80/tcp,udp

User-level process/services generally use port number value >= 1024.

* + 1. **Socket Functions For Elementary Tcp Client/Server**

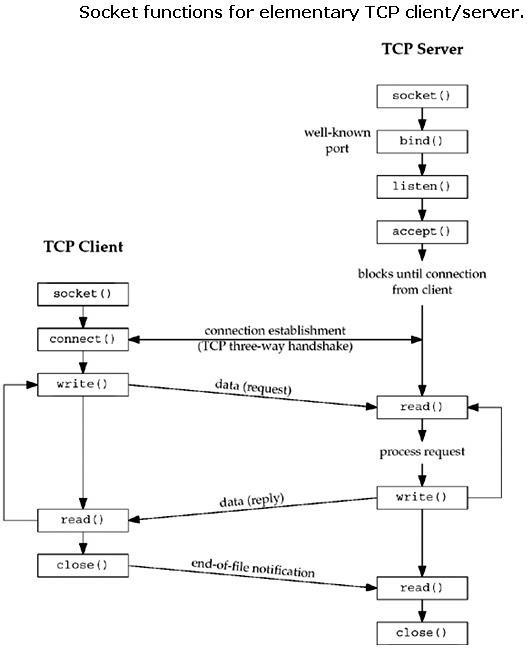


Figure 6.2: Socket functions for elementary TCP/IP client/server

* + - 1. **SOCKET()**

To perform network I/O , the first thing a process must do is call to the socket function.

# include <sys/socket.h>

int socket(int family,int type,int protocol);

returns -1 on error.

Here, family specifies the protocol family. The family can be one of the following.

* + - * + AF\_INET – Internet protocol
        + AF\_UNIX – Unix internal protocol Here AF prefix stands for “Address Family”

The second argument, socket type can be one of the following.

* + - * + SOCK\_STREAM – Stream socket
        + SOCK\_DGRAM – Datagram socket
        + SOCK\_SEQPACKET – Sequences packet socket
        + SOCK \_RAW – Raw socket.

The third argument to the socket system call is usually set to 0. On success the socket function returns a small non-negative integer value, similar to file descriptor.

* + - 1. **CONNECT()**

The connect function is used by a TCP client to establish a connection with a TCP server.

# include <sys/socket.h>

int connect(int sockfd, (const struct sockaddr\*) &servaddr, socklen\_t addrlen);

Returns 0 on success and -1 on error.

Here 'sockfd' is a socket descriptor returned by a socket function. The second and the third arguments are a pointer to a socket address structure and its size. Here the socket address structure servaddr must contain the IP address and port number of the server.

* + - 1. **BIND()**

The server's port is bound to the socket by filling in the Internet Socket address structure and calling BIND system call. Using BIND function, we associate the created socket with a port on the local machine.

# include < sys/socket/h>

struct sockaddr\_in my\_addr; // Internet socket address structure my\_addr.sin\_family = AF\_INET; // set the protocol family my\_addr.sin\_port = htons(MYPORT); // set PORT number you want to bind my\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY); // use my IP address

int bind(int sockfd, const struct sockaddr \* myaddr, socklen\_t addrlen);

Returns 0 on success and -1 on error.

sockfd is the socket file descriptor returned by socket(). my\_addr is a pointer to a struct sockaddr that contains information about server's address, namely, port and IP address. By specifying Wild card INADDR\_ANY allows the server to accept connection on any interface addrlen is be set to sizeof(struct sockaddr). by setting my\_addr.sin\_addr.s\_addr to INADDR\_ANY, you are telling it to automatically fill in the IP address of the machine the process is running on.

* + - 1. **LISTEN()**

The listen() is called only by the TCP server and it converts an unconnected socket into a passive socket, indicating that the kernel should accept incoming connection request directed to this socket. The second argument to this function specifies the maximum number of connections the kernel should queue for this socket.

int listen(int sockfd,int backlog);

Returns 0 on success and -1 on error.

* + - 1. **ACCEPT()**

Accept() is called by the TCP server. Accept() is a blocking operation that does not return until a remote client has establish a connection. When it does complete, it returns a new socket that corresponds to this newly established connection.

Syntax:

int accept(int sockfd,struct sockaddr \* cliaddr,socklen\_t \*addrlen);

Returns non-negative descriptor if OK and -1 on error.

* + - 1. **CLOSE()**

Closes a socket and deletes descriptor from system tables. All data sent before close are delivered to other side. After close(), sockfd is not valid for reading or writing.

Syntax :

int close (int sockfd);

Returns non-negative descriptor if OK and -1 on error. Sockfd: socket file descriptor (returned from socket) Socket include files:

* + - * + sys/types.h – Types used in sys/socket.h and netinet/in.h
        + netinet/in.h – Internet domain address structures and functions
        + sys/socket.h – Structures and functions used for socket API.i accept(), bind(), connect(), listen(), recv(), send(), setsockopt(), shutdown(), etc .
        + arpa/inet.h – Definitions for internet operations. Prototypes functions such as htonl(), htons(), ntohl(), tohs(), inet\_addr(), inet\_ntoa(), etc ...
        + unistd.h – Defines constants and types
        + errno.h – Defines sytem error numbers

 **Lab Procedure**

* + 1. **Running a very simple Python UDP client/server application**

To get a feel for network programming, we shall firstly run very simple Python applications that use sockets to communicate. The application we consider is very minimal, and has no error/exception handling, but serves the purpose of illustrating the various system calls used in the Berkeley Sockets API. We note here that Python Socket API is closely modeled on the BSD Sockets API supported by C (which we shall get to in [2.4.3](#_3vwyadcu642r)).

1. Download the PythonUDPClient and PythonUDPServer file from the CCN classroom. Make these files executable by issuing the following commands:

$ chmod +x PythonUDPClient.py

$ chmod +x PythonUDPServer.py

1. Open two terminals and open PythonUDPClient in one terminal (by issuing the command ./PythonUDPClient.py in the directory hosting this file) and PythonUDPServer (by issuing the command ./PythonUDPServer.py while being in the directory hosting PythonUDPServer.py).

(terminal 1): $ ./PythonUDPClient.py (terminal 2): $ ./PythonUDPServer.py

1. Alternatively, these Python files can be run by giving the following command:

(terminal 1): $ python ./PythonUDPClient.py (terminal 2): $ python ./PythonUDPServer.py

1. To run the server in the background, waiting for connections you may suffix the command with an ampersand sign (&) in the following way:

(terminal 2): $ ./PythonUDPServer.py &

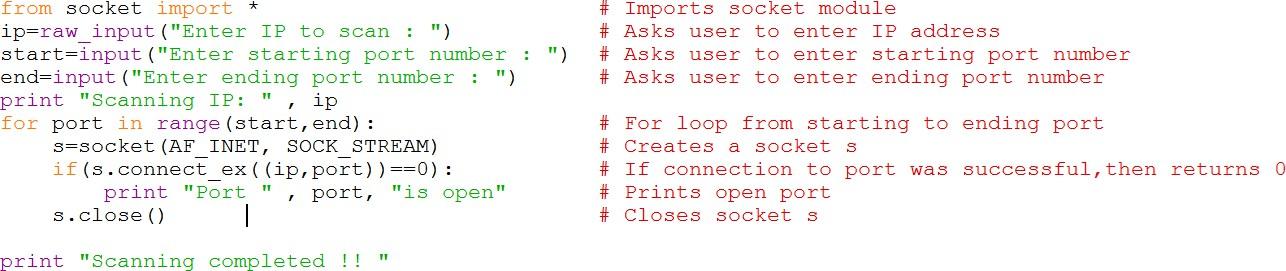
1. Compare the working of the command above with the command without the ampersand to know the difference between background process and foreground process.
2. Interact with this application, and explain how this application works. Come up with interesting variations of the original program and describe how your modified application works.



* + 1. **Creating a Python ‘port-scanner’ module and running it**

We’re going to create a basic Python port scanner:

1. To give you practice of constructing Python code, and to get indentation right, please type the following into a new file and have it as a Python module titled: ‘portscanner.py’.
2. You can use Python-IDLE to create this new file (File -> New Window) and to run the file (Run -> Run Module or simply type F5)



1. Run the portscanner and discover the listening ports on your machine. *Provide a screenshot of your results.*



* + 1. **Simple ‘daytime’ TCP client/server setup, compilation and execution in C**

In daytime client server, clients send the request to the server to send the date, day and time of the execution of the program and the server respond to the request of client accept the connection and send the services required by the client.

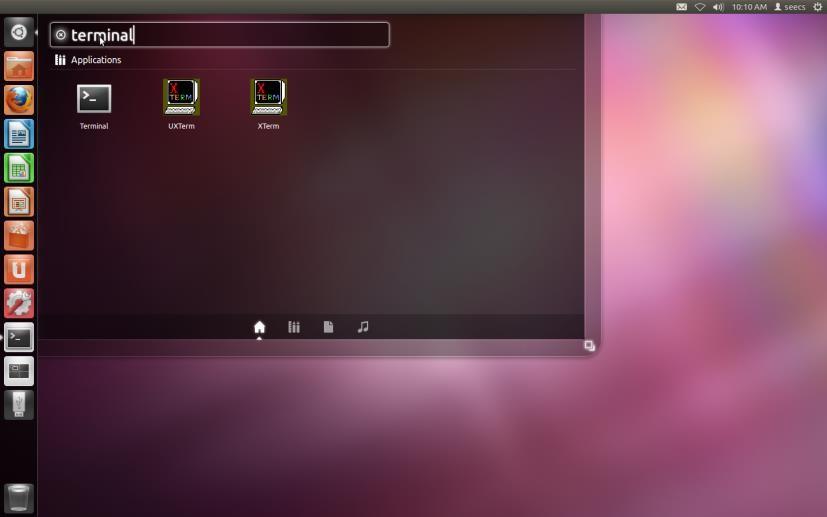
1. Startup the operating system Ubuntu, go to search and type terminal as shown in the [Figure 6.3.](#_uuvkjsl889ce)

Figure 6.3: Ubuntu terminal opening

Terminal is just like as a command prompt in windows. After opening terminal your desktop looks like just in [Figure 6.4:](#_iddbnnk94ueq)



Figure 6.4: Terminal window

1. As we are implementing client server application, we now have to create the client file and a server file. Before creating client file change the directory to desktop by writing cd Desktop as shown in [Figure 6.5.](#_rro43aymhb5u)



Figure 6.5: Changing directory

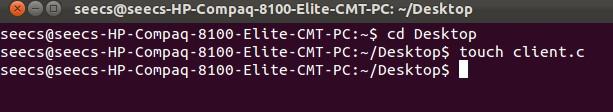
1. After changing directory, create an empty file client.c by issuing the command ‘touch client.c’ at the terminal.

Figure 6.6: Creating client file

1. After creating client file open and edit client file for writing and updating the client by double click on the client.c file, a new window will be open as in [Figure 6.7.](#_x80bdvpdu464)

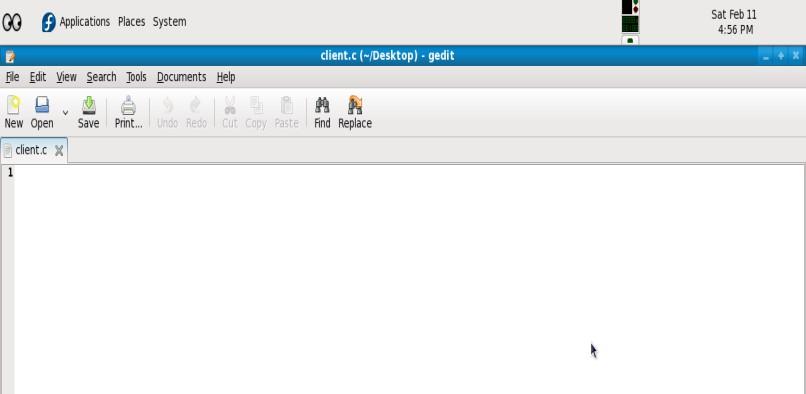


Figure 6.7: Editing client file

1. Now for creating server file repeat the above steps 1 to 4 as we did for client side with the name server.c
2. Now open the client file as in step 4 and write the program as in [Figure 6.8.](#_79flvq2zem5z)

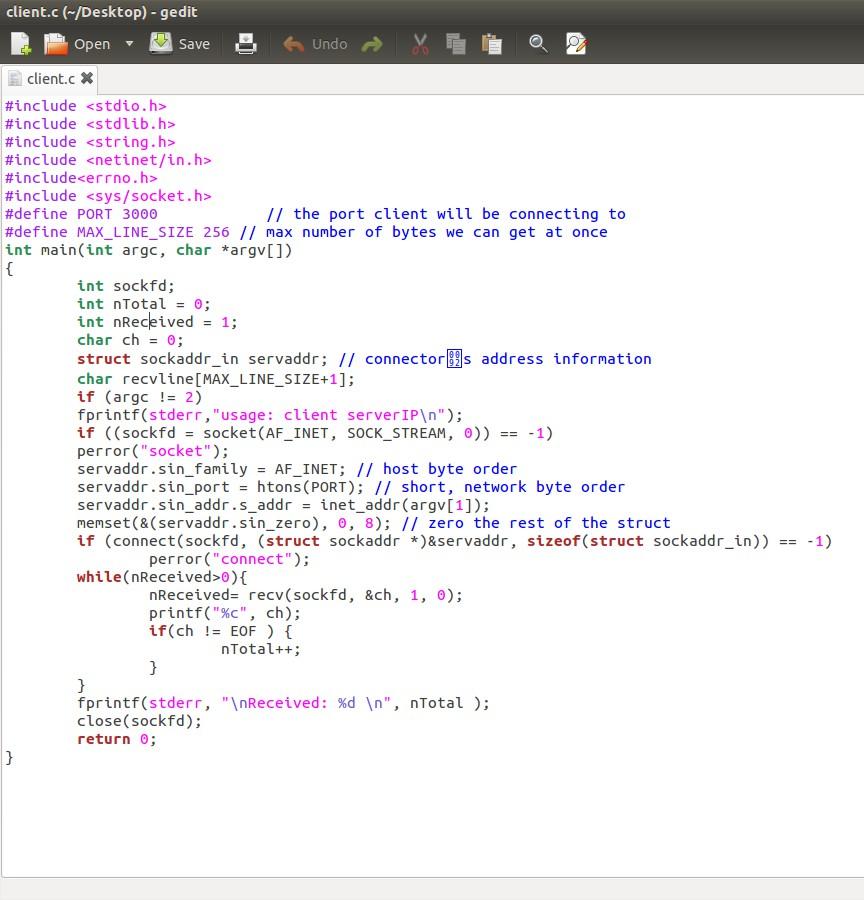


Figure 6.8: Client side of daytime program

1. Now open the server file and write the program as in [Figure 6.9.](#_7fovs7gsv9d7)

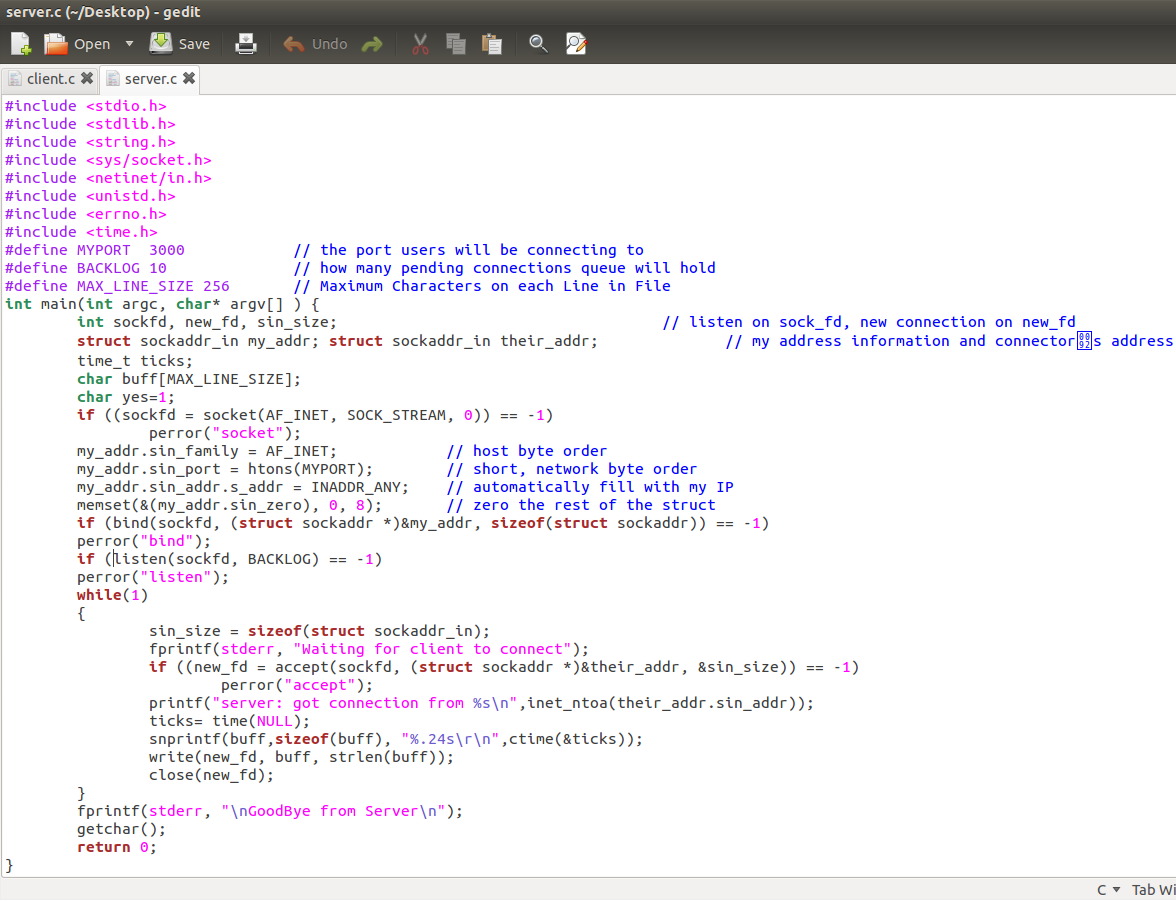


Figure 6.9: Server side of daytime program

1. After writing client and server program, it’s time to compile our program. For this we use a command gcc –o c client.c this will make a file name c which compile the client.c file and returns errors if the program would have some errors as shown in [Figure 6.10.](#_v0jl5isa47kh) It will shows that in which line there is an error. Then open the client.c file for observing the erroneous lines and correct it after that compilation of the file will give the result as in [Figure 6.11.](#_6mk3yw6n1f7h)

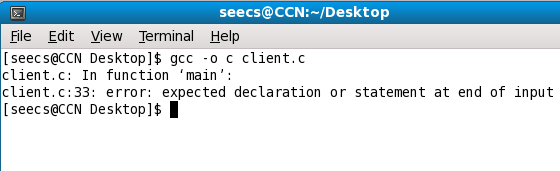


Figure 6.10: Client file showing errors

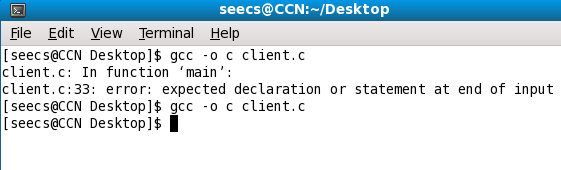


Figure 6.11: Client file after removing errors

1. After compiling the client and server sides now it’s the time to run and check the client server communication. As we know that the server is waiting for the client to connect so we run the server first on one terminal ,write ./s in the terminal after changing directory to desktop as the files are placed by us on the desktop of the operating system. The server will be blocked in accept function as shown in [Figure 6.12.](#_lo5bz5q0xn2v)

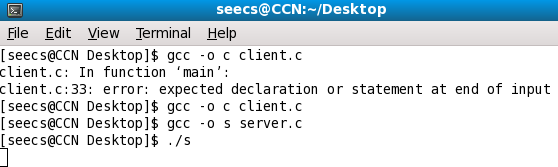


Figure 6.12: Running server

1. Last but not least, open terminal and after changing directory run the client by writing ./c 127.0.0.1 . Here this ip address 127.0.0.1 is the loopback Internet protocol (IP) address also referred to as the “localhost”. The address is used to establish an IP connection to the same machine or computer being used by the end-user. This step shows that the client want to connect to the server and asking for the day time and date. After connecting the server entertain the client and send the time date and day, the client read it and display it as shown in [Figure 6.13.](#_275t5wabb6t3) After serving to this client, the server will again go to block state and will wait for the other clients to come and ask for date time and day. The server gives its services to all the clients iteratively in this program as shown in [Figure 6.14.](#_hrlc3occwuxj)



Figure 6.13: Client showing day time and date

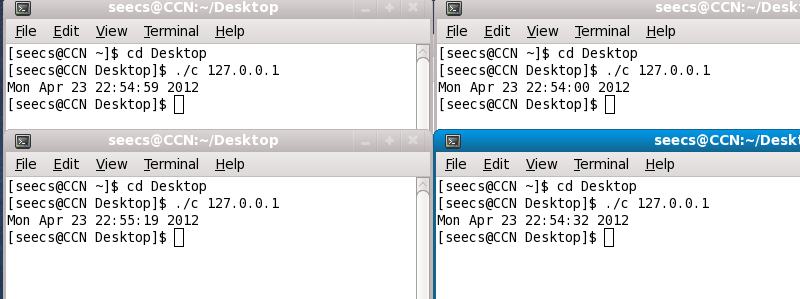


Figure 6.14: Server serves to the clients iteratively

* + 1. **Parameterized ‘daytime’ TCP Client/Server Setup, Compilation and Execution in C**

Modify the program written in [2.4.3](#_3vwyadcu642r) such that it is parameterized by the options provided when running this program.

We would like to server program to take the port it runs on as a command-line-argument. Similarly, we would like the client program to take the IP address and port (of the server machine and process, respectively) as a command line argument.

The client program (let’s say executable file c) will be run as follows with IP defining the IP address of server and PORT defining the port of the server process:

# ./c IP PORT

The server program (let’s say executable file s) will be run as follows with PORT defining the port of the server process:

# ./c IP PORT

The client and server should do error checking and give an error for incorrect number of arguments. For example, if the client is provided no arguments or just one argument, it should give an IP address and give an error message:

Incorrect numbers of arguments provided. Correct usage: ./c IP port

For this task make a group of two students, one student’s program should run as the client part and the other student’s program should run as the server process. The client process should be initiated after the server process is running in the following way:

./c IP PORT where IP is the IP address of the server machine (for example ./c 10.3.33.148) and PORT is the port the server process is running on.

Please note that for this part of the lab exercise you would NOT be using loopback address (127.0.0.1) for the server address since the client is not connecting to the server on the same machine but to a server on a different machine. The IP address of that machine should be provided as a command line argument.

*Please provide the code that you wrote for this part in the report here.*



**Hints:**

* You can find help online about how to read in command line arguments provided to a C program by reading about argc and argv defined to be input parameters for the main function of the C file.

The integer argc describes the count of arguments provided. (The name of the file is considered as the first argument while latter options follow as the second argument, and so on.). The array argv[] contains the arguments provided; argv[0] would contain the name of the executable file being run, argv[1] would be the first argument, and so on.

Read more at: <http://crasseux.com/books/ctutorial/argc-and-argv.html>

* The IP address and ports that are provided to the C program as command line argument would be read as strings (and not as integers). You will need to find a way to convert these strings into integers. You can find help on this by searching for how to convert ASCII to Integers in C.
* To output error, you can use the perror function which is also used in Lab task 3. You can also use fprintf and print to stderr file (which translates to the monitor typically).
* To get help, you can use the man command in the linux shell to access the manual pages. To access the pages of systems calls, you can type #man 2 call. Here 2 refers to the section of man pages. You can find the man pages (which you can access in the terminal on any UNIX machine) online by visiting: [http://www.linuxmanpages.com/.](http://www.linuxmanpages.com/) You can find the section 2 of the man pages (describing the system calls) at: <http://www.linuxmanpages.com/man2/>
* You can find help in general on socket programming by looking up Beej’s guide to Network Programming online. A link has been provided on the lab site. A definitive resource on UNIX networking is the book by Richard Stevens titled: “Unix Network Programming.”

### Assessment Rubrics for EE432: Computer Networks Lab 6

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 7 Internet Protocol (IP)

| **Name** | **Roll. No.** | **Total Marks** | **Obtained Marks** | **Viva Marks** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Checked on:

Signature:

64

 **Objectives**

In this lab, we’ll investigate the IP protocol, focusing on the IP datagram. We’ll do so by analyzing a trace of IP datagrams sent and received by an execution of the traceroute program. We’ll investigate the various fields in the IP datagram, and study IP fragmentation in detail.

We will also discuss IP address and discover its hierarchical nature. We will also discover the usage of subnet masks to specify the network portion of an IP address.

 **Instructions**

1. Read carefully before starting the lab.
2. These exercises are to be done individually.
3. You are supposed to provide the answers to the questions listed at the end of this document and upload the completed report to your course’s google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. Complete the lab half an hour before the lab ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

 **Background**

* + 1. **IP Addressing**

In this part, we are going to study IPv4 addressing format in which we use 32 bits to identify a host that connects to a TCP/IP network. The IPv4 address serves to standardize the logical address that is understood by all TCP/IP nodes thereby hiding the underlying heterogeneity that is a characteristic of link-layer networks.

IPv4 address is a 32 bit address containing some bits to identify the network and the remaining bits to identify a host in this network. It was decided earlier that we would have a few classes of IPv4 address according to which these 32 bits would be subdivided into network and hosts bits. These classes worked in the following way:

* + - * Class A: 8 bits for the network and the remaining 24 bits for identifying the host.
      * Class B: 16 bits for the network and the remaining 16 bits for identifying the host.
      * Class C: 24 bits for the network and the remaining 8 bits for identifying the host.
      * Class D was specified as a multicast class while another class, Class E was specified as an experimental class.

This simple division of IP addresses into these classes proved useful in the beginning due to its simplicity. However, such a scheme proved inflexible for the growing demands of the Internet because it led to wastage of addresses. For example, a point-to-point serial link requires only a network but it only requires for two nodes to be addressed. Using the class that has least number of host bits (Class C) still led to a lot of wastage of IP addresses---a class C IP network can be used to give addresses of 254 hosts (2 addresses of the total 2^8 addresses are reserved to refer to the network itself and to specify broadcast implying that all the network nodes should process the packet).

The next logical progress was variable length subnet masking in which instead of fixed classful addressing, we have a subnet mask in which we have more flexibility in defining the number of bits of the total 32 bits of IPv4 address that are used to identify the network. It must be pointed out that in an IP address, the network bits are contiguous and occur at the start (MSB) of an IP address. A subnet mask is a 32 bit number in which the leading bits are set to 1 commensurate to the number of network bits. For example, if 8 bits of an IP address is used to specify the network, and the remaining bits for specifying the host, then a subnet mask of 255.0.0.0 would be used (according to the decimal dotted notation explained next).

Decimal dotted notation: Since an IPv4 addresses (and subnet mask) are composed of 32 bits, it is easier to deal with 4 chunks of octets rather than with the 32 bits directly. In addition, each chunk comprising of 8 bits is represented by its decimal equivalent in the following way: 10000000 would be represented as 128. The IP address 10000000 00000000 00000000 00000001 would be represented as 128.0.0.1. Similarly, subnet masks are also described in decimal dotted

notation. Subnet masks specify the demarcation between network portion and host portion, since the network bits are contiguous and occur at the leading side of an IP address, subnet masks would only have octets of the following form:

| 11111111: | 255 |
| --- | --- |
| 11111110: | 254 |
| 11111100: | 252 |
| 11111000: | 248 |
| 11110000: | 240 |
| 11100000: | 224 |
| 11000000: | 192 |
| 10000000: | 128 |
| 00000000: | 0 |

As an example, a subnet mask of 255.192.0.0 would imply an IP address in which the first 10 bits specify the network part of the IP address and the remaining bits (22 bits) specify the host bits.

**7.3.1.1. Questions**

1. Determine the IP address of your lab machine using the ipconfig command. Also, write the subnet mask configured?
2. What is the network address of the network to which your machine belongs?

*Hint: you can determine this by setting all the host bits of your IP address to 0 and obtaining the decimal dotted equivalent of the IP address; For example, the machine having IP address 10.128.232.21 having the subnet mask 255.128.0.0 belongs to the network 10.128.0.0.*



1. Is the IP address being configured on your machine statically or dynamically? (Note, in dynamic configuration, your machine would act as the client of a DHCP server to obtain IP address automatically)



1. Is the IP address on your machine a private IP or a public IP?

*(Please note that RFC 1918 describes the three IP network classes that have been dedicated for private usage; you can download this very short RFC through the URL:* [*http://tools.ietf.org/html/rfc1918*](http://tools.ietf.org/html/rfc1918)*)*



1. If your answer to the question above is a private IP, then which IP network class (of those three defined in RFC 1918) does the IP address configured on your machine belongs to?



1. Write down all the three IP network classes that are private as defined by RFC 1918.
2. Note the IP address of the default gateway? Assuming the same subnet mask as is configured for your machine, determine the network address of this IP. Describe how it corresponds to the network address determined in question 2.



* + 1. **IP packet and IP fragmentation**

In order to generate a trace of IP datagrams for this lab, we’ll use the *traceroute* program to send datagrams of different sizes towards some destination, X. Recall that traceroute operates by first sending one or more datagrams with the time-to- live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; it then sends a series of datagrams towards the same destination with a TTL value of 3; and so on. Recall that a router must decrement the TTL in each received datagram by 1 (actually, RFC 791 says that the router must decrement the TTL by at least one). If the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) to the sending host. As a result of this behavior, a datagram with a TTL of 1 (sent by the host executing traceroute) will cause the router one hop away from the sender to send an ICMP TTL-exceeded message back to the sender; the datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the datagram sent with a TTL of 3 will cause the router three hops away to send an ICMP message back to the sender; and so on. In this

manner, the host executing traceroute can learn the identities of the routers between itself and destination X by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages.

We’ll want to run traceroute and have it send datagrams of various lengths.

The tracert program provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the tracert program. A nicer Windows traceroute program is pingplotter, available both in free version and shareware versions at [http://www.pingplotter.com.](http://www.pingplotter.com/) Download and install pingplotter, and test it out by performing a few traceroutes to your favorite sites. The size of the ICMP echo request message can be explicitly set in pingplotter by selecting the menu item Edit-> Options->Packet Options and then filling in the Packet Size field.

The default packet size is 56 bytes. Once pingplotter has sent a series of packets with the increasing TTL values, it restarts the sending process again with a TTL of 1, after waiting Trace Interval amount of time. The value of Trace Interval and the number of intervals can be explicitly set in pingplotter.

 **Procedure**

1. Startup Wireshark and begin packet capture (Capture->Start) and then press OK on the Wireshark Packet Capture Options screen (we’ll not need to select any options here).
2. If you are using a Windows platform, startup pingplotter and enter the name of a target destination in the “Address to Trace Window.” Enter 3 in the “# of times to Trace” field, so you don’t gather too much data. Select the menu item Edit- >Advanced Options->Packet Options and enter a value of 56 in the Packet Size field and then press OK. Then press the Trace button. You should see a pingplotter window that looks something like this:

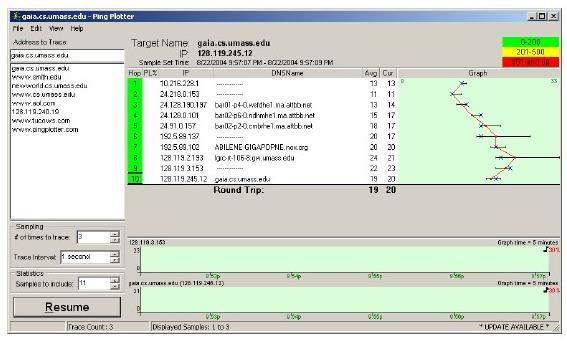


Figure 7.1: Pingplotter window

1. Next, send a set of datagrams with a longer length, by selecting *Edit->Advanced Options->Packet Options* and enter a value of 2000 in the Packet Size field and then press OK. Then press the Resume button.
2. Finally, send a set of datagrams with a longer length, by selecting *Edit->Advanced Options->Packet Options* and enter a value of 3500 in the Packet Size field and then press OK. Then press the Resume button.
3. Stop Wireshark tracing.
4. If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the book’s authors’ Windows computers. You may well find it

valuable to download this trace even if you’ve captured your own trace and use it, as well as your own trace, when you explore the questions below.

1. In your trace, you should be able to see the series of ICMP Echo Request (in the case of Windows machine) or the UDP segment (in the case of Unix) sent by your computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers.
2. In the questions below, it is assumed that you are using a Windows machine. Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet, use File->Print, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.
   * 1. **Questions**
3. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?



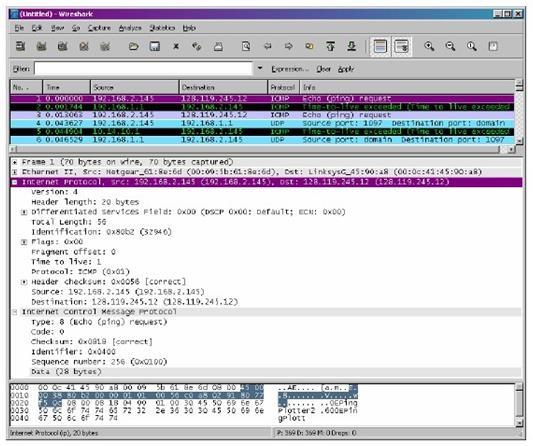


Figure 7.2

1. Within the IP packet header, what is the value in the upper layer protocol field?



1. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.



1. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.



Next, sort the traced packets according to IP source address by clicking on the Source column header; a small downward pointing arrow should appear next to the word Source. If the arrow points up, click on the Source column header again. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol portion in the “details of selected packet header” window. In the “listing of captured packets” window, you should see all of the subsequent ICMP messages (perhaps with additional interspersed packets sent my other protocols running on your computer) below this first ICMP. Use the down arrow to move through the ICMP messages sent by your computer.

1. Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?



1. Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?



1. Describe the pattern you see in the values in the Identification field of the IP datagram.

Next (with the packets still sorted by source address) find the series of ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router.

1. What is the value in the Identification field and the TTL field?
2. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest first hop) router? Why?



* + 1. **Fragmentation**

1. Sort the packet listing according to time again by clicking on the Time column.
2. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram?

*[Note: if you find your packet has not been fragmented, you should download the zip file* [*http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip*](http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip) *and extract the ip-ethereal-trace-1packet trace. If your computer has an Ethernet interface, a packet size of 2000 should cause fragmentation]*



1. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment?



1. How long is this IP datagram?
2. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?



1. What fields change in the IP header between the first and second fragment?

### Assessment Rubrics for EE432: Computer Networks Lab 7

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 8 MySQL: Relational Database Management System

| **Name** | **Roll. No.** | **Total Marks** | **Obtained Marks** | **Viva Marks** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

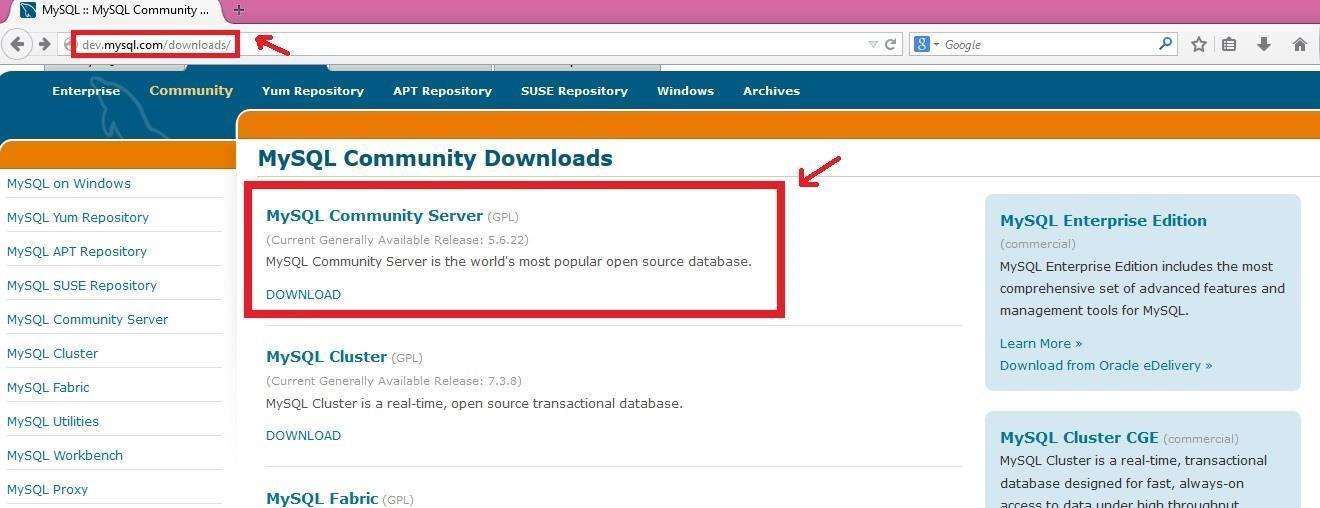
Checked on:

Signature:

EE432 Computer Networks

8.1.

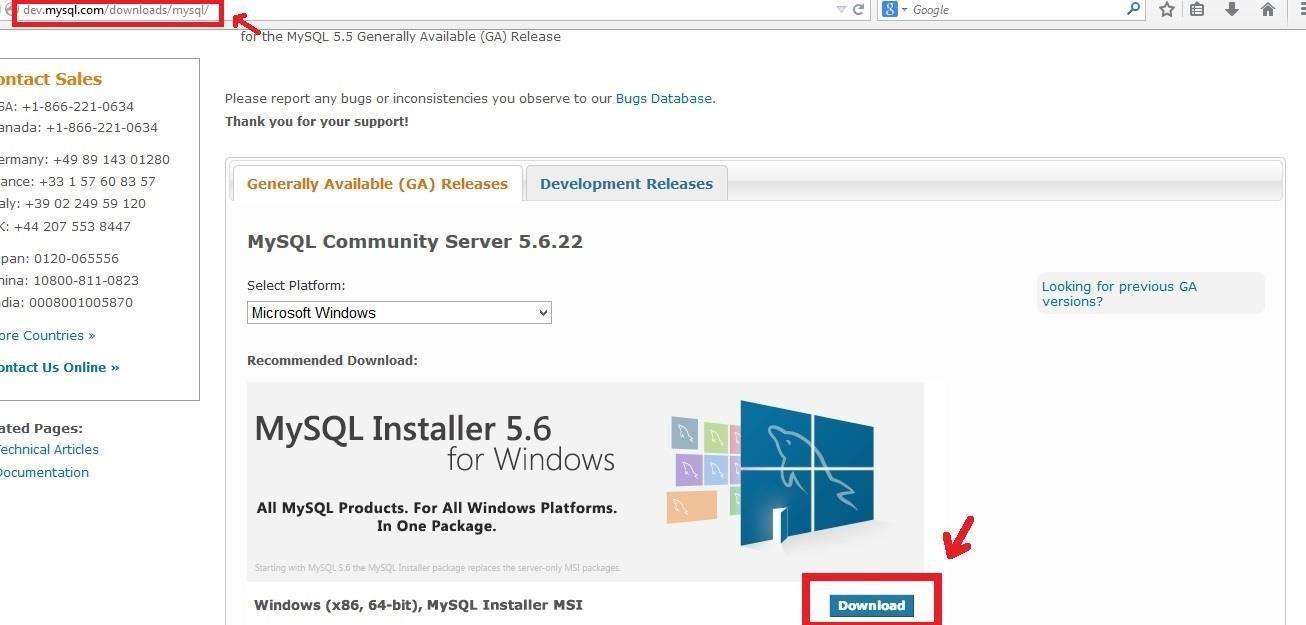
In order to download MySQL installer, go to [dev.mysql.com/downloads](http://dev.mysql.com/downloads) and click on MySQL Community Server.



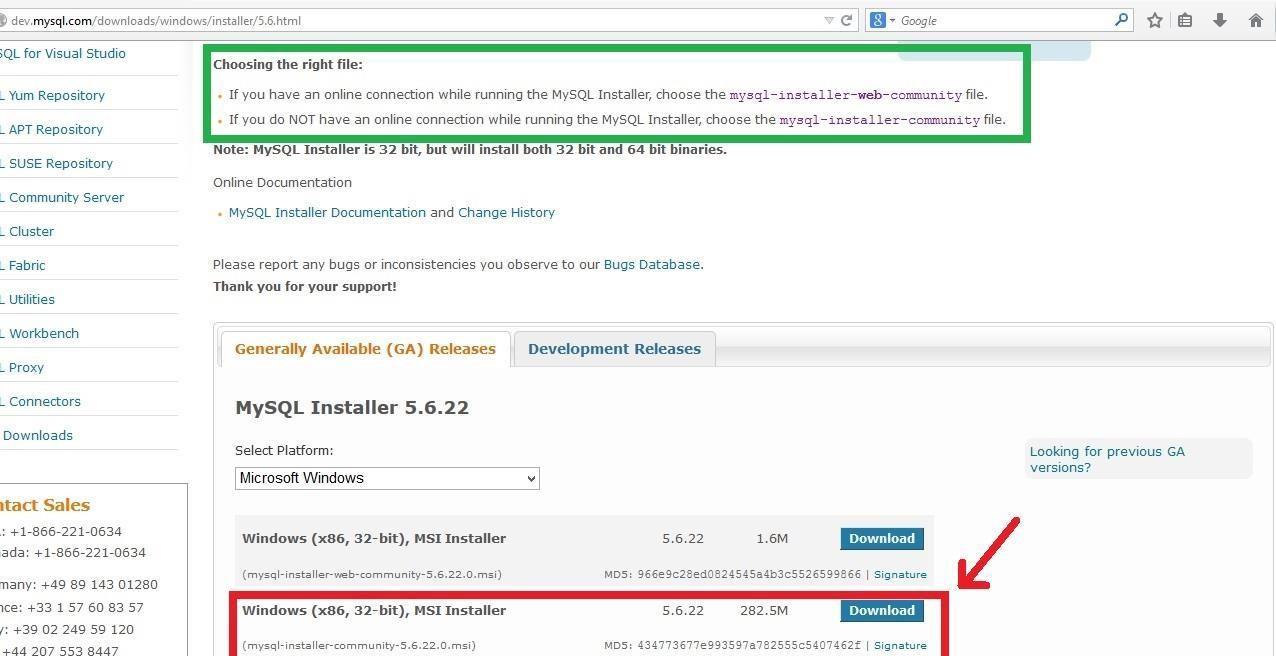
8.2

Scroll the newly opened page down and click on **download** written in front of the MSI package as

shown in the screenshot below:



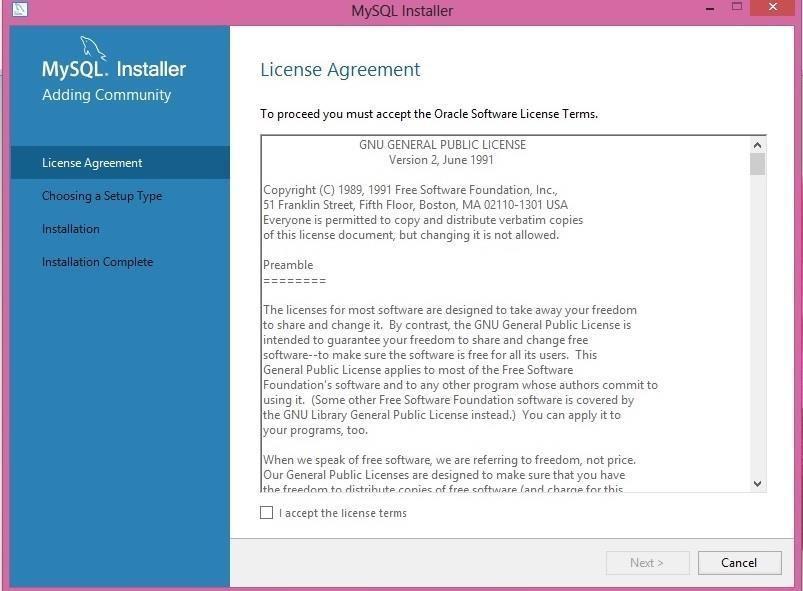
1. You will be directed to a new page on which download **the mysql-installer-community** file.



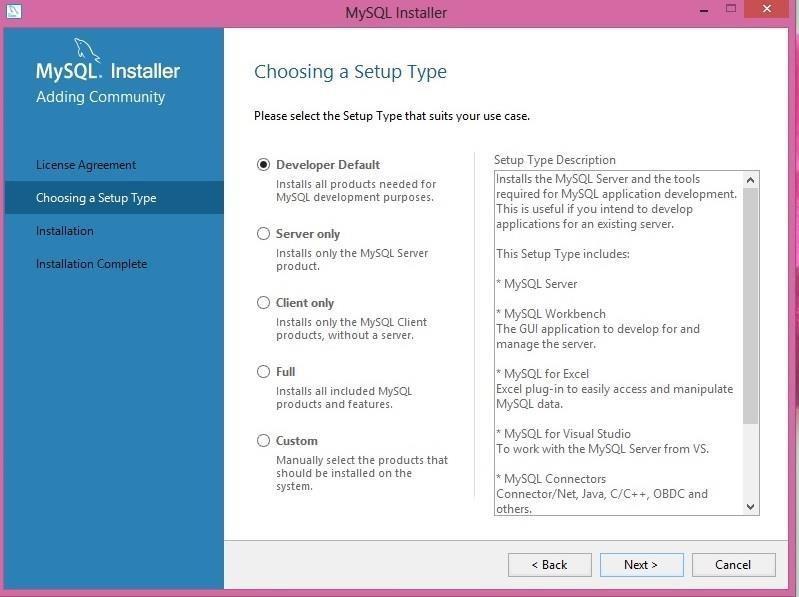
1. Double click on the downloaded MSI file and allow it to execute.



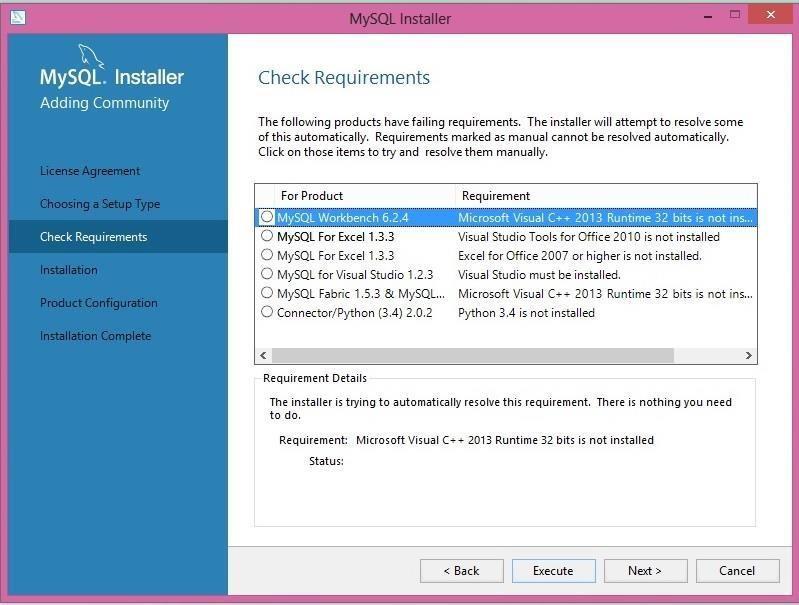
1. After the initial configuration of the installer, you will see the following screen. Accept the license terms (after reading of course☺) and click next.



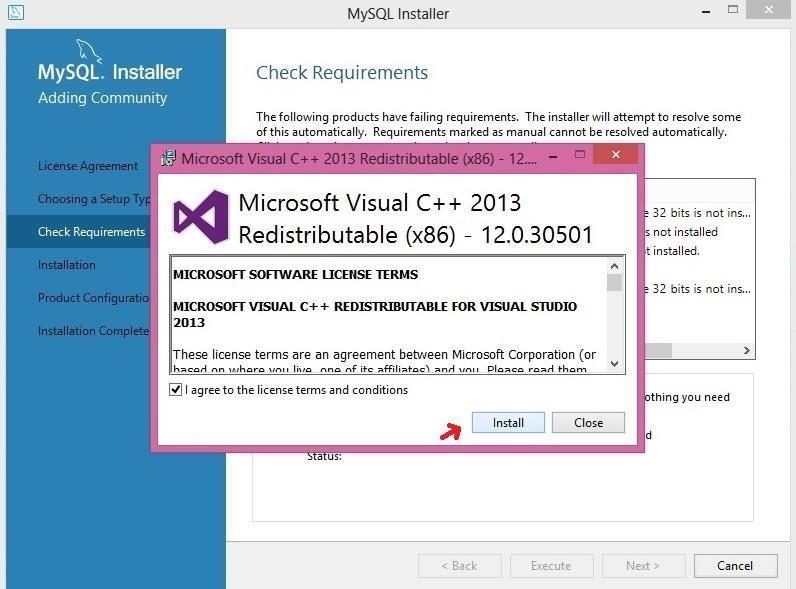
1. To install MySQL server and other MySQL tools related to MySQL development, choose Developer Default. Otherwise, choose the Custom setup type to manually select your desired MySQL products i.e., server and workbench. Click next.

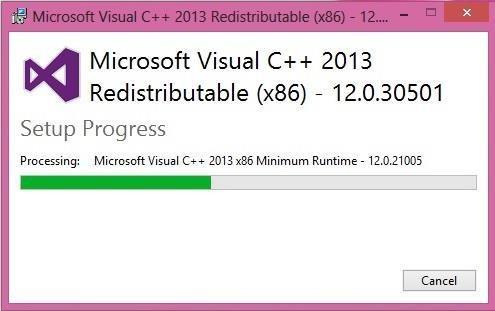


1. Some products might have some failing requirements. From these products, to resolve requirements, manual updates are required for some products and for others, the installer will automatically resolve issues. Click execute. Once the installer is done, manually resolve the other issues that are left. For this lab, the tools are not needed for which the manual set up is required so they can be ignored.

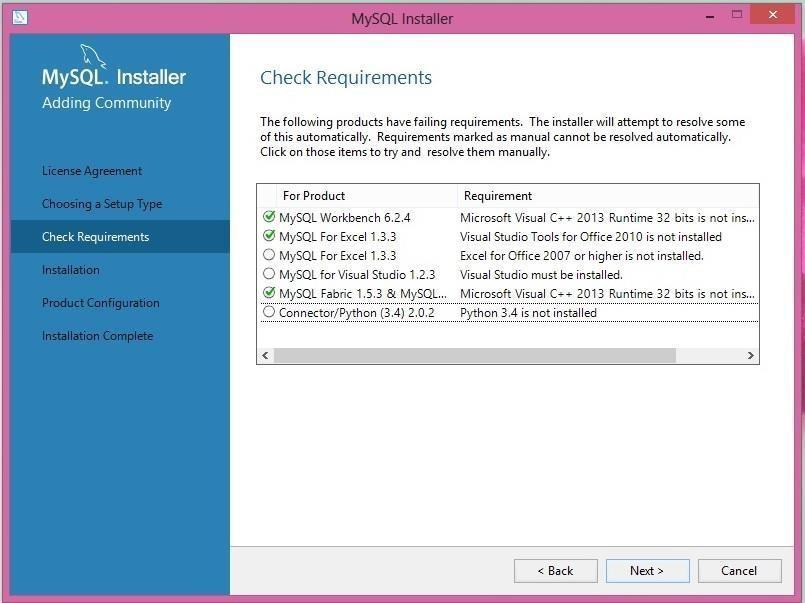


1. While resolving the requirements for mysql workbench, you will come across the following window. Agree to the license terms and conditions and install it.

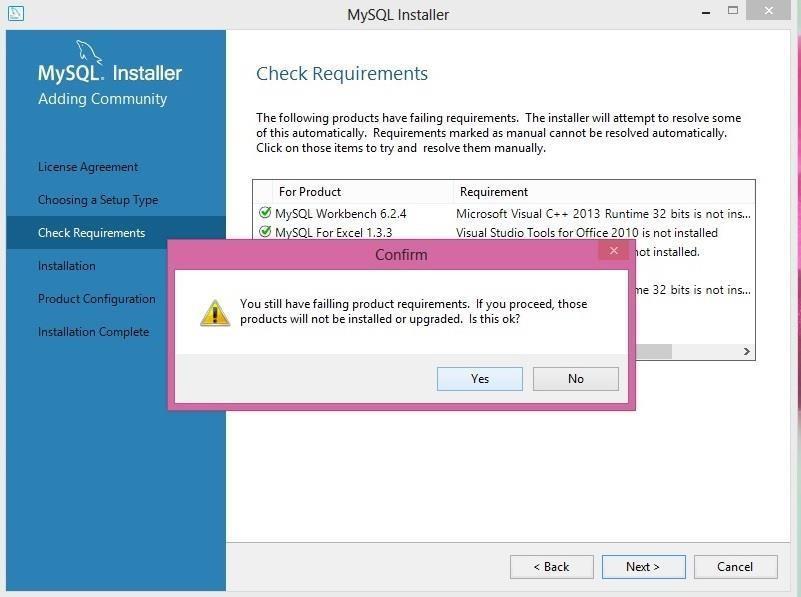




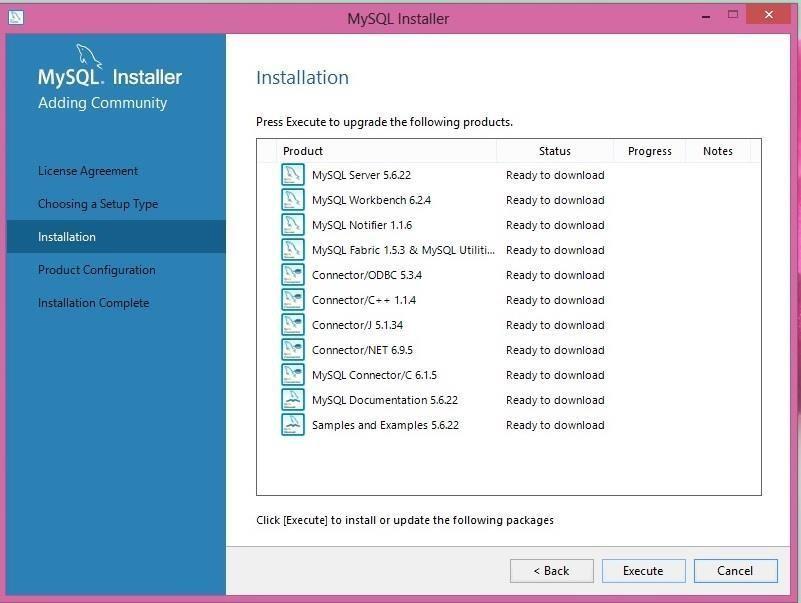
1. Similarly, you will encounter other prompts for the tools that are required for other products. Let the installer install whatever it asks for. Once done, click next.



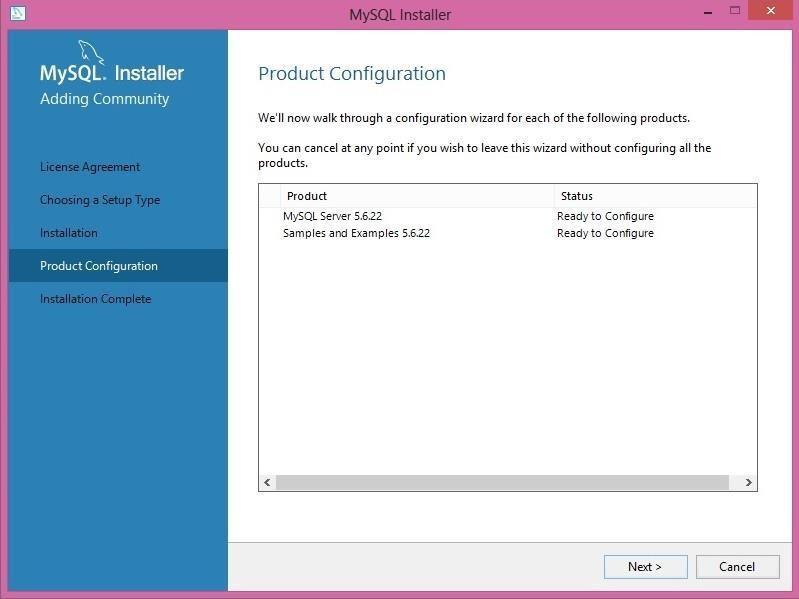
1. The installer will inquire about the failing product requirements. For this lab, those products are not required so you can ignore them and press yes.



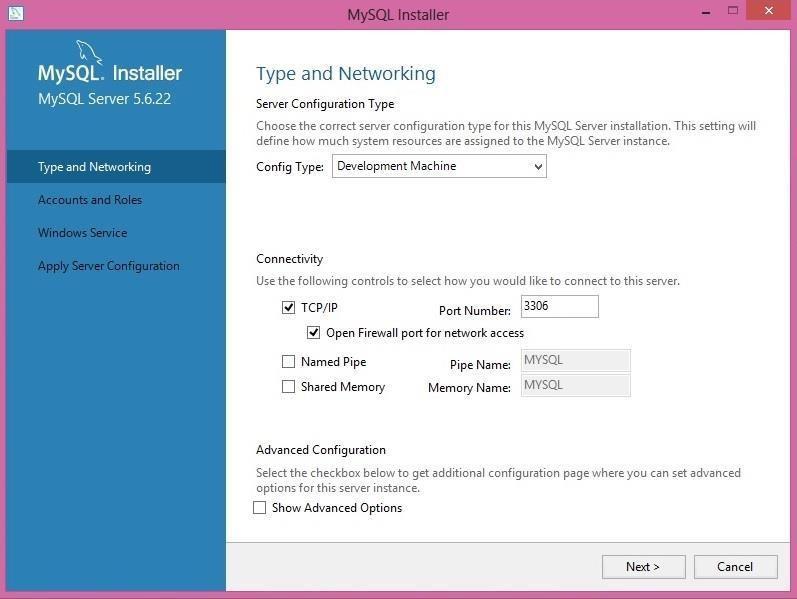
1. The following screen will appear. Press execute and the installer will start downloading all the products that are ready to be downloaded.

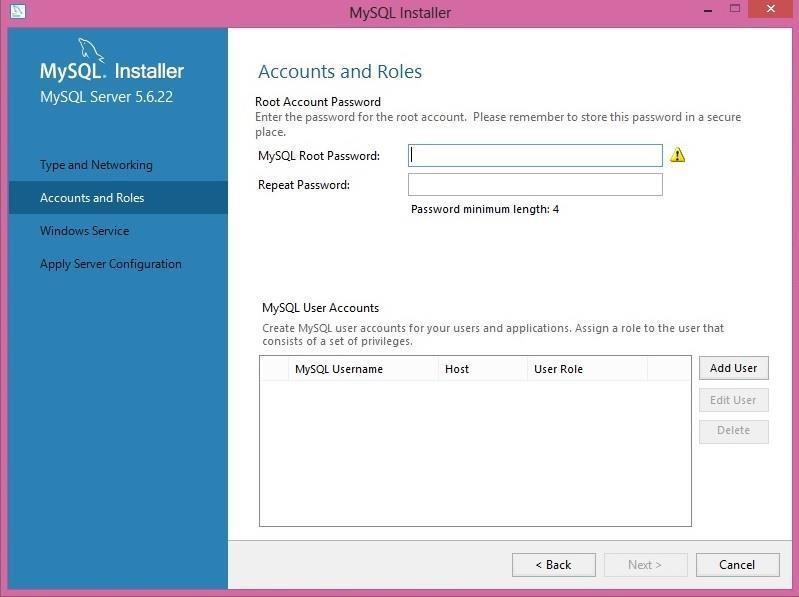


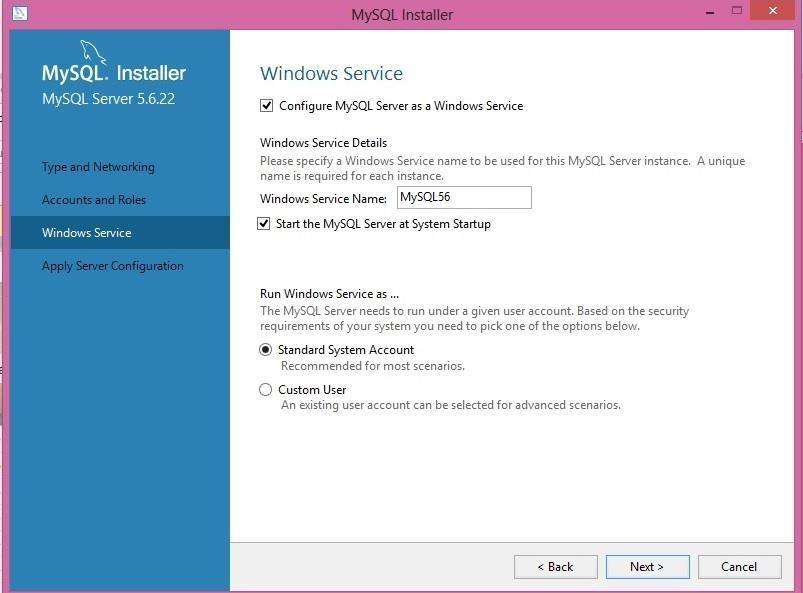
1. Once done, click on next and the following screen will appear. Press next so that the required products can be configured.



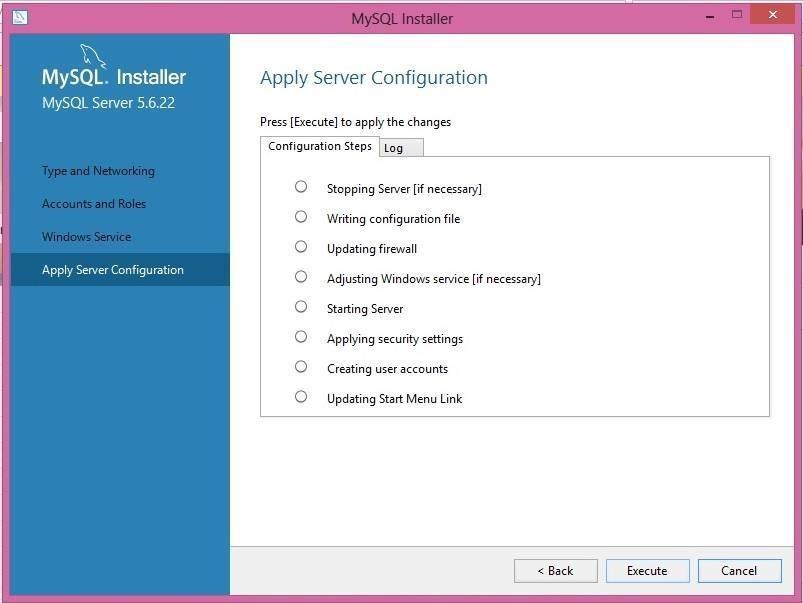
1. Keep the default settings and click next.



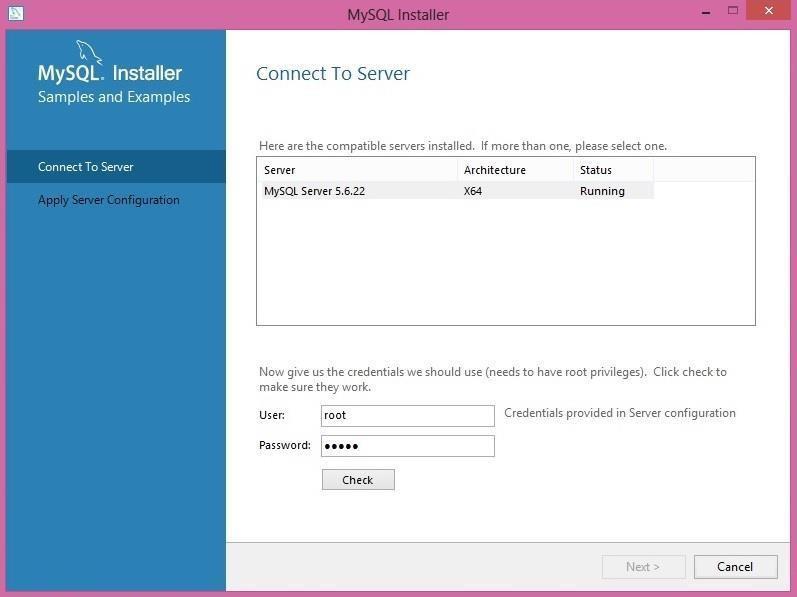
1. Choose a password for the root account that would be required to connect to the server, and click next.
2. Keep the settings as they are and click next.



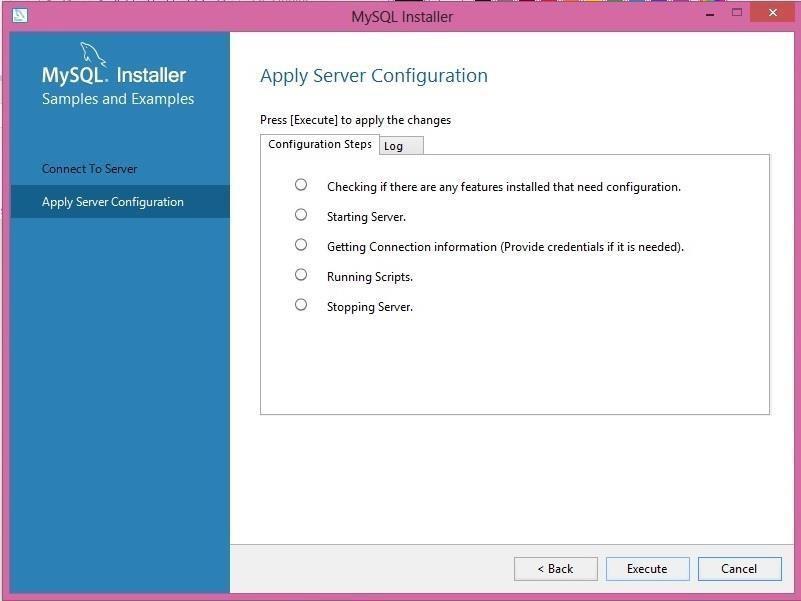
1. Click execute so that the installer can start the server configuration. When done, press finish.



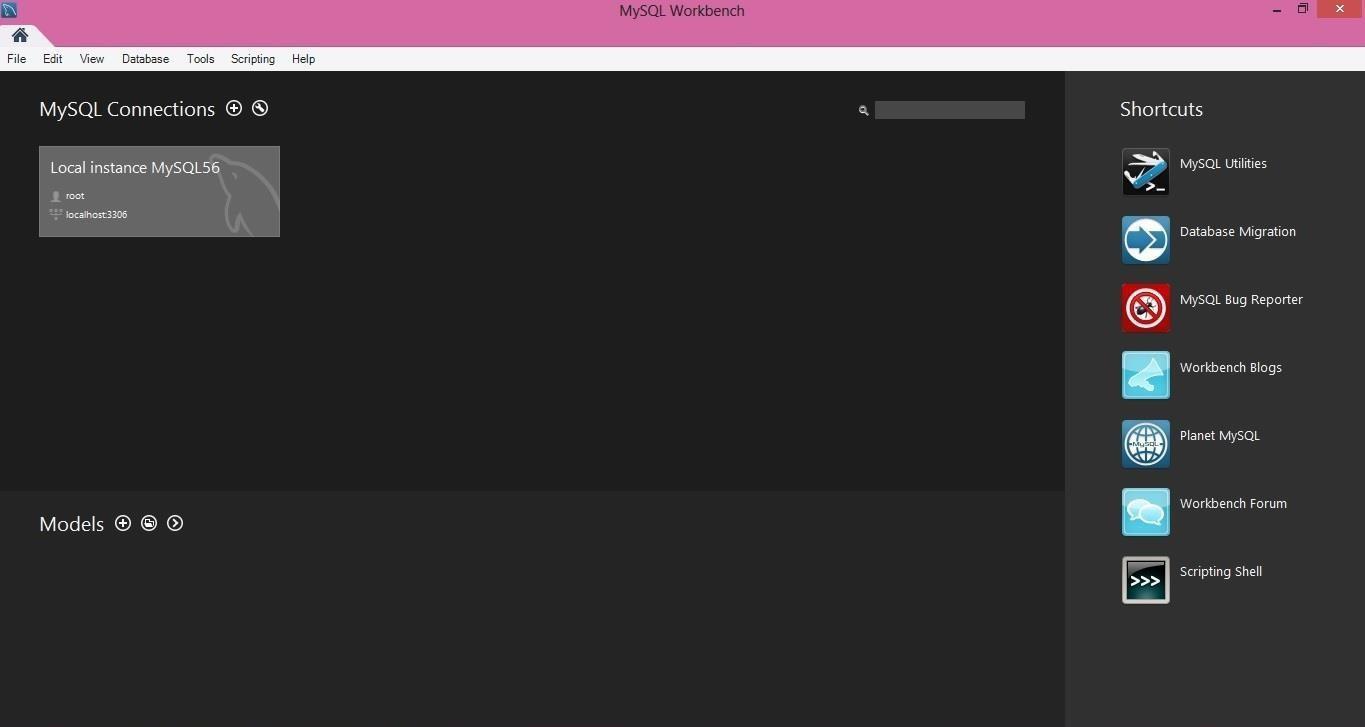
1. Now, to connect to the server, enter the root password and click check. Once the connection is successful, press next.



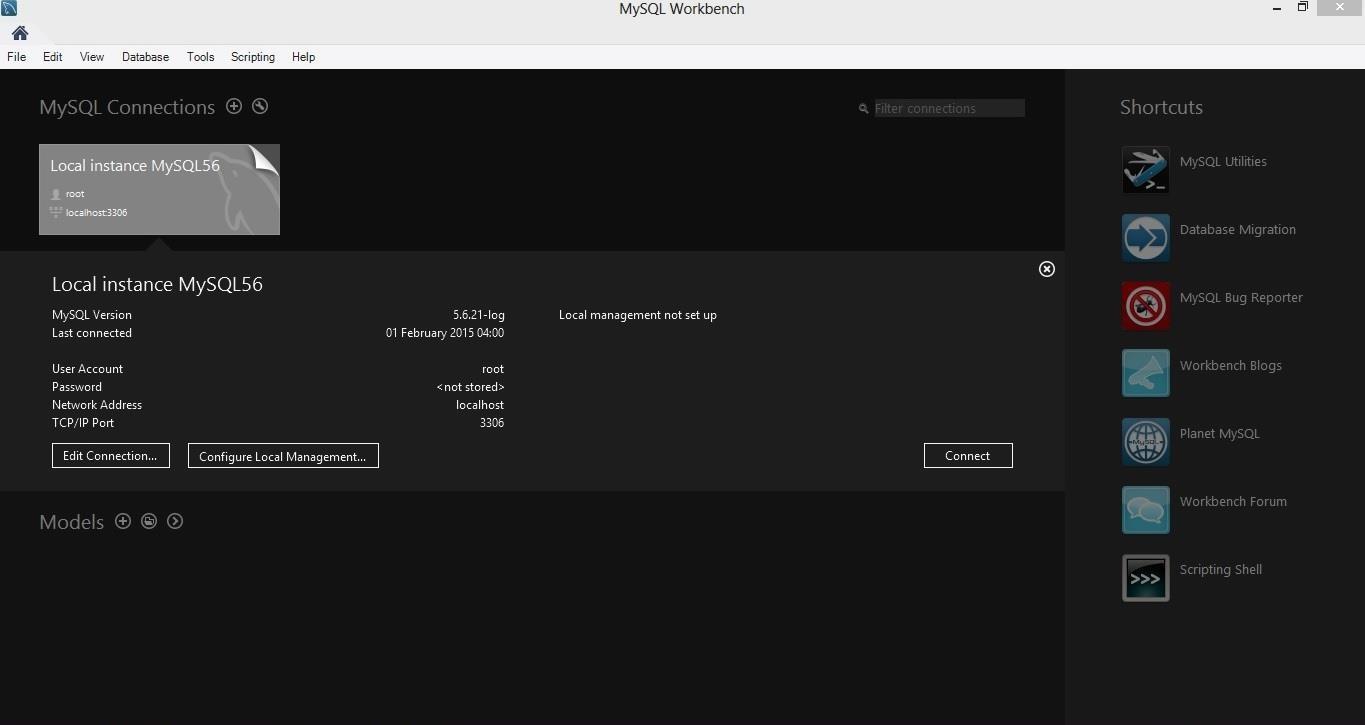
1. Click on execute to apply changes.

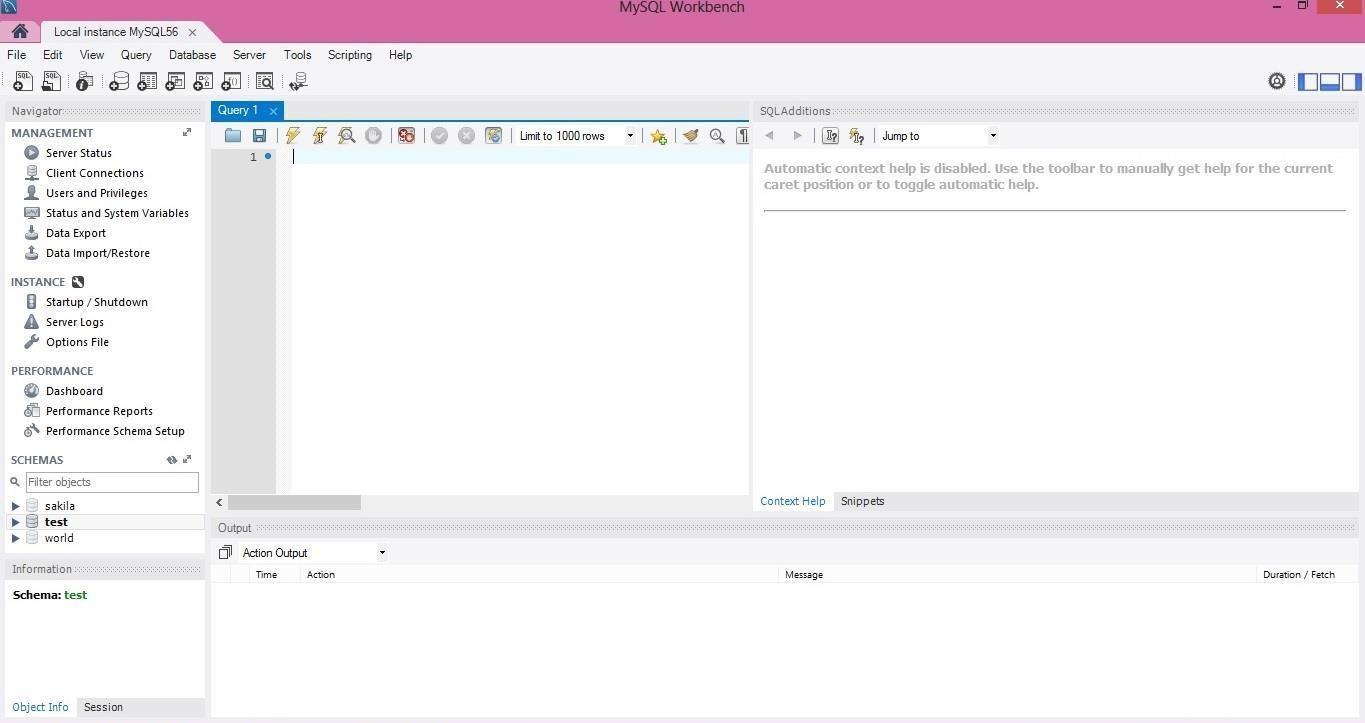


1. Once installation and configuration processes are done, click finish and launch the MySQL workbench.
2. The following screen will appear. Double click on the local instance.



1. Connect to the local instance and enter the root password that you set during the installation process.



1. Finally, you are connected to the server. Now, you can draw ER diagrams, create tables, and write and execute queries using the workbench.

### Assessment Rubrics for EE432: Computer Networks Lab 8

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of**  **experiment (a)** | Downloads and installs required software and sets up  the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and  unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the  software application rigorously | Needs assistance or guidance to proceed through experiment steps,  studies outputs with minor errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and disciplinary**  **rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps,  ensures that the data is entered in the lab manual according to | Completes data collection with minor error and enters data in lab manual with  slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of  experiment setup, unable to fill the lab manual properly |  |

|  | the specified instructions |  |  |  |
| --- | --- | --- | --- | --- |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 9 Data Import and Export in MySQL Workbench

| **Name** | **Roll. No.** | **Total Marks** | **Obtained Marks** | **Viva Marks** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

* 1. Task 1

Show all steps and attach the files that were used to import data and that were populated after exporting data from database

1. Create a new Database with all the tables as mentioned below:
   * + HOTEL (HotelNo, hotelName, city)
     + GUEST (GuestNo, guestName, guestAddress)
     + ROOM (RoomNo, HotelNo, type, price)
     + BOOKING (HotelNo, GuestNo, DateFrom, DateTo, RoomNo) INT: HotelNo, guestNo, RoomNo

FLOAT: Price

DateTime: DateFrom, DateTo

Varchar: HotelName, City, guestName, guestAddress, Type

1. Populate each table with at least 3 rows by importing data from the CSV file
2. Transfer the data of at least 2 tables from DB to CSV files
   1. Task 2

Introduction to DDL-DML and Constraints

Data Definition Language (DDL) statements are used to define the database structure or schema. Some examples:

* CREATE - to create objects in the database
* ALTER - alters the structure of the database
* DROP - deletes objects from the database
* TRUNCATE - removes all records from a table, including all spaces allocated for the records are removed Data Manipulation Language (DML) statements are used for managing data within schema objects. Some examples:
* SELECT - retrieves data from the a database
* INSERT - inserts data into a table
* UPDATE - updates existing data within a table
* DELETE - deletes records from a table, the space for the records remain Data Control Language (DCL) statements. Some examples:
* GRANT - gives user's access privileges to database
* REVOKE - withdraws access privileges given with the GRANT command

**Error! Reference source not found.**

Transaction Control (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

* COMMIT - saves work done
* SAVEPOINT - identifies a point in a transaction to which you can later roll back
* ROLLBACK - restores database to original since the last COMMIT
  1. Task 3

1. Create Table Departments with column DepartmentID, DepartmentName, DeptHeadID with DepartmentID as primary key. Set the data types in accordance to the real time scenario.
2. Create Table Employees with column EmployeeID, EmployeeName with EmployeeID as primary key. Set the data types in accordance to the real time scenario.
3. Alter Table Departments by adding new column DepartmentCode.
4. Insert at least two records in both tables.
5. Develop foreign key relation between two tables.
6. Implement referential integrity constraint of Set Null on Delete Rule on above relationship.
7. Implement referential integrity constraint of Set Cascade on Update Rule on above relationship.

Note:

Please zip your report and CSV files and name the zipped folder with your roll no. Please send your zip folder.

### Assessment Rubrics for EE432: Computer Networks Lab 9

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

* 1. Ability to condut experiments as well as to analyze and interpret data
  2. Ability to adhere to safety and disciplinary rules
  3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the  software application rigorously | Needs assistance or guidance to proceed through experiment steps,  studies outputs with minor errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered in the lab manual according to  the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with  sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from  theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### WRITING BASIC SQL QUERIES

| **Name** | **Roll. No.** | **Total Marks** | **Obtained Marks** | **Viva Marks** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

* + - 1. **Procedure**

Consider the following database schema for this lab:

***Sailor (sid, sname, rating, age) Boat (bid, bname, color) Reserve (sid, bid, day)***

**Schema definition, table definition, and example instances of sailor, reserve, and boat are given in a separate SQL file. Write down the queries for executing the following examples in SQL notation on your SQL Workbench. Please remember that your queries should work in general for all valid instances not for just given sample database.**

* + - * 1. Find the names and ages of all sailors.
        2. Find all sailors with a rating above 7.
        3. Find the names of sailors who have reserved boat number 103.
        4. Find the sids of sailors who have reserved a red boat.
        5. Find the names of sailors who have reserved a red boat.
        6. Find the colors of boats reserved by Lubber.
        7. Find the names of sailors who have reserved at least one boat.
        8. Find the ages of sailors whose name begins and ends with B and has at least three characters.
        9. Find the names of sailors who have reserved a red or a green boat.
        10. Find all sids of sailors who have a rating of 10 or reserved boat 104.

**For better understanding, instead of executing all queries on the workbench, try writing them on a piece of paper first.**

**Error! Reference source not found.**

### Assessment Rubrics for EE432: Computer Networks Lab 10

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the  software application rigorously | Needs assistance or guidance to proceed through experiment steps,  studies outputs with minor errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered in the lab manual according to  the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with  sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from  theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### LAB 11 Introduction to Aggregation, Group By, and JOINs

| **Name** | **Roll. No.** | **Total Marks** | **Obtained Marks** | **Viva Marks** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

30

**Procedure**

Please execute the following script, understand each line, draw the conceptual schema diagram for given relations and then start on the questions listed below:

**create** schema hotelBooking;

**use** hotelBooking;

**create** table hotel( hotelno varchar(10) primary key, hotelname varchar(20), city varchar(20) );

**insert** into hotel values('fb01', 'Grosvenor', 'London');

**insert** into hotel values('fb02', 'Watergate', 'Paris');

**insert** into hotel values('ch01', 'Omni Shoreham', 'London'); **insert** into hotel values('ch02', 'Phoenix Park', 'London'); **insert** into hotel values('dc01', 'Latham', 'Berlin');

**create** table room( roomno numeric(5), hotelno varchar(10), type varchar(10), price decimal(5,2), **primary key** (roomno, hotelno), **foreign key** (hotelno) REFERENCES hotel(hotelno) );

**insert** into room values(501, 'fb01', 'single', 19); **insert** into room values(601, 'fb01', 'double', 29); **insert** into room values(701, 'fb01', 'family', 39); **insert** into room values(1001, 'fb02', 'single', 58); **insert** into room values(1101, 'fb02', 'double', 86); **insert** into room values(1001, 'ch01', 'single', 29.99);

**insert** into room values(1101, 'ch01', 'family', 59.99); **insert** into room values(701, 'ch02', 'single', 10); **insert** into room values(801, 'ch02', 'double', 15); **insert** into room values(901, 'dc01', 'single', 18); **insert** into room values(1001, 'dc01', 'double', 30); **insert** into room values(1101, 'dc01', 'family', 35);

**create** table guest( guestno numeric(5), guestname varchar(20), guestaddress varchar(50),

**primary key** (guestno) );

**insert** into guest values(10001, 'John Kay', '56 High St, London'); **insert** into guest values(10002, 'Mike Ritchie', '18 Tain St, London'); **insert** into guest values(10003, 'Mary Tregear', '5 Tarbot Rd, Aberdeen'); **insert** into guest values(10004, 'Joe Keogh', '2 Fergus Dr, Aberdeen'); **insert** into guest values(10005, 'Carol Farrel', '6 Achray St, Glasgow'); **insert** into guest values(10006, 'Tina Murphy', '63 Well St, Glasgow'); **insert** into guest values(10007, 'Tony Shaw', '12 Park Pl, Glasgow');

**create** table booking( hotelno varchar(10), guestno numeric(5), datefrom datetime, dateto datetime, roomno numeric(5), **primary key** (hotelno, guestno, datefrom), **foreign key** (roomno, hotelno) REFERENCES room(roomno, hotelno), **foreign key** (guestno) REFERENCES guest(guestno) );

**insert** into booking values('fb01', 10001, '02-04-01', '02-04-08', 501);

**insert** into booking values('fb01', 10004, '04-04-15', '04-05-15', 601);

**insert** into booking values('fb01', 10005, '03-05-02', '03-05-07', 501);

**insert** into booking values('fb01', 10001, '04-05-01', null, 701);

**insert** into booking values('fb02', 10003, '09-04-05', '10-04-04', 1001);

**insert** into booking values('ch01', 10006, '04-04-21', null, 1101);

**insert** into booking values('ch02', 10002, '04-04-25', '04-05-06', 801);

**insert** into booking values('dc01', 10007, '06-05-13', '06-05-15', 1001);

EE432 Computer Networks

**insert** into booking values('dc01', 10003, '12-05-20', null, 1001);

**Exercise:**

* 1. List the names and addresses of all guests in London, alphabetically ordered by name.
  2. Display the names of all the hotels along with the number of rooms present in each of them.
  3. Display the AVG price of each hotel situated in London.
  4. Display the most expensive double, single and family rooms respectively.
  5. Display hotelname, cityname along with distinct number of room types available in each of them.
  6. Display the name and city of the hotel where guests from London are staying. The list should not contain any hotel twice.
  7. Display the name, city of all the hotels along with the number of reservations it has, in descending order.
  8. Display the names of all the guests who have not provided with the end date of their reservations.
  9. Display the HotelName and RoomNo which was reserved in either year 2003 or 2004, also display the Guest No of the respective guest.
  10. Display the name of the hotel and city which has not been reserved.
  11. How many different guests have made bookings till May, 2015?
  12. What is the total revenue per night from all double rooms?
  13. How many different guests have made bookings for August?
  14. List the price and type of all rooms at the ‘Avari’ Hotel in Lahore.
  15. List all guests currently staying at the ‘Marriott’ Hotel. (‘system date’ is used to determine current date)
  16. What is the total income from bookings for the ‘Hotel Inn’ Hotel today?
  17. List the rooms which are currently unoccupied at the ‘Hotel Inn’ Hotel*.*
  18. What is the lost income from unoccupied rooms at the ‘Hotel Inn’ Hotel?
  19. What is the lost income from unoccupied rooms at each hotel today?
  20. For each hotel with more than two different types of rooms, what is the lost income from unoccupied rooms?

### Assessment Rubrics for EE432: Computer Networks Lab 11

32

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment**  **(a)** | Downloads and installs required software and sets up the system according to the  experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with  required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and disciplinary**  **rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered in the lab manual according to  the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with  sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from  theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

1. **Install Google App Engine. Create *hello world* app and other simple web**

**applications using python/java**

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

34

**14.1 Procedure**

1. Install Google Plugin for Eclipse

Read this guide – how to install Google Plugin for Eclipse. If you install the Google App Engine Java SDK together with “Google Plugin for Eclipse“, then go to step 2, Otherwise, get the Google App Engine Java SDK and extract it.

1. Create New Web Application Project

In Eclipse toolbar, click on the Google icon, and select “New Web Application Project…” Figure – New Web Application Project

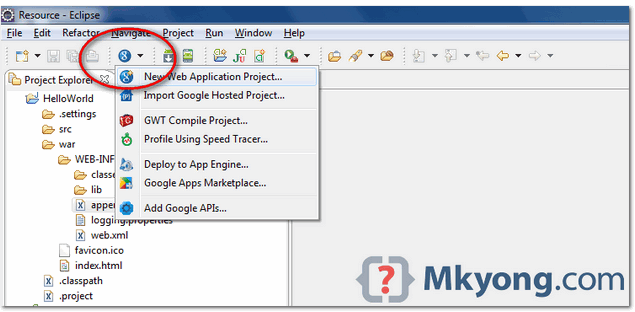
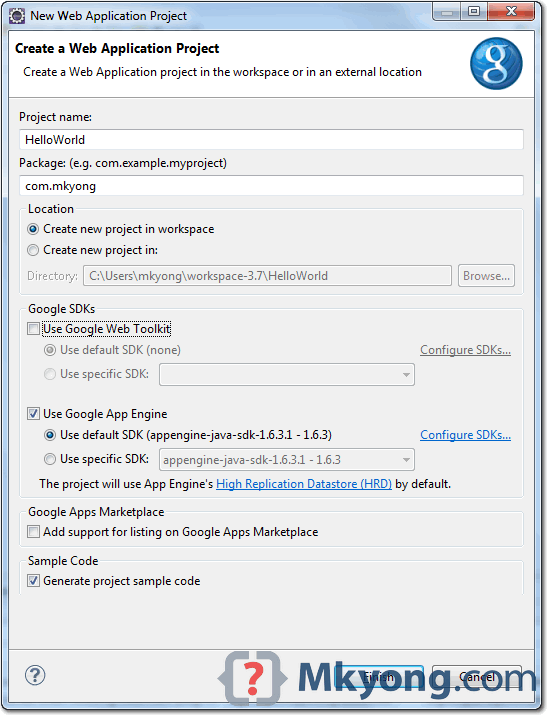


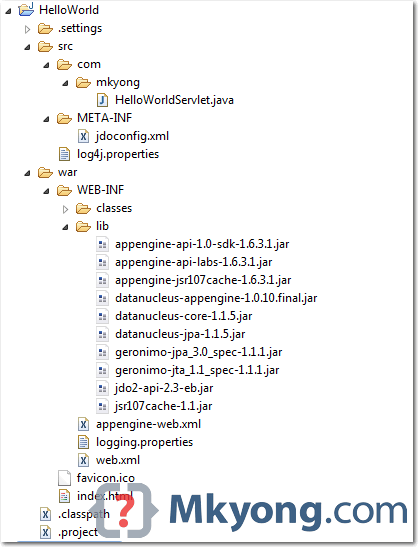
Figure – Deselect the “Google Web ToolKit“, and link your GAE Java SDK via the “configure SDK” link.



Click finished, Google Plugin for Eclipse will generate a sample project automatically.

1. *Hello World*

Review the generated project directory.



Nothing special, a standard Java web project structure. HelloWorld/ src/

...Java source code... META-INF/

...other configuration... war/

...JSPs, images, data files... WEB-INF/

...app configuration... lib/

...JARs for libraries... classes/

...compiled classes...

Copy

The extra is this file “appengine-web.xml“, Google App Engine need this to run and deploy the application. File : appengine-web.xml

<?xml version="1.0" encoding="utf-8"?>

<appengine-web-app xmlns="<http://appengine.google.com/ns/1.0>">

<application></application>

<version>1</version>

<!-- Configure java.util.logging -->

<system-properties>

<property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>

</system-properties>

</appengine-web-app> Copy

1. Run it local

Right click on the project and run as “Web Application“. Eclipse console :

//...

INFO: The server is running at <http://localhost:8888/>

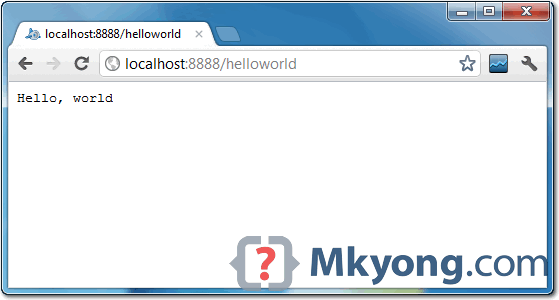
30 Mac 2012 11:13:01 PM com.google.appengine.tools.development.DevAppServerImpl start INFO: The admin console is running at <http://localhost:8888/_ah/admin>

Copy

Access URL [http://localhost:8888/,](http://localhost:8888/) see output



and also the hello world servlet – <http://localhost:8888/helloworld>



1. Deploy to Google App Engine

Register an account on https://appengine.google.com/, and create an application ID for your web application. In this demonstration, I created an application ID, named “mkyong123”, and put it in appengine web.xml.

File : appengine-web.xml

-

<?xml version="1.0" encoding="utf-8"?>

<appengine-web-app xmlns="<http://appengine.google.com/ns/1.0>">

<application>mkyong123</application>

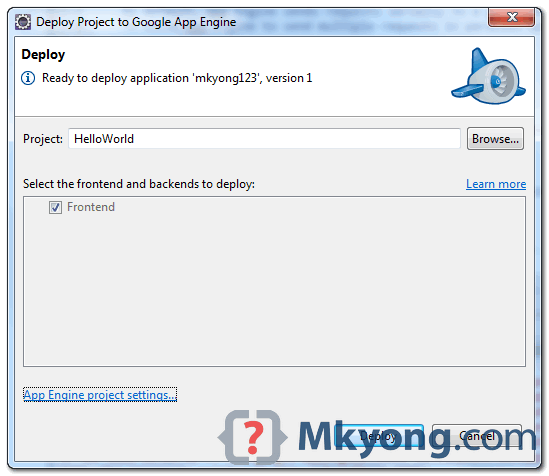
<version>1</version>

<!-- Configure java.util.logging -->

<system-properties>

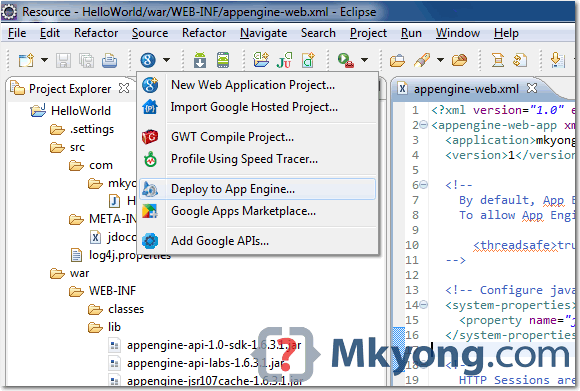
<property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>

</system-properties>

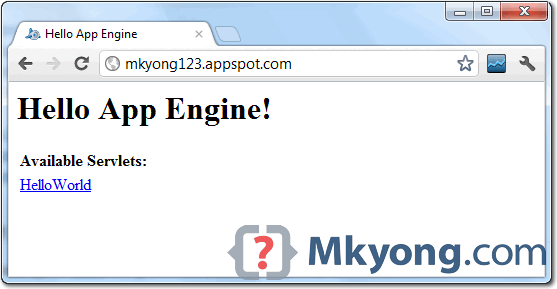
</appengine-web-app> Copy

To deploy, see following steps:

Figure – Click on GAE deploy button on the toolbar.



*Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL –* [*http://mkyong123.appspot.com/*](http://mkyong123.appspot.com/)



Thus the simple application was created successfully.

### Assessment Rubrics for EE432: Computer Networks Lab 12

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

1. **Simulate a cloud scenario using CloudSim and run a scheduling algorithm that**

**is not present in CloudSim.**

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

42

**14.1 Procedure**

How to use CloudSim in Eclipse

CloudSim is written in Java. The knowledge you need to use CloudSim is basic Java programming and some basics about cloud computing. Knowledge of programming IDEs such as Eclipse or NetBeans is also helpful. It is a library and, hence, CloudSim does not have to be installed. Normally, you can unpack the downloaded package in any directory, add it to the Java classpath and it is ready to be used. Please verify whether Java is available on your system.

To use CloudSim in Eclipse:

1. Download CloudSim installable files

from https://code.google.com/p/cloudsim/downloads/list and unzip

1. Open Eclipse
2. Create a new Java Project: File -> New
3. Import an unpacked CloudSim project into the new Java Project

The first step is to initialise the CloudSim package by initialising the CloudSim library, as follows CloudSim.init(num\_user, calendar, trace\_flag)

1. Data centres are the resource providers in CloudSim; hence, creation of data centres is a second step. To create Datacenter, you need the DatacenterCharacteristics object that stores the properties of a data centre such as architecture, OS, list of machines, allocation policy that covers the time or spaceshared, the time zone and its price:

Datacenter datacenter9883 = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList), s

1. The third step is to create a broker:

DatacenterBroker broker = createBroker();

1. The fourth step is to create one virtual machine unique ID of the VM, userId ID of the VM’s owner, mips, number Of Pes amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudletScheduler policy for cloudlets:

Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())

1. Submit the VM list to the broker: broker.submitVmList(vmlist)
2. Create a cloudlet with length, file size, output size, and utilisation model:

Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationMode

1. Submit the cloudlet list to the broker: broker.submitCloudletList(cloudletList) Sample Output from the Existing Example:

Starting

CloudSimExample1... Initialising...

Starting CloudSim version 3.0 Datacenter\_0 is starting...

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>null

Broker is starting... Entities started.

: Broker: Cloud Resource List received with 1 resource(s) 0.0: Broker: Trying to Create VM #0 in Datacenter\_0

: Broker: VM #0 has been created in Datacenter #2, Host #0 0.1: Broker: Sending cloudlet 0 to VM #0

400.1 : Broker: Cloudlet 0 received

: Broker: All Cloudlets executed. Finishing 400.1: Broker: Destroying VM #0

Broker is shutting down... Simulation: No more future events

CloudInformationService: Notify all CloudSim entities for shutting down. Datacenter\_0 is shutting down...

Broker is shutting down Simulation completed. Simulation completed.

========== OUTPUT ==========

Cloudlet ID STATUS Data center ID Finish Time 0 SUCCESS 2

0.1 400.1 VM ID Time 0 Start Time

400

\*\*\*\*\*Datacenter: Datacenter\_0\*\*\*\*\* User id Debt

3 35.6

CloudSimExample1 finished!

### Assessment Rubrics for EE432: Computer Networks Lab 13

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered in the lab manual according to  the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with  sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from  theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

1. **Install Hadoop single node cluster and run simple applications like**

**wordcount.**

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

* 1. **Procedure**

Install Hadoop

Step 1: Click here to download the Java 8 Package. Save this file in your home directory. Step 2: Extract the Java Tar File.

Command: tar -xvf jdk-8u101-linux-i586.tar.gz



Hadoop Installation – Extracting Java Files

Step 3: Download the Hadoop 2.7.3 Package.

Command: wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz

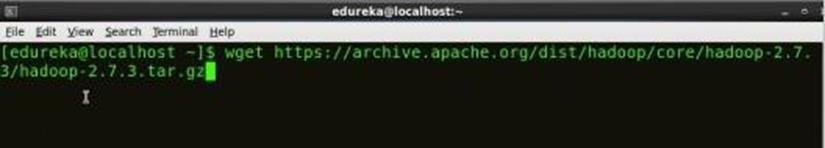


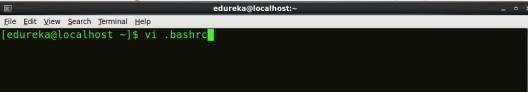
Fig: Hadoop Installation – Downloading Hadoop

**Step 4:** Extract the Hadoop tar File.

***Command***: tar -xvf hadoop-2.7.3.tar.gz



Fig: Hadoop Installation – Extracting Hadoop Files Step

**5:** Add the Hadoop and Java paths in the bash file (.bashrc). Open**. bashrc** file. Now, add Hadoop and Java Path as shown below. ***Command*:** vi .bashrc

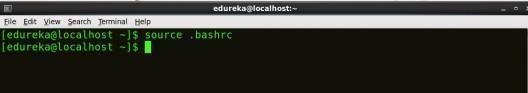


*Fig: Hadoop Installation – Setting Environment Variable*

Then, save the bash file and close it.

For applying all these changes to the current Terminal, execute the source command.

***Command*:** source .bashrc



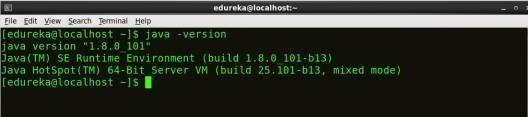
*Fig: Hadoop Installation – Refreshing environment variables*

To make sure that Java and Hadoop have been properly installed on your

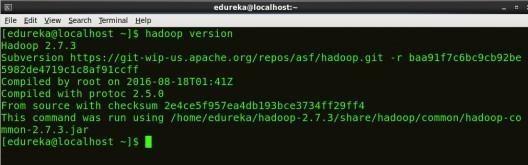
system and can be acces

***Command*:** java -version

*Fig: Hadoop Installation – Checking Java Version*



***Command*:** hadoop version



*Fig: Hadoop Installation – Checking Hadoop Version*

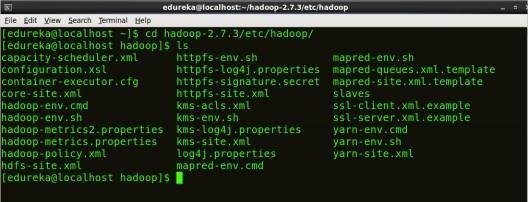
**Step 6:** Edit the **Hadoop Configuration files**.

***Command:*** cd hadoop-2.7.3/etc/hadoop/



***Command:*** ls

All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop**

directory as you can see in the snapshot below:

*Fig: Hadoop Installation – Hadoop Configuration Files*

**Step 7:** Open *core-site.xml* and edit the property mentioned below inside configuration tag:

*core-site.xml* informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

***Command*:** vi core-site.xml



*Fig: Hadoop Installation – Configuring core-site.xml*

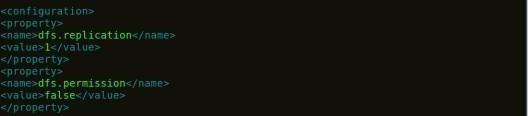
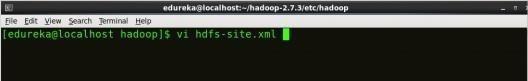


**Step 8:** Edit *hdfs-site.xml* and edit the property mentioned below inside

**configuration tag:**

*hdfs-site.xml* contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

***Command*:** vi hdfs-site.xml



*Fig: Hadoop Installation – Configuring hdfs-site.xml*



**Step 9:** Edit the *mapred-site.xml* file and edit the property mentioned below

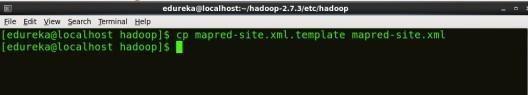
**inside configuration tag:**

*mapred-site.xml* contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the mapred- site.xml file using mapred-site.xml template.

***Command*:** cp mapred-site.xml.template mapred-site.xml

***Command*:** vi mapred-site.xml.



*Fig: Hadoop Installation – Configuring mapred-site.xml*



**Step 10:** Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag:

*yarn-site.xml* contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

***Command*:** vi yarn-site.xml



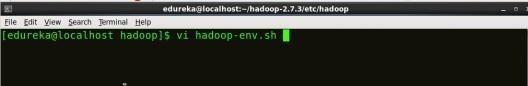
*Fig: Hadoop Installation – Configuring yarn-site.xml*

**Step 11:** Edit *hadoop-env.sh* and add the Java Path as mentioned below:



*hadoop-env.sh* contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

***Command*:** vi hadoop–env.sh



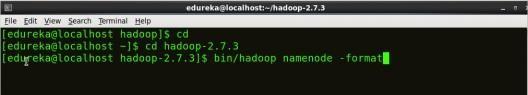
*Fig: Hadoop Installation – Configuring hadoop-env.sh*

**Step 12:** Go to Hadoop home directory and format the NameNode.

***Command*:** cd

***Command*:** cd hadoop-2.7.3

***Command*:** bin/hadoop namenode -format



*Fig: Hadoop Installation – Formatting NameNode*

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

Step 13: **Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.**

***Command:*** cd hadoop-2.7.3/sbin

Either you can start all daemons with a single command or do it individually.

***Command:*** *./*start-all.sh

The above command is a combination of ***start-dfs.sh, start-yarn.sh*** & ***mr- jobhistory- daemon.sh***

Or you can run all the services individually as below:

Start NameNode:

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

Command: ./hadoop-daemon.sh start namenode

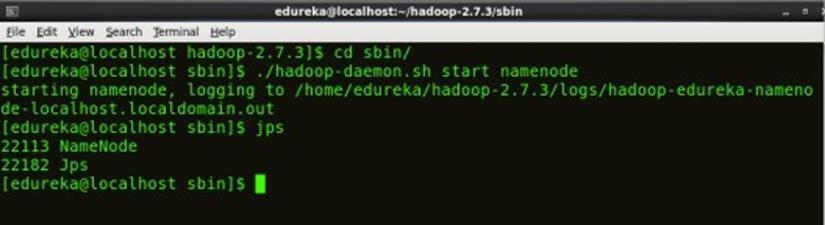
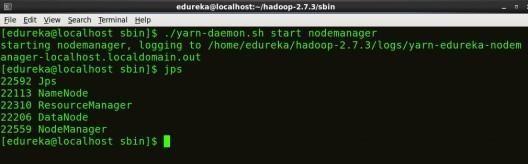


Fig: Hadoop Installation – Starting NameNode Start DataNode:



[See Batch Details](https://www.edureka.co/big-data-hadoop-training-certification)

*Fig: Hadoop Installation – Starting NodeManager*

 ***Start JobHistoryServer:***

JobHistoryServer is responsible for servicing all job history related requests from client.

***Command*:** ./mr-jobhistory-daemon.sh start historyserver

Step 14: **To check that all the Hadoop services are up and running, run the below command.**

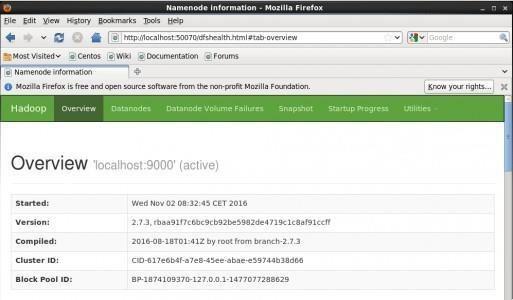
***Command:*** jps



*Fig: Hadoop Installation – Checking Daemons*

**Step 15:** Now open the Mozilla browser and go

to **localhost**:**50070/dfshealth.html** to check the NameNode interface.



*Fig: Hadoop Installation – Starting WebUI*

Congratulations, you have successfully installed a single node Hadoop cluster

#### Result:

Thus the Hadoop one cluster was installed and simple applications executed successfully.

### Assessment Rubrics for EE432: Computer Networks Lab 14

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |

| **Department of Electrical Engineering, UET Lahore** |
| --- |
| **EE432: Computer Networks** |

| **Course Instructor: Dr. Naveed Nawaz** | **Dated:** |
| --- | --- |
|  | **Semester: 7th** |
|  | **Session: Fall 2023** |

#### Microsoft Azure

| **Name** | **Roll. No.** | **Report Marks (10)** | **Viva Marks (5)** | **Total Marks (15)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on:

Signature:

**Module 1: Introduction to Microsoft Azure Lab: Managing Microsoft Azure**

* + 1. **Scenario**

A. Datum Corporation wants to expand their cloud presence by taking advantage of the benefits of Azure. Your task is to explore and compare the available IaaS features by using the Azure portal, Windows PowerShell, and Azure CLI.

###### Objectives

After completing this lab, you will be able to:

* + - * Use the Azure portals.
      * Use Azure Resource Manager features via the Azure portal.
      * Use Azure PowerShell.
      * Use Azure CLI

###### Lab Setup

Estimated Time: 50 minutes

Virtual Machine: **20533E-MIA-CL1**

User Name: **Student**

Password: **Pa55w.rd**

Before starting this lab, ensure that you have performed the "Preparing the Environment" demonstration at the beginning of the first lesson in this module, and that the setup script has completed.

**Note:** The Microsoft Azure portal is continually improved, and the user interface might have been updated since this lab was written. Your instructor will make you aware of any differences between the steps described in the lab and the current Azure portal user interface.

 **Exercise 1: Using the Azure portals**

###### Scenario

A. Datum has asked you to explore the available browser-based Azure portals to assess how the corporation will use them. In the Azure portal, you must observe the organization of resources and customize the interface to make your testing environment more accessible. In the Azure Account Center,

you must view and download your current billing data. You must also identify the association between your subscription and an Azure AD tenant.

The main tasks for this exercise are as follows:

1. Use the Azure portal
2. Use the Azure account portal
   * + 1. **Task 1: Use the Azure portal**
          1. In Microsoft Edge, browse to the dashboard page of the Azure portal.
          2. Edit the dashboard by changing the size of the **All resources** tile to **4x6**.
          3. Move the **Service health** tile and the **Marketplace** tile such that they remain adjacent but their top edge aligns with the bottom edge of the **Quickstart tutorials** tile.
          4. Save the edits.
          5. Review the results and reset the dashboard to the default state.

**Note**: This will remove all customizations of your current Dashboard.

* + - * 1. Add **Tags** to the hub menu.
        2. In the Azure portal, navigate to the blade displaying properties of your subscription.
        3. On the blade representing your subscrption, in the **Overview** section, note the **Directory** entry, referencing the Azure Active Directory tenant associated with your subscription. In addition, note the **Change directory** entry in the toolbar, which allows you to switch the association to a different tenant.
        4. Leave the Microsoft Edge window open.
      1. **Task 2: Use the Azure account portal**
         1. Start Microsoft Edge and browse to the Azure account portal.
         2. If prompted, sign in by using the Microsoft account that is the Account Administrator of your Azure subscription.
         3. On the Account portal page, navigate to the summary page of your Azure subscription and review the billing summary for your subscription.
         4. From the summary page, download usage details in Version 2.
         5. Open the usage details in Notepad. Note that this is intended to simply review its content – typically to analyze it in more details, you would use Microsoft Excel or other program capable of parsing csv files. The file might not include any data at this point if you have not yet deployed any resources into your subscription.
         6. Close Notepad.
         7. Close the Microsoft Edge window.

**Result**: After completing this exercise, you should have used the Azure portals.

#### Exercise 2: Using the Azure Resource Manager features in the Azure portal



###### 2.3.1. Scenario

A. Datum has asked you to create some temporary resources in Azure via the Azure portal. You must create a resource group and a resource, and then tag them to indicate that they are part of the lab environment. Finally, you must delegate the contributor permissions to the resource.

The main tasks for this exercise are as follows:

1. Create and manage a resource group
2. Create Azure resources
3. Configure tagging
4. Configure RBAC
   * + 1. **Task 1: Create and manage a resource group**
          1. Switch back to the Microsoft Edge window displaying the Azure portal.
          2. In the Azure portal, navigate to the Resource groups blade.
          3. From the Resource groups blade, add a new resource group with the following settings:

Resource group name: **20533E0101-LabRG**

Subscription: the name of your Azure subscription

Resource group location: the Azure region closest to the lab location

* + - 1. **Task 2: Create Azure resources**
         1. In the Azure portal, navigate to the **New** blade.
         2. From the **New** blade, create a new route table with the following settings:

Name: **20533E0101-rt**

Subscription: the same Azure subscription in which you created the resource group

Resource group name: **20533E0101-LabRG**

Location: the same Azure region in which you created the resource group

BGP route propagation: **Disabled**

* + - 1. **Task 3: Configure tagging**
         1. In the Azure portal, assign the tag named **project** with the value **test** to the resource group **20533E0101-LabRG**.
         2. In the Azure portal, assign the tag named **project** with the value **test** to the route table **20533E0101-rt**
         3. From the hub menu, navigate to the **Tags** blade.
         4. View entries with the tag **project : test**.
         5. Pin the list of resources with the tag **project : test** to Dashboard.
      2. **Task 4: Configure RBAC**
         1. In the Azure portal, navigate to the **20533E0101-LabRG** resource group.
         2. From the resource group blade, grant the contributor role to a valid Microsoft account name.

**Result**: After completing this exercise, you should have used the Azure Resource Manager features in the Azure portal.

#### Exercise 3: Using Azure PowerShell



###### 2.4.1. Scenario

A. Datum has asked you to investigate the capabilities of Azure PowerShell. You must connect to your Azure subscription by using Azure PowerShell, use Azure PowerShell to create a resource group and a resource, and then move the resource to another resource group.

The main tasks for this exercise are as follows:

1. Connect to your Azure subscription by using Azure PowerShell
2. Manage Azure resources and resource groups by using Azure PowerShell
   * + 1. **Task 1: Connect Azure PowerShell to your Azure subscription**
          1. On MIA-CL1, start Windows PowerShell ISE as Administrator.
          2. From the console pane of the Windows PowerShell ISE window, authenticate to Azure Resource Manager endpoint of your Azure subscription.
          3. From the console pane of the Windows PowerShell ISE window, review the list of subscriptions associated with the account you used to sign in.
          4. From the console pane of the Windows PowerShell ISE window, enumerate Azure resource providers, their registration state, their resource types, and the Azure regions where these resources are available.
       2. **Task 2: Manage Azure resources and resource groups by using Azure PowerShell**
          1. In the Windows PowerShell ISE window, open the **F:\Labfiles\Lab01\Starter\Set- 20533E0101Lab.ps1** file.
          2. In the **# Variables** section, note the values of predefined variables. They need to match the names of resource and the resource group you created in the previous exercise.
          3. Under the line that states **# Identify the location of the resource group containing the resource**, type the following:



* + - * 1. Run the resulting script.
        2. Under the line that states **# Create a new resource group in the same location**, type the following:



* + - * 1. Run the newly typed line only.
        2. Under the line that states **# Retrieve an object representing the resource and store it in a variable**, type the following:



* + - * 1. Run the newly typed line only.
        2. Under the line that states **# Move the resource to the new resource group**, type the following:



* + - * 1. Use the resulting script to move the resource represented by the $res variable to the resource group represented by the variable $g2.
        2. Under the line that states **# View resources in the new resource group**, type the following:



* + - * 1. Run the newly typed line.

**Result**: After completing this exercise, you should have used Azure PowerShell to manage Azure resources and resource groups.

#### Exercise 4: Using Azure CLI



###### Scenario

A. Datum has asked you to investigate the capabilities of Azure CLI. You must connect to your Azure subscription by using Azure CLI. Then you must use Azure CLI to create a resource group and a resource, and move the resource to another resource group.

The main tasks for this exercise are as follows:

1. Connect to your Azure subscription by using Azure CLI
2. Manage Azure resources and resource groups by using Azure CLI
3. Remove the lab environment
   * + 1. **Task 1: Connect to your Azure subscription by using Azure CLI**
          1. On MIA-CL1, start **Command Prompt** as Administrator
          2. From **Administrator: Command Prompt**, use Azure CLI 2.0 to sign in to your Azure subscription.
          3. From **Administrator: Command Prompt**, use Azure CLI 2.0 to display properties of the Azure subscription associated with the account you used to sign in. Take note of the value of

the **id** parameter, representing your Azure subscription ID. You will need it in the next task.

* + - * 1. From **Administrator: Command Prompt**, use Azure CLI 2.0 to list Azure resource providers, their registration state, and their resource types.
      1. **Task 2: Manage Azure resources and resource groups by using Azure CLI**
         1. From **Administrator: Command Prompt**, use Azure CLI 2.0 to display properties of the resource group **20533E0101-LabRG**.
         2. From **Administrator: Command Prompt**, use Azure CLI 2.0 to list resources in the resource group **20533E0102-LabRG**.
         3. In the list of resources, note the value of the **id** property of the **20533E0101-rt**.
         4. From **Administrator: Command Prompt**, use Azure CLI 2.0 to move the **20533E0101-rt** resource from the resource group **20533E0102-LabRG** to the resource group **20533E0101-LabRG**.
         5. From **Administrator: Command Prompt**, use Azure CLI 2.0 to list resources in the resource group **20533E0101-LabRG**
      2. **Task 3: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: After completing this exercise, you should have used Azure CLI to manage Azure resources and resource groups.

**Question** Why did you use Azure PowerShell cmdlets that contained **Rm** in the lab?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

# Module 2: Implementing and managing Azure networking

**Lab A: Using a deployment template and Azure PowerShell to implement Azure virtual networks**



##### Scenario

A. Datum Corporation plans to create several virtual networks in their Azure subscription. They will all reside in the same Azure region. You want to test the deployment of Azure virtual networks by using both imperative and declarative methods.

##### Objectives

After completing this lab, you will be able to:

* Create a virtual network by using deployment templates.
* Create a virtual network by using Azure PowerShell.
* Create a virtual network by using Azure CLI

##### Lab Setup

Estimated Time: 30 minutes

Virtual Machine: **20533E-MIA-CL1**

User Name: **Student**

Password: **Pa55w.rd**

**Note:** The Microsoft Azure portal is continually improved, and the user interface might have been updated since this lab was written. Your instructor will make you aware of any differences between the steps described in the lab and the current Azure portal user interface.

## Exercise 1: Creating an Azure virtual network by using a deployment template



##### Scenario

A. Datum wants to test the provisioning of virtual networks. You must configure these virtual networks by using deployment templates from GitHub

The main tasks for this exercise are as follows:

1. Review a GitHub Azure quickstart template
2. Perform the deployment from the Azure portal

**Task 1: Review a GitHub Azure quickstart template**

1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**.
2. Start Microsoft Edge and browse to the **Virtual Network with two Subnets** Github- hosted Azure quickstart template at [**http://aka.ms/Mt32e4**.](http://aka.ms/Mt32e4)
3. From the **Virtual Network with two Subnets** page, click **Deploy to Azure**.
4. If prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
5. In the Azure portal, click **Edit template**.
6. Review the structure of the JavaScript Object Notation (JSON) file. Examine the placeholders for values that can be edited during the deployment. This template contains the following

parameters: ***vnetName***, ***vnetAddressPrefix***, ***subnet1Prefix***, ***subnet1Name***, ***subnet2P refix***, and ***subnet2Name***.

1. Review the content of the **Resources** section to identify type of the resource, its name, and properties.
2. Close the Edit Template blade without making any changes.

**Task 2: Perform the deployment from the Azure portal**

1. From the **Create a Virtual Network with two Subnets** blade, deploy the template with the following settings:
   * Subscription: the name of your subscription
   * Resource group: create a new group named **20533E0203-LabRG**
   * Location: an Azure region you chose when running the provisioning script at the beginning of this module
   * Vnet Name: **20533E0203-vnet**
   * Vnet Address Prefix: **10.10.0.0/16**
   * Subnet1Prefix: **10.10.0.0/24**
   * Subnet1Name: **Subnet1**
   * Subnet2Prefix: **10.10.1.0/24**
   * Subnet2Name: **Subnet2**
   * Location: **[resourceGroup().location]**
2. Verify that provisioning of the new virtual network named **20533E0203- vnet** completed successfully.

**Result**: After completing this exercise, you should have created virtual networks for A. Datum HQ.

## Exercise 2: Creating a virtual network by using Azure PowerShell



##### Scenario

A. Datum is expanding their services in Azure by using both declarative and imperative deployment methods and they ask you to test provisioning of a new network by using Azure PowerShell.

The main tasks for this exercise are as follows:

1. Create a virtual network by using PowerShell

**Task 1: Create a virtual network by using PowerShell**

1. On MIA-CL1, start Windows PowerShell ISE as Administrator.
2. From the console pane of the Windows PowerShell ISE window, authenticate to Azure Resource Manager endpoint of your Azure subscription.
3. From the console pane of the Windows PowerShell ISE window, review the list of subscriptions associated with the account you used to sign in. Identify the value of the subscription Id property of the Azure subscription you want to use in this lab.
4. If there are multiple Azure subscriptions associated with your account, run the **Set- AzureRmContext** with the **-SubscriptionId** parameter to designate the one you want to use in this lab.
5. Run the **New-AzureRMResourceGroup** cmdlet to create a new resource group named **20533E0204-LabRG** in the same Azure region you chose in the previous exercise.
6. Run the **New-AzureRmVirtualNetwork** cmdlet to create a new virtual network named **20533E0204-vnet** with the address space **10.11.0.0/16** in the **20533E0204- LabRG** resource group and the same Azure region as the resource group.
7. Run the **Add-AzureRmVirtualNetworkSubnetConfig** cmdlet to add a subnet named **Subnet1** with the address prefix **10.11.0.0/24** to the virtual

network **20533E0204-vnet**.

1. Finalize your configuration by running the **Set-AzureRmVirtualNetwork** cmdlet.

**Result**: After completing this exercise, you should have created a virtual network by using Azure PowerShell.

## Exercise 3: Creating a virtual network by using Azure CLI



##### Scenario

A. Datum is expanding their services in Azure by using both declarative and imperative deployment methods. They have asked you to test the provisioning of a new network by using Azure CLI.

The main tasks for this exercise are as follows:

1. Creating a virtual network by using Azure CLI

**Task 1: Creating a virtual network by using Azure CLI**

1. On MIA-CL1, start **Command Prompt** as Administrator
2. From **Administrator: Command Prompt**, use Azure CLI 2.0 to sign in to your Azure subscription.
3. From **Administrator: Command Prompt**, use Azure CLI 2.0 to display properties of the Azure subscription associated with the account you used to sign in. Take note of the value of the **id** parameter, representing your Azure subscription ID.
4. Run the **az account set** command to specify the subscription in which you are going to create a virtual network.
5. Run the **az group create** command to create a new resource group

named **20533E0205-LabRG** in the same Azure region you chose in the previous exercise.

1. Run the **az network vnet create** command to create a virtual network named **20533E0205-vnet** with the address space **10.12.0.0/16** and a subnet named **Subnet1** with the address prefix of **10.12.0.0/24** in the **20533E0205- LabRG** resource group and the same Azure region as the resource group.
2. Run the **az network vnet subnet create** command to add a subnet named **Subnet2** with the address prefix **10.12.1.0/24** to the virtual network **20533E0205-vnet**.

**Result**: After completing this exercise, you should have created a virtual network by using Azure CLI.

**Question** What are some of the methods you can use to create an Azure virtual network?

# Lab B: Configuring VNet peering



##### Scenario

Now that A. Datum Corporation has deployed Azure Resource Manager VNets, the company wants to be able to provide direct connectivity between them. Your plan is to implement VNet peering to provide the optimal performance with minimum cost.

##### Objectives

After completing this lab, you will be able to:

* Connect Azure virtual networks using VNet peering.
* Configure VNet peering-based service chaining
* Validate virtual network connectivity using Azure-based and VM-based tools.

##### Lab Setup

Estimated Time: 35 minutes

Virtual Machine: **20533E-MIA-CL1**

User Name: **Student**

Password: **Pa55w.rd**

Before starting this lab, ensure that you have performed the "Preparing the Environment" demonstration tasks at the beginning of the first lesson in this module, and that the setup script has completed.

**Note:** The Microsoft Azure portal is continually improved, and the user interface might have been updated since this lab was written. Your instructor will make you aware of any differences between the steps described in the lab and the current Azure portal user interface.

## Exercise 1: Using the Azure portal to configure VNet peering



##### Scenario

A. Datum wants to use VNet peering to provide connectivity between pairs of virtual networks.

The main tasks for this exercise are as follows:

1. Configure VNet peering for the first virtual network
2. Configure VNet peering for the second virtual network

**Task 1: Configure VNet peering for the first virtual network**

1. Ensure that you are signed in to MIA-CL1 as **Student** with the

password **Pa55w.rd** and that the **Add-20533EEnvironment** script successfully completed. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.

1. In Microsoft Edge, navigate to the **20533E0201-vnet** virtual network blade.
2. From the **20533E0201-vnet** blade, create a VNet peering with the following settings:
   * Name: **20533E0201-vnet-To-20533E0202-vnet**
   * Virtual network deployment model: **Resource manager**
   * Subscription: the name of your Azure subscription
   * Virtual network: **20533E0202-vnet**
   * Allow virtual network access: **Enabled**
   * Allow forwarded traffic: disabled
   * Allow gateway transit: disabled
   * Use remote gateways: disabled

**Task 2: Configure VNet peering for the second virtual network**

1. In Microsoft Edge, navigate to the **20533E0202-vnet** virtual network blade.
2. From the **20533E0202-vnet** blade, create a VNet peering with the following settings:
   * Name: **20533E0202-vnet-To-20533E0201-vnet**
   * Virtual network deployment model: **Resource manager**
   * Subscription: the name of your Azure subscription
   * Virtual network: **20533E0201-vnet**
   * Allow virtual network access: **Enabled**
   * Allow forwarded traffic: disabled
   * Allow gateway transit: disabled
   * Use remote gateways: disabled

**Result**: After completing this exercise, you should have configured VNet peering between two virtual networks.

## Exercise 2: Configuring VNet peering–based service chaining



##### Scenario

A. Datum now wants to test the service chaining capabilities of VNet peering to minimize cost and management overhead of the Azure virtual network infrastructure.

The main tasks for this exercise are as follows:

1. Configure IP forwarding
2. Configure user defined routing
3. Configure routing on an Azure VM running Windows Server 2016

**Task 1: Configure IP forwarding**

1. In Microsoft Edge, navigate to the **20533E0201-nic1** blade.
2. On the **20533E0201-nic1** blade, modify the **IP configurations** by setting **IP forwarding** to **Enabled**.

**Task 2: Configure user defined routing**

1. In the Azure portal, create a new route table with the following settings:
   * Name: **20533E02-rt1**
   * Subscription: the name of your Azure subscription
   * Resource group: **20533E0202-LabRG**
   * Location: the same Azure region in which you created the virtual network 20533E0202- vnet
   * BGP route propagation: **Disabled**
2. In the Azure portal, add to the route table a route with the following settings:
   * Route name: **custom-route-to-20533E0201-vnet**
   * Address prefix: 10.0.0.0/22
   * Next hop type: **Virtual appliance**
   * Next hop address: **10.0.0.4**
3. In the Azure portal, associate the route table with the **subnet-1** of the **20533E0202- vnet**.

**Task 3: Configure routing on an Azure VM running Windows Server 2016**

1. On MIA-CL1, from the Azure portal, start a Remote Desktop session to **20533E0201- vm1** Azure VM.
2. When prompted to authenticate, specify the following credentials:
   * User name: **Student**
   * Password: **Pa55w.rd1234**
3. Once you are connected to 20533E0201-vm1 via the Remote Desktop session, from **Server Manager**, install the **Remote Access** server role with the **Routing** role service and all required features.
4. In the Remote Desktop session to 20533E0201-vm1, start the **Routing and Remote Access** console.
5. In the **Routing and Remote Access** console, run **Routing and Remote Access Server Setup Wizard** and enable **LAN routing**.
6. Start **Routing and Remote Access** service.
7. In the Remote Desktop session to 20533E0201-vm1, start the **Windows Firewall with Advanced Security** console and enable **File and Printer Sharing (Echo Request - ICMPv4-In)** inbound rule for all profiles.

**Result**: After completing this exercise, you should have configured VNet peering–based service chaining.

## Exercise 3: Validating virtual network connectivity



##### Scenario

A. Datum now wants to validate the VNet peering configuration by testing connectivity between virtual machines on different virtual networks.

The main tasks for this exercise are as follows:

1. Configure Windows Firewall with Advanced Security on an Azure VM
2. Test service chaining between peered virtual networks
3. Remove the lab environment

**Task 1: Configure Windows Firewall with Advanced Security on an Azure VM**

1. On MIA-CL1, from the Azure portal, start a Remote Desktop session to **20533E0201- vm2** Azure VM.
2. When prompted to authenticate, specify the following credentials:
   * User name: **Student**
   * Password: **Pa55w.rd1234**
3. In the Remote Desktop session to 20533E0201-vm2, start the **Windows Firewall with Advanced Security** console and enable **File and Printer Sharing (Echo Request - ICMPv4-In)** inbound rule for all profiles.

**Task 2: Test service chaining between peered virtual networks**

1. On MIA-CL1, from the Azure portal, start a Remote Desktop session to **20533E0202- vm1** Azure VM.
2. When prompted to authenticate, specify the following credentials:
   * User name: **Student**
   * Password: **Pa55w.rd1234**
3. Once you are connected to 20533E0202-vm1 via the Remote Desktop session, start **Windows PowerShell**.
4. In the **Windows PowerShell** window, run the following:

Test-NetConnection -ComputerName 10.0.1.4 -TraceRoute

1. Verify that test is successful and note that the connection was routed over 10.0.0.4

**Task 3: Remove the lab environment**

1. On MIA-CL1, close all open windows without saving any files.
2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
4. If you have multiple Azure subscriptions, select the one you want the script to target.
5. If prompted, specify the current lab number.
6. When prompted for confirmation, type **y**.
7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
8. In the Azure portal, reset the dashboard to the default state.
9. Close all open windows.

**Result**: After completing this exercise, you should have validated virtual network connectivity in the VNet peering configuration

**Question** What do you consider to be the most important advantages of VNet peering?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 3: Implementing virtual machines Lab B: Deploying Azure VMs

##### Scenario

As part of the planning for deployment of Azure VMs to Azure, Adatum Corporation has evaluated its deployment options. You must use the Azure portal and Azure PowerShell to deploy two Microsoft Azure VMs for the database tier of the Research and Development application. To facilitate resource tracking, you should ensure that the virtual machines are part of the same resource group. Both VMs should be part of the same availability set.

You must use an Azure Resource Manager template to deploy two additional Linux VMs and two additional Windows VMs that the ResDev application will use. The virtual machines should be part of the resource group, to facilitate resource tracking. Linux virtual machines should reside on the virtual networks’ app subnet, and Windows virtual machines should reside on the web subnet of the 20533E0301-LabVNet virtual network.

##### Objectives

After completing this lab, you will be able to:

* + - * Create Azure VMs by using the Azure portal and Azure PowerShell.
      * Validate virtual-machine creation.
      * Use Visual Studio and an Azure Resource Manager template to deploy Azure VMs
      * Use Azure PowerShell and an Azure Resource Manager template to deploy Azure VMs
      * Use Azure CLI and an Azure Resource Manager template to deploy Azure VMs

##### Lab Setup

Estimated Time: 60 minutes

**Virtual machine**: **20533E-MIA-CL1 User name**: **Student**

**Password**: **Pa55w.rd**

 **Exercise 1: Creating Azure VMs by using the Azure portal and Azure PowerShell**



##### Scenario

You must deploy two Azure VMs that are running Windows Server 2016 Datacenter. Name these machines 20533E03LabVM1 and 20533E03LabVM2. You will use the Azure portal to deploy one VM, and Azure PowerShell to deploy the other VM. You must deploy both virtual machines into the 20533E0301-LabRG resource group, and you must configure the virtual machines to use the database subnet of the 20533E0301-LabVNet virtual network. Both VMs should use managed disks and be part of the same availability set. After deploying the virtual machines, you will confirm successful deployment of the virtual machines.

The main tasks for this exercise are as follows:

* + - 1. Use the Azure portal to create a virtual machine
      2. Use Azure PowerShell to create a virtual machine
      3. **Task 1: Use the Azure portal to create a virtual machine**
         1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**.
         2. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         3. In the Azure portal, create a new Azure VM based on the **Windows Server 2016 Datacenter** image with the following settings:

Subscription: the name of your Azure subscription

Resource group: a new resource group named **20533E0301-LabRG**.

Name: **20533E03labVM1**

Region: an Azure region close to the location of the lab environment, in which you can provision Azure VMs

Availability set: **20533E0301-db-avset** with 2 fault domains and 5 update domains

Size: any available size with one vCPU

User name: **Student**

Password: **Pa55w.rd1234**

Use managed disks: **Yes**

Public inbound ports: **RDP**

Windows licensing : included in the runtime rate of the VM

OS disk type: **Standard HDD**

Virtual network: **20533E0301-labVNet** with address space **10.0.0.0/20** and a subnet named **database** with the address range **10.0.0.0/24**

NIC network security group: **Basic**

Load balancing: **No**

Boot diagnostics: **On**

Diagnostics storage account: accept the default setting

OS guest diagnostics: **Off**

System assigned maanged identity: **Off**

Enable auto-shutdown: **Off**

Enable backup: **Off**

Accept the default settings for the **Guest config** and **Tags** settings.

* + - * 1. Wait for the deployment to complete successfully.
        2. Leave the Microsoft Edge with the Azure portal window open.
      1. **Task 2: Use Azure PowerShell to create a virtual machine**
         1. On MIA-CL1, open a Windows PowerShell ISE window as Administrator
         2. In the Windows PowerShell ISE, open the script **F:\Labfiles\Lab03\Starter\New- 20533E03labVM2.ps1** and review its content.
         3. Run the script.
         4. When prompted, sign in using the Microsoft account that is the Service Administrator of your Azure subscription.
         5. If you have multiple subscriptions, select the one you used when running **Add- 20533EEnvironment** at the beginning of this module.
         6. When the script is complete, leave the Windows PowerShell ISE window open.

**Result**: After completing this exercise, you have created virtual machines by using the Azure portal and Azure PowerShell.

 **Exercise 2: Validating Azure VM deployment**



##### Scenario

You now must validate the creation and configuration of the Azure VMs that you created, to ensure that they function properly.

The main tasks for this exercise are as follows:

* + - 1. Use Azure PowerShell to validate virtual machine deployment
      2. Use the Azure portal to validate virtual machine deployment
      3. **Task 1: Use Azure PowerShell to validate virtual machine deployment**
         1. In the Windows PowerShell ISE window, at the command prompt, run the following command:

Get-AzureRmResource | Where-Object ResourceType -like "\*VirtualMachines"

* + - * 1. Confirm that the 20533E03labVM1 and the 20533E03labVM2 virtual machines are listed.
        2. Close the Windows PowerShell ISE window.
      1. **Task 2: Use the Azure portal to validate virtual machine deployment**
         1. On MIA-CL1, in the Microsoft Edge window, in the Azure portal, navigate to the **20533E0301-LabRG** resource group blade.
         2. On the 20533E0301-LabRG blade, review the list of resources associated with both virtual machines.
         3. In the Azure portal, navigate to the 20533E03labVM1 blade, and confirm the following values:

Resource group: **20533E0301-LabRG**

Virtual network/subnet: **20533E0301-labVNet/database**

* + - * 1. Repeat step 3 for the **20533E03labVM2** virtual machine.

**Result**: After completing this exercise, you will have validated the creation and configuration of Azure Virtual Machines.

 **Exercise 3: Using Visual Studio and an Azure Resource Manager template to deploy Azure VMs**



##### Scenario

You must use Visual Studio to deploy two Linux Azure Resource Manager virtual machines for use as app servers in the ResDev app. You should name the

servers **20533E03LabVM3** and **20533E03LabVM4**. You have a deployment-template solution and the deployment details for both virtual machines. You must deploy the two virtual machines from Visual Studio, and then confirm that the virtual machines have been deployed successfully by using Azure PowerShell.

The main tasks for this exercise are as follows:

* + - 1. Use Visual Studio to deploy Linux app servers Azure VMs
      2. Use Azure PowerShell to validate the deployment of the app servers Azure VMs
      3. **Task 1: Use Visual Studio to deploy Linux app servers Azure VMs**
         1. On MIA-CL1, start Visual Studio 2017. If prompted, sign in with the Microsoft account that is the Service Administrator of your Azure subscription.
         2. In Visual Studio, open the

solution **ResDevLinuxDeploy.sln** from **F:\Labfiles\Lab03\Starter\Projects\ResDevLi nuxDeploy**.

* + - * 1. View the contents of the **azuredeploy.json** template.
        2. From the Solution Explorer, start a new deployment process of the first virtual machine into the **20533E0301-LabRG** resource group with the following settings:

vmName: **20533E03LabVM4**

adminUsername: **Student**

adminPassword: **Pa55w.rd1234**

virtualNetworkName: **20533E0301-LabVNet**

resourceGroupName: **20533E0301-LabRG**

subnetName: **app**

subnetPrefix: **10.0.1.0/24**

vmSize: set the value to the same Azure VM size you used when you provisioned an Azure VM from the Azure portal

ubuntuOSVersion: **16.04.0-LTS** or a more recent version if available

storageAccountType: **Standard\_LRS**

**Note:** Deployment will run with the output that appears in the Output pane, which is at the bottom of the window. When deployment is complete, you will receive a message stating that the template was deployed successfully to the resource group 20533E0301-LabRG.

* + - * 1. View the contents of the **Azuredeploy.parameters.json** file to verify that the parameters that you provided during deployment have been saved in this file.
        2. Start another deployment process by using the deployment that you used for the first virtual machine.
        3. Deploy another Azure VM by using the same template, setting its name to **20533E03LabVM3** but leaving all other parameter values the same.
        4. Close the solution but leave Visual Studio open.
      1. **Task 2: Use Azure PowerShell to validate the deployment of the app servers Azure VMs**
         1. On MIA-CL1, start Windows PowerShell ISE as Administrator.
         2. From the console pane of the Windows PowerShell ISE window, authenticate to Azure Resource Manager endpoint of your Azure subscription.
         3. From the console pane of the Windows PowerShell ISE window, review the list of subscriptions associated with the account you used to sign in. Identify the value of the subscription Id property of the Azure subscription you want to use in this lab.
         4. If there are multiple Azure subscriptions associated with your account, run the **Set- AzureRmContext** with the **-SubscriptionId** parameter to designate the one you want to use in this lab.
         5. From the console pane of the Windows PowerShell ISE window, identify all resources in the resource group **20533E0301-LabRG**, including

their **ResourceName** and **ResourceType** properties by running the **Find- AzureRmResource** cmdlet.

**Note** If you are using AzureRM 6.8.1 or newer, use the **Get-AzureRmResource** cmdlet instead

* + - * 1. In the cmdlet output, note the resources created in this exercise including virtual machines, disks, NICs, public IPs, and a storage account.
        2. Leave the Windows PowerShell ISE window open for the next exercise.

**Result**: After completing this exercise, you will have deployed Azure Virtual Machines by using Visual Studio and an Azure Resource Manager template.

 **Exercise 4: Using Azure PowerShell and an Azure Resource Manager template to deploy Azure VMs**



##### Scenario

You must deploy the Web tier virtual machines by using an Azure Resource Manager template and the Azure portal. The Web tier should consist of two virtual machines named **20533E03LabVM5** and **20533E03LabVM6**, running Windows Server 2016. You should deploy these two VMs to the **20533E0301-LabRG** resource group and

the **web** subnet of the **20533E0301-LabVNet** virtual network. You have a template and a Windows PowerShell script that you should edit to use to deploy the first of these two VMs. After you deploy the first VM, confirm the deployment by viewing the newly deployed resources in the Azure portal.

The main tasks for this exercise are as follows:

* + - 1. Use Azure PowerShell to deploy the Windows virtual machines
      2. Use the Azure portal to monitor deployment
      3. Use the Azure portal to validate deployment of the Windows virtual machine
      4. **Task 1: Use Azure PowerShell to deploy the Windows virtual machines**
         1. In the Windows PowerShell ISE window that you launched in the previous exercise, open **F:\Labfiles\Lab03\Starter\Templates\Deploy-AzureResourceGroup.ps1**
         2. Review the script that will deploy the template.

**Note:** Note the $templateFile and $rgName variables. These represent the location of the Azure Resource Manager template file and the resource group to which you will deploy the virtual machines.

* + - 1. Switch to Visual Studio and open the

file **F:\Labfiles\Lab03\Starter\Templates\azuredeploywebvm.json**.

**Note:** Note that the template has a very similar structure to the template for the Linux virtual machines in the previous exercise. The primary differences between the two templates include the variables identifying the operating system image, the target subnet, and the availability set. You could replace these variables with equivalent parameters, in order to minimize the number of templates used to deploy Azure VMs.

* + - 1. Close Visual Studio.
      2. Switch back to the Windows PowerShell ISE window and run the **Deploy- AzureResourceGroup.ps1** script. When prompted, provide the following values:
         * vmName: **20533E03LabVM5**
         * adminUsername: **Student**
         * adminPassword: **Pa55w.rd1234**
         * virtualNetworkName: **20533E0301-LabVNet**
         * vmSize: **as specified by the instructor**
      3. **Task 2: Use the Azure portal to monitor deployment**
         1. To monitor the progress of the deployment, in Microsoft Edge, in the Azure portal, navigate to the **20533E0301-LabRG** resource group blade.
         2. On the 20533E0301-LabRG blade, in the Settings section, click the **Deployments** link.
         3. On the 20533E0301-LabRG - Deployments blade, click the **WebTierVM1- Deployment** link.
      4. **Task 3: Use the Azure portal to validate deployment of the Windows virtual machine**
         1. In Microsoft Edge, in the Azure portal, navigate back to the **20533E0301- LabRG** blade.
         2. On the 20533E0301-LabRG blade, in the Overview section, view the list of resources.
         3. Navigate to the **20533E03LabVM5** blade and, in the Essentials section, note that 20533E03LabVM5 has been assigned to the **20533E0301-LabVNet/web** virtual network/subnet and the operating system is **Windows**.

**Result**: After completing this exercise, you should have deployed Azure Virtual Machines by using Azure PowerShell and Resource Manager templates.

 **Exercise 5: Using Azure CLI and an Azure Resource Manager template to deploy Azure VMs**



##### Scenario

You also want to test an alternative process of deploying Azure VMs by using Azure CLI and Azure Resource Manager templates.

The main tasks for this exercise are as follows:

* + - 1. Use Azure CLI to deploy the Windows virtual machines
      2. Use the Azure portal to monitor deployment
      3. Use the Azure portal to validate deployment of the Windows virtual machine
      4. Remove the lab environment
      5. **Task 1: Use Azure CLI to deploy the Windows virtual machines**
         1. On MIA-CL1, start **Command Prompt** as Administrator
         2. From **Administrator: Command Prompt**, use Azure CLI 2.0 to sign in to your Azure subscription.
         3. From **Administrator: Command Prompt**, use Azure CLI 2.0 to display properties of the Azure subscription associated with the account you used to sign in. Take note of the value of the **id** parameter, representing your Azure subscription ID.
         4. Run the **az account set** command to specify the subscription in which you are going to create a virtual network.
         5. Run the **az group deployment create** command to create a deployment

named **WebTierVM2-Deployment** of an Azure VM named **20533E03LabVM6** into the virtual network **20533E0301-LabVNet** and the resource group **20533E0301-**

**LabRG** by using the

template **F:\Labfiles\Lab03\Starter\Templates\azuredeploywebvm.json**

* + - * 1. When prompted to provide securestring value

for **adminUsername** , **adminPassword** , **vmSize** , type **Student** , **Pa55w.rd1234**, vmSize (Ask your instructor to determine the size of the VM) respectively.

* + - 1. **Task 2: Use the Azure portal to monitor deployment**
         1. To monitor the progress of the deployment, in Microsoft Edge, in the Azure portal, navigate to the **20533E0301-LabRG** resource group blade.
         2. On the 20533E0301-LabRG blade, in the Settings section, click the **Deployments** link.
         3. On the 20533E0301-LabRG - Deployments blade, click the **WebTierVM2- Deployment** link.
      2. **Task 3: Use the Azure portal to validate deployment of the Windows virtual machine**
         1. In Microsoft Edge, in the Azure portal, navigate back to the **20533E0301- LabRG** blade.
         2. On the 20533E0301-LabRG blade, in the Overview section, view the list of resources.
         3. Navigate to the 20533E03LabVM6 blade and, in the Essentials section, note that 20533E03LabVM6 has been assigned to the **20533E0301-LabVNet/web** virtual network/subnet and the operating system is **Windows**.
      3. **Task 4: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: After completing this exercise, you should have deployed Azure Virtual Machines by using Azure CLI and Resource Manager templates.

**Question** What differences regarding Azure VM resources did you notice when you created a virtual machine in the Azure portal versus in Azure PowerShell?

**Question** Can Microsoft Visual Studio and Azure PowerShell use the same Azure Resource Manager template to deploy an Azure VM?

**Question** How would you configure an Azure Resource Manager template to deploy multiple Azure VMs with different configurations?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 4: Managing virtual machines Lab C: Managing Azure virtual machines

##### Scenario

Now that you have validated basic deployment options of Azure VMs, you need to start testing more advanced configuration scenarios. Your plan is to step through a sample configuration of a two-tier A. Datum ResDev application. As part of your tests, you will set up the front-end tier consisting of a pair of load balanced Azure VMs hosting IIS configured by using the VM DSC extension. You will also set up a multi-disk volume by using Storage Spaces in a Windows Azure VM in the back-end tier.

##### Objectives

After completing this lab, you will be able to:

* Set up load balancing for a pair of Azure VMs and configure their workload by using the VM DSC extension.
* Implement Storage Space–based simple volumes in Azure VMs.

##### Lab Setup

Estimated Time: 60 minutes

Virtual Machine: **20533E-MIA-CL1**

User name: **Student**

Password: **Pa55w.rd**

 **Exercise 1: Configuring front-end web tier**



##### Scenario

You need to test the ability of Azure VMs in the same availability set to operate in a load balanced configuration by leveraging Azure load balancer. You also need to test the implementation of the desired state configuration in Azure by using VM Agent DSC extension to install the default IIS website on two Azure VMs that will host the web tier of the

A. Datum ResDev application. Once the installation is complete, you must test the availability of this setup by verifying that load balanced access to the default website is not affected by shutting down one of the Azure VMs.

The main tasks for this exercise are as follows:

1. Review the existing deployment
2. Implement an Azure Load Balancer
3. Install and configure IIS by using DSC and Windows PowerShell
4. Test the DSC configuration and virtual machine availability
   * + 1. **Task 1: Review the existing deployment**
          1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**.
          2. Start Microsoft Edge, browse to the Azure portal and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
          3. In the Azure portal, navigate to the resource group **20533E0401-LabRG**.
          4. On the 20533E0401-LabRG blade, review the list of resources. Note that includes an availability set named 20533E0401-avset.
          5. Navigate to the **20533E0401-avset** blade and note that the availability set has 2 fault domains, 5 update domains, and it contains two virtual machines. Also note that each VM has a unique fault domain and update domain.
          6. Leave the Microsoft Edge window with the Azure portal open.
       2. **Task 2: Implement an Azure Load Balancer**
          1. On MIA-CL1, from the Azure portal, create an Azure load balancer with the following settings:

Name: **20533E0401-ilb**

Type: **Public**

SKU: **Basic**

Public IP address: create an IP address named **20533E0401-ilbfe** with dynamically assigned IP address

Subscription: the name of your Azure subscription

Resource group: **20533E0401-LabRG**

Location: the same Azure region you chose when running the provisioning script at the beginning of this module

* + - * 1. Configure the newly created load balancer with the backend pool

named **20533E0401-ilb-bepool** and associate it to the availability set **20533E0401- avset** with **ipconfig1** of **20533E0401-vm0** and **ipconfig1** of **20533E0401-vm1**.

* + - * 1. Configure the load balancer with the health probe that has the following settings:

Name: **20533E0401-ilb-probetcp80**

Protocol: **HTTP**

Port: **80**

Path: **/**

Interval: **5**

Unhealthy threshold: **2**

* + - * 1. Configure the load balancer with the following load balancing rule:

Name: **20533E0401-ilb-ruletcp80**

IP Version: **IPv4**

Frontend IP address: **LoadBalancerFrontEnd**

Protocol: **TCP**

Port: **80**

Backend port: **80**

Backend Pool: **20533E0401-ilb-bepool (2 virtual machines)**

Probe: **20533E0401-ilbprobetcp80 (HTTP:80)**

Session persistence: **None**

Idle timeout: **4**

Floating IP (direct server return): **Disabled**

* + - * 1. Add to the load balancer with the following inbound NAT rule:

Name: **20533E0401-ilb-natrulerdpvm0**

Frontend IP address: **LoadBalancerFrontEnd**

Service: **Custom**

Protocol: **TCP**

Port: **33890**

Associated to: **20533E0401-avset (availability set)**

Target virtual machine: **20533E0401-vm0**

Network IP configuration: **ipconfig1**

Port mapping: **Custom**

Floating IP (direct server return): **Disabled**

Target port: **3389**

* + - * 1. Add to the load balancer with the following inbound NAT rule:

Name: **20533E0401-ilb-natrulerdpvm1**

Frontend IP address: **LoadBalancerFrontEnd**

Service: **Custom**

Protocol: **TCP**

Port: **33891**

Associated to: **20533E0401-avset (availability set)**

Target virtual machine: **20533E0401-vm1**

Network IP configuration: **ipconfig1**

Port mapping: **Custom**

Floating IP (direct server return): **Disabled**

Target port: **3389**

**Note:** This configuration will allow you to connect to both Azure VMs via RDP even though they do not have directly assigned public IP address.

* + - * 1. On the 20533E0401-ilb blade, review the **Overview** section and identify the public IP address assigned to the load balancer. Note that at this point, you will not be able to connect to the two virtual machines in the backend pool, because they are not running a web server and the connectivity is additionally restricted by default network security group settings and the operating system-level firewall. You will change these settings later in this lab.
      1. **Task 3: Install and configure IIS on Azure VMs by using DSC and Windows PowerShell**
         1. On MIA-CL1, start File Explorer and browse to the **F:\Labfiles\Lab04\Starter** folder.
         2. In the **F:\Labfiles\Lab04\Starter** folder, right-click on the **IISInstall.ps1** file and select **Edit** from the right-click menu. This will open the file in the **Windows PowerShell ISE**.
         3. Review the content of the file. Note that this is a DSC configuration that controls the installation of the Windows Server 2016 Web-Server role.
         4. Close the Windows PowerShell ISE window.
         5. In the File Explorer, right click on the **F:\Labfiles\Lab04\Starter\Deploy- 20533E0401DSC.ps1** file and select **Edit** from the right-click menu. This will open the file in the **Windows PowerShell ISE** window with the current directory set

to **F:\Labfiles\Lab04\Starter**.

* + - * 1. Review the content of the script. Note the variables that it uses, including the storage account and its key. The script first retrieves the storage account from the resource group, and then publishes the DSC configuration defined in the **Install.ps1** into it, placing it in the default DSC container named **windows-powershell-dsc**, stores the resulting module URL in a variable, and then sets the Azure Agent VM DSC extension on two virtual machines deployed by the provisioning script by referencing that URL. The script generates a shared access signature token that provides read only access to the blob representing the DSC configuration archive.
        2. Start the execution of the script. When prompted, sign in with the username and the password of an account that is either a Service Administrator or a Co-Admin of your Azure subscription. Wait until the script completes.
        3. On MIA-CL1, open Internet Explorer and navigate to the Azure portal.
        4. Initiate a Remote Desktop session to **20533E0401-vm0** from the Azure portal.
        5. When prompted to enter credentials to connect, type **Student** as the user name and **Pa55w.rd1234** as the password.
        6. Once you establish a Remote Desktop session to the VM, in the **Server**

**Manager** window, verify that IIS appears in the left pane, indicating that the Web Server (IIS) server role is installed.

* + - * 1. Repeat steps 9 through 11 for the other virtual machine, **20533E0401-vm1**.
        2. After completing the tasks, switch back to your lab computer MIA-CL1. Leave both Remote Desktop sessions open.
      1. **Task 4: Test the DSC configuration and virtual machine availability**
         1. From the Azure portal within the Internet Explorer window on MIA-CL1, create a new inbound security rule for the **20533E0401-web-nsg** security group with the following settings:

Source: **Any**

Source port ranges: **Any**

Destination: **Any**

Destination port ranges: **80**

Protocol: **TCP**

Action: **Allow**

Priority: **1100**

Name: **allow-http**

* + - * 1. From the Azure portal, identify the IP address of the **20533E0401-ilb** load balancer.
        2. From MIA-CL1, open a new InPrivate Browsing Internet Explorer session and browse to this IP address.
        3. Verify that you can access the default IIS webpage and close the InPrivate Browsing session.
        4. From the Remote Desktop sessions to two Azure VMs, stop the **World Wide Web Publishing Service** service on both **20533E0401-vm0** and **20533E0401-vm1**
        5. From MIA-CL1, open a new InPrivate Browsing Internet Explorer session.
        6. In the new InPrivate Browsing window, delete browsing history.
        7. Browse to the IP address of the **20533E0401-ilb** load balancer again and verify that you can no longer access the default IIS webpage.
        8. From the Remote Desktop session window, start the **World Wide Web Publishing Service** service on **20533E0401-vm0**.
        9. Once the service is running, switch back to MIA-CL1 and refresh the InPrivate Browsing Internet Explorer window. Verify that you can again access the default the default IIS webpage. Note that you might need to wait about a minute after you start the **World Wide Web Publishing Service** service.

**Note:** Optionally you can repeat this sequence, but this time stopping the **World Wide Web Publishing Service** on **20533E0401-vm0** and starting it on **20533E0401-vm1**. As long as

the service is running on at least one of the two virtual machines, you should be able to access the webpage.

**Result**: After completing this exercise, you should have created and configured a load balancer in front of two Azure VMs in the same availability set and implemented DSC-based configuration on these VMs.

 **Exercise 2: Implementing Storage Spaces–based volumes**



##### Scenario

To test provisioning of multi-disk volumes on Azure VMs, you want to create three new VM disks, attach them to the Azure VMs that will host the database tier of the A. Datum ResDev application, and then use Storage Spaces to create a new volume.

The main tasks for this exercise are as follows:

* + - 1. Attach VHDs to an Azure VM
      2. Configure a Storage Spaces simple volume
      3. Remove the lab environment.
      4. **Task 1: Attach VHDs to an Azure VM**
         1. On MIA-CL1, from the Azure portal in the Internet Explorer window, attach to the 20533E0401-vm2 virtual machine a managed data disks with the following settings:

Name: **20533E0401-vm2-data01**

Resource group: Select **20533E0401-LabRG** in the drop down list marked "Select Existing".

Account type: **Standard HDD**

Source type: **None (empty disk)**

Size: **128**

HOST CACHING: **None**

* + - * 1. On MIA-CL1, from the Azure portal in the Internet Explorer window, attach to the 20533E0401-vm2 virtual machine a managed data disks with the following settings:

Name: **20533E0401-vm2-data02**

Resource group: Select **20533E0401-LabRG** in the drop down list marked "Select Existing".

Account type: **Standard HDD**

Source type: **None (empty disk)**

Size: **128**

HOST CACHING: **None**

* + - 1. **Task 2: Configure a Storage Spaces simple volume**
         1. On MIA-CL1, switch to the Remote Desktop session to 20533E0401-vm2.
         2. While connected to 20533E0401-vm2, from the Server Manager window, create a storage pool named **StoragePool1** consisting of two newly attached disks.
         3. From the Server Manager window, create a new virtual disk

named **VirtualDisk1** using **StoragePool1** with the **Simple** storage layout, the **Fixed** provisioning type, and the maximum size.

* + - * 1. From the Server Manager window, create a new volume of maximum size, mount it as the F: drive and format it with NTFS and a default allocation unit.
        2. From the desktop of 20533E0401-vm2, open File Explorer and verify that there is a new drive F:.
        3. Close the Remote Desktop session to 20533E0401-vm2.
      1. **Task 3: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: After completing this exercise, you should have implemented Storage Spaces based volumes.

**Question** Why would you use Storage Spaces in an Azure VM considering that Azure already provides highly available storage built into a storage account?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 5: Implementing Azure App Service Lab: Implementing web apps

##### Scenario

The A. Datum Corporation's public-facing web app currently runs on an IIS web server at the company's chosen ISP. A. Datum wants to migrate this web app into Azure. You must test the Web Apps functionality by setting up a test A. Datum web app. An internal team provides you with a test web app to deploy. You must ensure that they can continue to stage changes to the test web app before deploying those changes to the public-facing site. A. Datum is a global company, so you also want to test Azure Traffic Manager, and show your organization's decision makers how it distributes traffic to instances close to users of the web app.

##### Objectives

After completing this lab, you will be able to:

* Create a new web app.
* Deploy a web app.
* Manage web apps.
* Implement Traffic Manager to load-balance web apps.

##### Lab Setup

Estimated Time: 60 minutes Virtual machine: **20533E-MIA-CL1** User name: **Student**

Password: **Pa55w.rd**

 **Exercise 1: Creating web apps**



##### Scenario

You must set up a test web app in Azure. As the first step in the setup process, you want to create a new web app. Later in this lab, you will deploy this web app to the test web app.

The main tasks for this exercise are as follows:

* + - 1. Create a web app
      2. Add a deployment slot
      3. Configure deployment credentials
      4. **Task 1: Create a web app**
         1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**.
         2. Open Microsoft Edge, browse to the Azure portal, and then sign in using the Microsoft account that is the Service Administrator of your subscription.
         3. To create a new web app, use the following information:

App name: any unique valid name

Resource Group: **20533E0501-LabRG**

Web Hosting Plan Name: **20533E0501LabPlan**

Location: an Azure region close to the lab location, in which you can create app service plans.

Pricing tier: **S1 Standard**

Application Insights: leave at its default value

* + - 1. **Task 2: Add a deployment slot**
         1. In the Azure portal, add a new deployment slot to the web app that you created in the first task, using the following information:

Name: **Staging**

Configuration Source: accept the default setting

* + - * 1. Open Windows PowerShell window and authenticate to your Azure subscription by signing in using the Microsoft account that is the Service Administrator of your subscription.
        2. If you have multiple subscriptions, select the target one by running the Azure PowerShell **Set-AzureRmContext** cmdlet.
        3. Use the Azure PowerShell **Get-AzureRmWebApp** and **Get- AzureRmWebAppSlot** cmdlets to identify the web app and staging slot that you created.
        4. Keep the Azure PowerShell window open.
      1. **Task 3: Configure deployment credentials**
         1. In the Azure portal, on the web app blade, set the following deployment credentials for the web app that you created in the first task:

FTP/Deployment User Name: a unique name

Password: **Pa55w.rd**

**Result**: After completing this exercise, you should have created a new web app in the Azure portal, and configured the new web app with deployment slots and deployment credentials.

 **Exercise 2: Deploying a web app**



##### Scenario

Now that you created a web app in Azure, and added a deployment slot for the web app, you can publish the internally developed web app that the A. Datum web-development team supplied. In this exercise, you will use a publishing profile in Visual Studio to connect to the new web app and deploy the web content.

The main tasks for this exercise are as follows:

* + - 1. Obtain a publishing profile
      2. Deploy a web app
      3. **Task 1: Obtain a publishing profile**
         1. From the Azure portal, download the publish profile for the Web app you created in Exercise 1.
         2. Open the web-application project stored

in **F:\LabFiles\Lab05\Starter\AdatumWebsite\AdatumWebsite.sln** in **Visual Studio**.

* + - * 1. Start debugging the web application, examine the web page automatically displayed on a new Microsoft Edge tab and then, close that tab.

**Note:** When you start the web application in Visual Studio, the web app runs in IIS Express on your local workstation.

* + - 1. **Task 2: Deploy a web app**
         1. In Visual Studio, start the Publish Wizard for the **AdatumWebsite** project, and then import the **.PublishSettings** file that you downloaded in task 1 of this exercise.
         2. Publish the new website to Azure.

**Note:** When the operation is complete, Microsoft Edge opens and displays the new web app hosted in Azure.

* + - * 1. Verify that A. Datum's web app opens in Microsoft Edge and then verify the web app's current address.
        2. Close Microsoft Edge.
        3. Leave Visual Studio open.

**Result**: After completing this exercise, you should have deployed a web app hosted in Azure.

 **Exercise 3: Managing web apps**



##### Scenario

The web-deployment team created an updated style sheet for the A. Datum's test web app. You have to demonstrate how you can deploy these changes to a staging slot, and then test them, before you deploy to the production A. Datum web app. In this exercise, you will upload the new web app to the staging slot that you created in Exercise 1, and you then will swap the new version of the web app into the production slot.

The main tasks for this exercise are as follows:

* + - 1. Deploy a web app for staging
      2. Swap deployment slots
      3. Roll back a deployment
      4. **Task 1: Deploy a web app for staging**
         1. In the Azure portal, download a publishing profile for the Staging slot for your web app.
         2. Open the project

in **F:\LabFiles\Lab05\Starter\NewAdatumWebsite\AdatumWebsite.sln** in **Visual Studio**.

* + - * 1. Start the web app publishing process and import the staging publishing profile that you downloaded in the first step of this task.
        2. Publish the new web app to the Staging slot.
        3. Close Microsoft Edge.
        4. Leave Visual Studio open.
      1. **Task 2: Swap deployment slots**
         1. In Microsoft Edge, in the Azure portal, navigate to the web app that you created in Exercise 1.
         2. From the Azure portal, use the **URL** link for your web app to open it in another Microsoft Edge tab.
         3. Notice that the color scheme has not changed, because the Web app with the new color scheme is still in the staging slot. Close the Microsoft Edge tab displaying the A. Datum web app.
         4. From the web app blade in the Azure portal, swap the staging and production web- app slots.
         5. When the swap completes, use the **URL** link again to browse to the web app and notice that the color scheme has changed.
         6. Close the Microsoft Edge tab that displays the A. Datum's web app.
      2. **Task 3: Roll back a deployment**
         1. In the Azure portal, swap the staging and production slots again.

**Note:** By swapping the slots a second time, you simulate a deployment rollback.

* + - * 1. When the swap is complete, browse to the web app. Notice that the color scheme has reverted to the original one.
        2. Close the Microsoft Edge tab displaying the A. Datum web app.

**Result**: After completing this exercise, you should have an updated web app in the staging slot and have tested the slot swap functionality.

 **Exercise 4: Implementing Traffic Manager**



##### Scenario

Because A. Datum has customers around the globe, you must ensure that the A. Datum web apps perform well when serving requests from multiple locations around the world. You must evaluate Traffic Manager to verify that web content is served from a location that is close to customers. To accomplish this, you will set up a deployment of Traffic Manager serving content of a test web app from two different Azure regions.

The main tasks for this exercise are as follows:

* + - 1. Deploy a web app to another region
      2. Create a Traffic Manager profile
      3. Add endpoints, and configure Traffic Manager
      4. Test Traffic Manager
      5. Remove the lab environment
      6. **Task 1: Deploy a web app to another region**
         1. In Azure PowerShell, identify the settings of your test web app by using the **Get- AzureRmWebApp** cmdlet. Note the name of the web app and its location.
         2. Choose an Azure region where you can provision an Azure web apps and which is different from the location of the original web app, preferably on a different continent. This will become the ***SecondLocation***.
         3. Use the **New-AzureRmResourceGroup** cmdlet to create a new resource group named **20533E0502-LabRG** located in the ***SecondLocation***.
         4. Use the **New-AzureRmAppServicePlan** cmdlet to create a new App Service plan named **20533E0502LabPlan** with the Standard pricing tier in the resource

group **20533E0502-LabRG** and the ***SecondLocation***.

* + - * 1. Use the **New-AzureRmWebApp** cmdlet to create a new web app. Use the following information for the web app:

Resource group: **20533E0502-LabRG**

Name: a unique name (use the **Test-AzureRmDnsAvailability** cmdlet to identify it)

Service plan: **20533E0502LabPlan**

Location: ***SecondLocation***

* + - * 1. In the Azure portal, download a publishing profile for the web app you just created.
        2. Open the project

in **F:\LabFiles\Lab05\Starter\AdatumWebsite\AdatumWebsite.sln** in **Visual Studio**.

* + - * 1. Start the Publish Web Wizard, and then import the publish settings file that you just downloaded.
        2. Publish the web app, and then close Microsoft Edge and Visual Studio.
      1. **Task 2: Create a Traffic Manager profile**
         1. In the Azure portal, create a new Traffic Manager profile by using the following information:

Name: a unique domain name

Routing Method: **Performance**

Resource Group: **20533E0503-LabRG**

Resource group location: ***the Azure region in which you provisioned the first web app***

* + - 1. **Task 3: Add endpoints, and configure Traffic Manager**
         1. From the Traffic Manager profile blade in the Azure portal, add the web apps that you created in Exercise 1 and Exercise 4 as the Traffic Manager profile endpoints.
         2. From the Traffic Manager profile blade, modify the profile configuration by setting the DNS TTL value to 30 seconds.
      2. **Task 4: Test Traffic Manager**
         1. From the Azure portal, use the DNS name of the Traffic Manager profile to browse to the web app instance corresponding to the closest endpoint.
         2. Use the **nslookup** command to resolve the DNS name of the Traffic Manager profile.

**Note:** Review the DNS records listed in the output of the command to identify the web app instance returned from the Traffic Manager profile

* + - * 1. In the Azure portal, disable the Traffic Manager endpoint representing the web app instance you identified in the previous step.
        2. Use the **nslookup** command again to resolve the DNS NAME for your Traffic Manager profile. The results should differ from those in step 2.

**Note:** You might have to wait in order for the endpoint state change to take effect. Wait about 1 minute and re-run the **nslookup** command.

* + - 1. **Task 5: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: After completing this exercise, you should have implemented two Azure web apps and a Traffic Manager profile configured to distribute requests between them.

**Question** In Exercise 2, you deployed the A. Datum production web app to Azure. In Exercise 3, you deployed a new version of the site to a staging slot. How can you tell, within Microsoft Edge, which is the production site and which is the staging site?

**Question** At the end of Exercise 4, you used an FQDN within the trafficmanager.net domain to access your web app. How can you use your own registered domain name to access this web app?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 6: Planning and implementing Azure Storage Lab : Planning and implementing Azure Storage

##### Scenario

The IT department at A. Datum Corporation uses an asset management application to track IT assets such as computer hardware and peripherals. The application stores images of asset

types and invoices for asset purchases As part of A. Datum's evaluation of Azure, you need to test migration of these images and invoice documents to Azure storage. A. Datum also wants to evaluate Azure File storage for providing SMB 3.0 shared access to invoices. Currently, corporate file servers host this content.

##### Objectives

After completing this lab, you will be able to:

* Creating and configuring Azure Storage.
* Use Azure file storage.

##### Lab Setup

Estimated Time: 50 minutes Virtual machine: **20533E-MIA-CL1**

* User name: **Student**
* Password: **Pa55w.rd**

Before starting this lab, ensure that you have performed the "Preparing the environment" demonstration tasks at the beginning of the first lesson in this module and that the setup script has completed.

 **Exercise 1: Creating and configuring Azure Storage**



##### 2.17.1. Scenario

A. Datum currently stores images for IT assets on the on-premises file servers. As part of your Azure evaluation, you want to test storing these images as blobs in Azure storage so that a new Azure-based version of the asset management application can easily access them.

The main tasks for this exercise are as follows:

1. Create a storage account
2. Install AzCopy
3. Use AzCopy to upload blobs
   * + 1. **Task 1: Create a storage account**
          1. Ensure that you are signed in to the MIA-CL1 virtual machine as **Student** with the password **Pa55w.rd** and that the setup script that you ran in the "Preparing the environment" demonstration has completed.
          2. Use Internet Explorer to sign in to the Azure portal by using the Microsoft account that is the Service Administrator or a Co-Administrator of your Azure subscription.
          3. Create a new storage account with the following settings:

Subscription: the name of your Azure subscription

Resource group: ensure that **Create new** is selected and, in the textbox below, type **20533E0602-LabRG**.

Name: a valid, unique name consisting of between 3 and 24 lower case characters or digits

Performance: **Standard**

Account kind: **Storage (general purpose v1)**

Location: the same Azure region that you chose when running the provisioning script at the beginning of this module

Replication: **Locally-redundant storage (LRS)**

Secure transfer required: **Disabled**

Allow access from: **All networks**

Hierarchical namespace: **Disabled**

1. After the storage account is provisioned, create a blob container named **asset-images** with private access.
   * + 1. **Task 2: Install AzCopy**
          1. Download and install AzCopy from [**http://aka.ms/AzCopy**.](http://aka.ms/AzCopy) Note that this page also includes documentation and examples for using AzCopy.
          2. Start Windows PowerShell ISE as Administrator.
          3. In the console pane of Windows PowerShell ISE, change the current directory by running:

Set-Location -Path 'C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy'

* + - * 1. Test the installation by running the following command at a command prompt:

.\AzCopy /?

* + - * 1. Keep the Windows PowerShell ISE window open for the next task.
      1. **Task 3: Use AzCopy to upload blobs**
         1. In the Windows PowerShell ISE window, type the following in the script pane:



* + - * 1. In the Azure portal, copy the name of the Storage account you created earlier in this exercise.
        2. In the script pane of the Windows PowerShell ISE, replace the ***<storage-account- name>*** entry with the storage account name you copied from the Azure portal.
        3. In the Azure portal, copy the first access key of the Storage account.
        4. In the script pane of the Windows PowerShell ISE, replace the ***<access-key>*** entry with the storage account key you copied from the Azure portal.
        5. Execute the command in the script pane and wait for the command to complete. Review the file transfer information.
        6. In the Azure portal, navigate to the **asset-images** container blade and verify that the container contains six blobs.

**Result**: At the end of this exercise, you should have created a new Azure storage account with a container named **asset-images** and copied files from your local computer to that container by using the **AzCopy** utility.

 **Exercise 2: Using Azure File storage**



##### Scenario

A. Datum currently stores invoices for IT assets on the on-premises file servers. As part of your evaluation of Azure, you want to test an upload of these files to a file share in your Azure storage account.

The main tasks for this exercise are as follows:

1. Create a file share and upload files
2. Access a file share from a VM
   * + 1. **Task 1: Create a file share and upload files**
          1. Switch to the Windows PowerShell ISE window and run the **Add- AzureRmAccount** cmdlet. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Microsoft Azure subscription.
          2. From the Windows PowerShell ISE, open **F:\Labfiles\Lab06\Starter\New- 20533E06FileShare.ps1**.
          3. In the script pane, in the **$storageAccountName** variable declaration at the beginning, replace the ***<storage-account-name>*** value with the name of the Azure storage account that you created in the previous exercise.
          4. Review the script, noting that it:

Sets the values of variables named **$shareName** and **$directoryName** for the file share and the directory to create in the Azure Storage account

Uses the **Get-AzureRmStorageAccountKey** cmdlet to retrieve the access key for your storage account.

Uses the **New-AzureStorageContext** cmdlet to create a security context for connections to the target storage account based on the key you retrieved

Uses the **New-AzureStorageShare** cmdlet to create an Azure Storage account file share

Uses the **New-AzureStorageDirectory** cmdlet to create a directory in the share

Sets the location of the folder hosting source files to be copied to the Azure Storage file share directory

Loops through the files in the source folder and uses the **Set- AzureStorageFileContent** cmdlet to copy each of them the folder in the Azure file share.

* + - * 1. Run the script to upload the files.
        2. Observe the script as it runs, and then view the output. When you finish, close Windows PowerShell ISE.
      1. **Task 2: Access a file share from a VM**
         1. Connect to the 20533E0601-vm1 VM in your Azure subscription via Remote Desktop by using the following credentials:

User name: **Student**

Password: **Pa55w.rd1234**

* + - * 1. Once connected, on 20533E0601-vm1, turn off **IE Enhanced Security Configuration** for administrators.
        2. Use Internet Explorer to navigate to the Azure portal and, when prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
        3. In the Azure portal, navigate to the **assets** file service blade of the storage account that you created in the previous exercise, click **Connect** and copy the Windows PowerShell script from the **Connecting from Windows** section that needs to be run in order to connect to the **assets** file share from a Windows computer.
        4. In the Remote Desktop session, start Windows PowerShell ISE and paste the script into the script window.
        5. In Windows PowerShell ISE, execute the command and verify it completed successfully and created a **Z:** drive mapping.
        6. In the Command Prompt window, enter the following command to view the contents of the invoices folder in drive Z:, which is now mapped to the assets file share that you created in the previous task:

Get-ChildItem -Path 'Z:\invoices'

* + - * 1. Verify that invoices are listed.
        2. Sign out of the 20533E0601-vm1 VM to end the remote desktop session.
      1. **Task 3: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: At the end of this exercise, you should have created an Azure storage account file share named **assets** that contains a folder named **invoices** with copies of invoice documents. You should have also mapped a drive from an Azure VM to the Azure storage account file share.

**Question** The asset management application stores images of hardware components as blobs and invoices as files. If the application also needed to search the location of each asset by using an asset type, a unique asset number, and a text description of the location, what storage options should you consider?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 7: Implementing containers in Azure Lab A: Implementing containers on Azure VMs

##### Scenario

Adatum Corporation plans to implement some of its applications as Docker containers on Azure VMs. To optimize this implementation, you intend to combine multiple containers by using Docker Compose. A. Datum would also like to deploy its own private Docker registry in Azure to store containerized images. Your task is to test the functionality of tools that facilitate deployment of Docker hosts and Docker containers. You also need to evaluate Azure Container Registry.

##### Objectives

After completing this lab, you will be able to:

* + - * Deploy containers to Azure VMs
      * Deploy multicontainer applications with Docker Compose to Azure VMs
      * Implement Azure Container Registry

##### Lab Setup

Estimated Time: 30 minutes

Virtual Machine: **20533E-MIA-CL1**

User Name: **Student**

Password: **Pa55w.rd**

Before starting this lab, ensure that you have performed the "Preparing the Environment" demonstration tasks at the beginning of the first lesson in this module, and that the setup script has completed.

 **Exercise 1: Implementing Docker hosts on Azure VMs**



##### Scenario

To test the planned deployment, you must identify the methods that would allow you to deploy Docker containers to Azure VMs. To accomplish this, you want to test deployment of a sample containerized nginx web server, available from Docker Hub.

Note: The Microsoft Azure portal is continually improved, and the user interface might have been updated since this lab was written. Your instructor will make you aware of any differences between the steps described in the lab and the current Azure portal user interface.

The main tasks for this exercise are as follows:

* + - 1. Connect to an Azure VM running Linux
      2. Install Docker and deploy a container to an Azure VM
      3. **Task 1: Connect to an Azure VM running Linux**
         1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**, and that the **Add-20533EEnvironment** script successfully completed. The script provisions a Linux Azure VM running Docker.
         2. Start Microsoft Edge and browse to the Azure portal. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         3. In the Azure portal, start **Bash (Linux)** session in **Cloud Shell**. If prompted, create a new storage account with the following settings:

Subsciption: the same Azure subscription you chose when running the provisioning script at the beginning of this module

Cloud Shell region: the same Azure region you chose when running the provisioning script at the beginning of this module

Resource group: **20533E0701-LabRG**

Storage account: create a new storage account

File share: create a new file share

**Note:** You can open **Cloud shell** in a new window for ease of use via [**https://shell.azure.com**](https://shell.azure.com/)

* + - * 1. From the Cloud Shell pane, use Azure CLI to identify the fully qualified domain name associated with the public IP address of the Linux Azure VM Docker host

named **20533E0701-vm0**, which was created by the provisioning script.

* + - * 1. From the Cloud Shell pane, establish an SSH session as **student** with the

password **Pa55w.rd1234** to the Linux Azure VM Docker host **20533E0701-vm0** using the DNS name you identified in the previous step.

* + - 1. **Task 2: Install Docker and deploy a container to an Azure VM**
         1. Within the SSH session to the Linux Azure VM Docker host, within the Cloud Shell pane, use the **apt install docker.io** command to install the Docker CE.
         2. Within the SSH session to the Linux Azure VM Docker host, within the Cloud Shell pane, use the **docker run** command to start an **nginx** container from the Docker Hub, making it available via TCP port 80.
         3. Monitor the progress of the container deployment. Verify the successful outcome, by running the **docker ps** command.
         4. From MIA-CL1, start Microsoft Edge and browse to the fully qualified DNS name you obtained in the previous task. Verify that Microsoft Edge displays the **Welcome to nginx!** page

**Result**: After you complete this exercise, you should have successfully run a sample containerized web server nginx on the Linux Azure VM Docker host.

 **Exercise 2: Deploying multi-container applications with Docker Compose to Azure VMs**



##### Scenario

You intend to implement some A. Datum applications by using multiple containers. To accomplish this, you will test deployment of containers by using Docker Compose.

The main tasks for this exercise are as follows:

* + - 1. Install docker-compose and create a compose file
      2. Deploy the containers with docker-compose to an Azure VM
      3. **Task 1: Install docker-compose and create a compose file**
         1. In the Cloud Shell pane, within the SSH session to the Azure VM Docker host, install Docker Compose by running the **apt install docker-compose** command.
         2. In the Cloud Shell pane, within the SSH session to the Azure VM Docker host, create a new file named **docker-compose.yml** with the following content (you can find the file in the **F:\Labfiles\Lab07\Solution** folder):



* + - 1. **Task 2: Deploy the containers with docker-compose to an Azure VM**
         1. In the Cloud Shell pane, within the SSH session to the Azure VM Docker host, deploy containers defined in the **docker-compose.yml** file.
         2. Monitor the progress of the container deployment. Use the **docker ps** command to verify the successful outcome.
         3. On MIA-CL1, start Microsoft Edge and browse to the port 8080 on the target host via the same URL you used in the previous exercise. Verify that Microsoft Edge displays the initial Wordpress setup page.

**Note**: This is possible because the template used to provision the deployment of the Linux Azure VM Docker host included a network security group with a rule that allows inbound traffic on TCP port 8080. If that was not the case, you would need to add this rule.

* + - * 1. Exit the SSH session and close the Cloud Shell pane.

**Result**: After you completed this exercise, you should have successfully implemented a multi- container application by using Docker Compose.

 **Exercise 3: Implementing Azure Container Registry**



##### Scenario

Now that you have successfully implemented a Docker host in an Azure VM and deployed containerized images from Docker Hub, you want to test the setup and image management by using Container Registry. In your tests, you will use a sample image available from Docker Hub. You will start by creating a container registry. Next, you will download the sample image to your lab computer and upload it to the newly created private registry. Finally, you will deploy the image from the private registry to the Docker host in Azure VM.

The main tasks for this exercise are as follows:

* + - 1. Create an Azure Container Registry
      2. Identify Azure Container Registry authentication settings.
      3. Push an image to Azure Container Registry.
      4. Download and deploy images from the Azure Container Registry
      5. Remove the lab environment
      6. **Task 1: Create an Azure Container Registry**
         1. In the Azure portal in the Microsoft Edge window, create a new **Azure Container Registry** instance with the following settings:

Registry name: a unique name consisting of between 5 and 50 alphanumeric characters

Subscription: the name of the Azure subscription you are using in this lab

Resource group: **20533E0702-LabRG**

Location : the same Azure region that you chose when running the provisioning script at the beginning of this module

Admin user: **Enable** (this will allow you to use the registry name as username and admin user access key as its password to connect from the Docker host to the registry by using the **docker login** command)

SKU : **Basic**

* + - * 1. Wait for the operation to complete.
      1. **Task 2: Identify Azure Container Registry authentication settings.**
         1. If needed, on MIA-CL1, in the Azure portal, in the Microsoft Edge window, start Cloud Shell.
         2. Use Azure CLI to identify the value of the **Login server** property of the newly created Azure Container Registry instance and one of the passwords necessary to access it.
      2. **Task 3: Push an image to Azure Container Registry.**
         1. From the Cloud Shell pane, connect via SSH to the Linux Azure VM Docker host.
         2. Within the SSH session to the Linux Azure VM Docker host, within the Cloud Shell pane, use the **docker login** command to sign into the newly created Azure Container Registry instance. To sign in, use the name of the registry, the name of the **Login server**, and the password you identified in the previous task.
         3. Ensure that you receive the **Login succeeded** message. Next, pull the **microsoft/aci- helloworld** image from Docker Hub by running the **docker pull** command.
         4. Wait for the image to be downloaded to the Azure VM Docker host. Next, tag the image with the Azure Container registry name by running the **docker tag** command with the ***login-server*/aci-helloworld:v1** parameter, replacing the ***login-server*** entry with the value you identified in the previous task.
         5. Push the tagged image to the Azure Container registry by running the **docker push** command with the ***login-server*/aci-helloworld:v1** parameter, replacing the ***login-server*** entry with the value you identified in the previous task.
         6. Wait for the image to be pushed to the registry. Next, on MIA-CL1, in the Azure portal, from the container registry blade, verify that the Azure Container registry contains

the **aci-helloworld** repository

* + - 1. **Task 4: Download and deploy an image from the Azure Container Registry**
         1. Within the SSH session to the Linux Azure VM Docker host, within the Cloud Shell pane, pull the newly tagged image from the Azure Container registry, by running the **docker pull** command with the ***login-server*/aci-helloworld:v1** parameter, replacing the ***login-server*** entry with the value you identified earlier in this exercise.
         2. Note that, in this case, the image does not need to be downloaded, since it is already present on the target Docker Azure VM.
         3. Deploy the image downloaded from the Azure Container registry, by running the **docker run** command with the ***login-server*/aci-helloworld:v1** parameter,

replacing the ***login-server*** entry with the value you identified earlier in this exercise. Deploy the web server running in the container such that it is accessible via port 8081.

* + - * 1. Run the **docker ps** command to verify that the image has been successfully deployed.
        2. Note that the output includes the tagged image.
        3. Terminate the SSH session then close the Cloud Shell pane.
      1. **Task 5: Connect to a container running on a Docker host**
         1. In order to connect to the newly provisioned container, you will need to modify the network security group associated with the network interface of the Linux Azure VM Docker host. On MIA-CL1, in the Azure portal window, navigate to the **20533E0701- web-nsg** blade.
         2. From the **20533E0701-web-nsg** blade, add an inbound security rule with the following settings:

Source: **Any**

Source port ranges: **\***

Destination: **Any**

Destination port ranges: **8081**

Protocol: **TCP**

Action: **Allow**

Priority: **1300**

Name: **custom-allow-8081**

* + - 1. Wait for the operation to complete. On MIA-CL1, start Microsoft Edge and browse to the port 8081 on the target host via the same URL you used in the previous exercise. Verify that Microsoft Edge displays the **Welcome to Azure Container Instances!** page.

**Result**: After you complete this exercise, you should have successfully tested image deployment by using Azure Container Registry.

**Question**

Which method would you use when deploying Docker hosts on Azure VMs?

**Question**

What authentication and authorization method do you intend to use when implementing Azure Container Registry?

#### Lab B: Implementing Azure Container Service (AKS)



##### Scenario

A. Datum is considering implementing containers on a larger scale by leveraging the capabilities that AKS offers. You intend to test its functionality. You want to test load balancing and scaling of a sample containerized application.

##### Objectives

After completing this lab, you will be able to:

* Create an AKS cluster.
* Manage the AKS cluster.

##### Lab Setup

Estimated Time: 30 minutes

Virtual Machine: **20533E-MIA-CL1**

User Name: **Student**

Password: **Pa55w.rd**

 **Exercise 1: Creating an AKS cluster**



##### Scenario

You must start by identifying the prerequisites for deploying an AKS cluster. You plan to install the cluster by using Azure CLI.

The main tasks for this exercise are as follows:

* + - 1. Create an AKS cluster
      2. Connect to the AKS cluster.
      3. **Task 1: Create an AKS cluster**
         1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**. In the Azure portal, in the Microsoft Edge window, start the Bash prompt in **Cloud Shell**.
         2. From the bash prompt, in the Cloud Shell pane, create a new resource group named **20533E0703-LabRG** in the Azure region that you chose when running the provisioning script at the beginning of this module.
         3. From the bash prompt, in the Cloud Shell pane, use the **az aks create** command with the following parameters to create a new Kubernetes cluster:

--resource-group: **20533E0703-LabRG**

--name: **20533E0703-k8scluster**

--generate-ssh-keys

--node-count: **1**

--node-vm-size: the VM size you chose when running the provisioning script at the beginning of this module

* + - 1. Wait for the deployment to complete.
      2. **Task 2: Connect to the AKS cluster.**
         1. Download and configure the credentials to access the AKS cluster, by running the **az aks kubernetes get-credential** command with the following parameters from the bash prompt in the Cloud shell pane:

--resource-group: **20533E0703-LabRG**

--name **20533E0703-k8scluster**

* + - * 1. Verify connectivity to the AKS cluster, by running the **kubectl get nodes** command from the bash prompt in the Cloud Shell pane
        2. Review the output and verify that the agent nodes are reporting the **Ready** status.

**Result**: After you complete this exercise, you should have successfully deployed a new AKS cluster.

 **Exercise 2: Managing an AKS cluster**



##### Scenario

With the new AKS cluster running, you must connect to it, deploy a sample containerized application, and validate its availability and resiliency by testing clustering features such as scaling and load balancing.

The main tasks for this exercise are as follows:

* + - 1. Deploy a containerized application to the AKS cluster
      2. Manage deployment of a containerized application on the AKS cluster
      3. Remove the lab environment.
      4. **Task 1: Deploy a containerized application to the AKS cluster**
         1. In the Azure portal, in the Microsoft Edge window, in the Cloud shell pane, create a deployment named **nginx-20533e0703** using the nginx image from the Docker Hub by running the **kubectl run** command with the following parameters:

--image: **nginx**

--replicas: **1**

--port: **80**

* + - * 1. Verify that a Kubernetes pod has been created by running **kubectl get pods** command from the bash prompt in the Cloud Shell pane.
        2. Identify the state of the deployment by running **kubectl get deployment** command from the bash prompt in the Cloud Shell pane.
        3. Make the deployment **nginx-20533e0703** available from Internet by running **kubectl expose** command from the bash prompt in the Cloud Shell pane with the following parameters:

--port: **80**

--type: **LoadBalancer**

* + - * 1. Identify whether the public IP address has been provisioned by running **kubectl get services** command from the bash prompt in the Cloud Shell pane.
        2. Repeat step 5 until the value in the **EXTERNAL-IP** column for **nginx-**

**20533e0703** changes from **<pending>** to a public IP address. At that point, note the public IP address in the **EXTERNAL-IP** column for **nginx-20533e0703**.

* + - * 1. Start Microsoft Edge and browse to the IP address you obtained in the previous step. Verify that Microsoft Edge displays the **Welcome to nginx!**
      1. **Task 2: Manage deployment of a containerized application on the AKS cluster**
         1. Scale the deployment **nginx-20533e0703** by running **kubectl scale** command from the bash prompt in the Cloud Shell pane with the --replicas parameter set to **2**.
         2. Verify the outcome of scaling the deployment by running **kubectl get pods** command from the bash prompt in the Cloud Shell pane.
         3. In the output of the command you ran in the previous step, verify that the number of pods increased to 2.
         4. Delete the **nginx-20533e0703** deployment by running **kubectl delete** command from the bash prompt in the Cloud Shell pane.
         5. Verify that the command you ran in the previous step completed successfully by running **kubectl get deployment** command from the bash prompt in the Cloud Shell pane.
      2. **Task 3: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: After you complete this exercise, you should have successfully deployed a containerized workload to the new AKS cluster and validated its availability.

**Question**

What deployment methodology would you choose when deploying AKS clusters?

**Question**

What are the primary advantages of using AKS for container orchestration?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 8: Planning and implementing backup and disaster recovery

**Lab : Implementing Azure Backup and Azure Site Recovery**



##### Scenario

Adatum wants to evaluate the ability of Azure Backup to protect the content of on-premises computers and Azure IaaS virtual machines. A. Datum Corporation also wants to evaluate Azure Site Recovery for protecting Azure VMs.

##### Objectives

At the end of this lab, you will be able to:

* Implement Azure Backup.
* Implement Azure Site Recovery–based protection of Azure VMs

##### Lab Setup

Estimated Time: 60 minutes Virtual machine: **20533E-MIA-CL1** User name: **Student**

Password: **Pa55w.rd**

 **Exercise 1: Protecting data with Azure Backup**



##### Scenario

Adatum currently uses an on-premises backup solution. As part of your Azure evaluation, you want to test the protection of on-premises master copies of your image files and invoices by backing them up to the cloud. To accomplish this, you intend to use Azure Backup.

The main tasks for this exercise are as follows:

* + - 1. Create a recovery services vault
      2. Configure the vault for on-premises backup
      3. Install and configure the Azure Recovery Services Agent
      4. Create a backup schedule
      5. Run a backup
      6. Perform a restore
      7. Disable backups and delete the vault
      8. **Task 1: Create a recovery services vault**
         1. In Internet Explorer, open the Azure portal.
         2. Create a new recovery services vault with the following settings:

Name: **vault20533E0801**

Subscription: the name of your Azure subscription

Resource group: **20533E0801-LabRG**

Location: the same Azure region that you chose when running the provisioning script at the beginning of this module

* + - * 1. Wait until the vault is provisioned.
      1. **Task 2: Configure the vault for on-premises backup**
         1. In the Azure portal, in the newly created vault, configure the backup goal with the following settings:

Where is your workload running?: **On-premises**

What do you want to back up?: **Files and folders**

* + - * 1. Click **Prepare Infrastructure**.
      1. **Task 3: Install and configure the Azure Recovery Services Agent**
         1. Download the **Microsoft Azure Recovery Services Agent** from the Azure portal and install it on MIA-CL1 with the default settings. On the **Microsoft Update Opt-In** page of the installation wizard, select the **I do not want to use Microsoft Update** option.
         2. Download the vault credentials file from the Azure portal
         3. Register MIA-CL1 with the vault. Prior to registration, generate a passphrase and store it in the **F:\Labfiles\Lab08\Starter** folder.
         4. At the end of the registration process, start the Azure Backup console and leave it open for the next task.
      2. **Task 4: Create a backup schedule**
         1. Use Azure Backup to schedule a daily backup to run at 4:30 AM and protect the following subfolders in the **F:\Labfiles\Lab08\Starter** folder:

**asset-images**

**invoices**

* + - * 1. Keep the default values for the other backup settings.
      1. **Task 5: Run a backup**
         1. From the Microsoft Azure Backup console, run an on-demand backup.
         2. From the Azure portal, verify that MIA-CL1 is registered with the Recovery Services vault and note the most recent backup items, which should include files and folders on the F: drive.
      2. **Task 6: Perform a recovery**
         1. From the Microsoft Azure Backup console, initiate data recovery.
         2. From the Recover Data Wizard, mount the backed up volume.
         3. Use File Explorer to copy the content of the backed up directories to their original location, overwriting existing data
         4. Unmount the backed up volume.
      3. **Task 7: Disable backups and delete the Azure Recovery Services vault**
         1. From the Azure portal, in the Recovery Services vault, delete references to **mia-cl1.**
         2. From the Azure portal, delete the Recovery Services vault.

 **Exercise 2: Implementing protection of Azure VMs by using Azure Site Recovery**



##### Scenario

Adatum Corporation wants to test a disaster recovery of its Azure-based Azure VMs. As part of Adatum’s evaluation of integration with Microsoft Azure, you have been asked to use Site Recovery to configure the protection of your test Azure VM environment.

The main tasks for this exercise are as follows:

* + - 1. Create an Azure Recovery Services vault
      2. Configure Azure VM replication
      3. Review Azure VM replication settings
      4. Disable replication of an Azure VM and delete the Azure Recovery Services vault
      5. Remove the lab environment
      6. **Task 1: Create an Azure Recovery Services vault**
         1. On MIA-CL1, from the the Azure portal, create an Azure Recovery Services vault with the following settings:

Name: **vault20533E0802**

Subscription: the name of your Azure subscription

Resource group: **20533E0802-LabRG**

Location: an Azure region **different** from the one you chose when running the provisioning script at the beginning of this module

* + - * 1. Wait until the vault is provisioned.
      1. **Task 2: Configure Azure VM replication**
         1. On MIA-CL1, in the Azure portal, navigate to the blade of the newly provisioned Azure Recovery Services vault.
         2. Enable replication with the following settings:

Source: **Azure**

Source location: the same Azure region that you chose when running the provisioning script at the beginning of this module

Azure virtual machine deployment model: **Resource Manager**

Source resource group: **20533E0801-LabRG**

Replication policy: **Create new**

Name: **12-hour-retention-policy**

Recovery point retention: **12 Hours**

App consistent snapshot frequency: **6 Hours**

Multi-VM consistency: **No**

* + - * 1. Wait until the replication is enabled. This might take about 15 minutes.
      1. **Task 3: Review Azure VM replication settings**
         1. On MIA-CL1, in the Azure portal, navigate to the replicated item blade representing the Azure VM **20533E0801-vm1**.
         2. On the replicated item blade, review the **Health and status**, **Latest available recovery points**, and **Failover readiness** sections. Note the **Failover** and **Test Failover** entries in the toolbar. Scroll down to the **Infrastructure view**.
         3. If time permits, wait until the status of the Azure VM changes to **Protected**. This might take additional 15-20 minutes. At that point, examine the values of **RPO**, as well

as **Crash-consistent** and **App-consistent** recovery points.

* + - 1. **Task 4: Disable replication of an Azure VM and delete the Azure Recovery Services vault**
         1. In the Azure portal, disable replication of the Azure VM **20533E0801-vm1**.
         2. Wait until the replication is disabled. This might take about 15 minutes.
         3. From the Azure portal, delete the Recovery Services vault.

**Note**: You must ensure that the replicated item is removed first before you can delete the vault.

* + - 1. **Task 5: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: At the end of this exercise, you should have created an Azure Recovery Services vault in your subscription, downloaded vault credentials, and installed the Azure Recovery Services agent on the MIA-CL1 lab computer. You should have backed up the contents of the asset- images and invoices folders to the Recovery Services vault.

**Question** Why did the lab not include failover and failback?

**Question** If you wanted to protect Azure VMs that reside behind an Azure load balancer, how would you configure your Site Recovery solution?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 9: Implementing Azure Active Directory Lab: Implementing Azure AD

##### Scenario

The IT department at A. Datum Corporation currently uses AD DS, and a range of Active Directory-aware applications. While preparing for synchronizing its AD DS to Azure AD, A. Datum wants you to test some of the features of Azure AD. The company wants you to control access to third-party SaaS apps by using Azure AD users and groups. A. Datum also wants you to configure SSO to these apps and protect them by using Multi-Factor Authentication.

In addition to these tasks, A. Datum wants you to evaluate some of the advanced features Azure AD Premium offers. It also wants you join a Windows 10-based computer to an Azure AD tenant to prepare for implementing this configuration on all the Windows 10-based computers in the Research department.

##### Objectives

After completing this lab, you will be able to:

* Administer Azure AD.
* Configure SSO for Azure AD gallery applications.
* Configure multi-factor authentication for administrators.
* Use the advanced features offered by Azure AD Premium.
* Configure SSO from a Windows 10-based computer that is joined to Azure AD.

##### Lab Setup

Estimated Time: 60 minutes

Virtual Machine: **20533E-MIA-CL1**

Username: **Student**

Password: **Pa55w.rd**

Before you start this lab, ensure that you complete the tasks in the Preparing the environment demonstration, which is in the first lesson of this module. Also ensure that the setup script is complete.

 **Exercise 1: Administering Active AD**



##### Scenario

You want to test the functionality of Azure AD by first creating a new Azure AD tenant and enabling the Premium functionality. You then want to create some pilot users and groups in Azure AD. You plan to use the Azure portal interface and Microsoft Azure Active Directory Module for Windows PowerShell.

The main tasks for this exercise are as follows:

* + - 1. Create directories
      2. Activate Azure AD Premium P2 trial
      3. Manage users by using the Azure portal
      4. Manage groups by using the Azure portal
      5. Manage users and groups by using Azure PowerShell
      6. **Task 1: Create directories**
         1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**.
         2. Start Microsoft Edge, browse to the Azure portal at [**http://portal.azure.com**](http://portal.azure.com/)and then, when prompted, and then sign in using the Microsoft account that is the Service Administrator of your subscription.
         3. Add a directory by using the following settings:

Organization name: **Adatum**

Initial domain name: a unique, valid name

Country or region: **United States**

* + - * 1. Leave Microsoft Edge open and wait until the Azure Active Directory tentant is provisioned. Note the unique name you specified, since you will need it later in this task.
      1. **Task 2: Activate Azure AD Premium P2 trial**
         1. In the Azure portal, navigate to the **Adatum** directory.
         2. Activate the Azure AD Premium P2 trial.
      2. **Task 3: Manage users by using the Azure portal**
         1. Create a user in the Adatum directory with the following settings:

Name: **Remi Desforges**

User name: [**rdesforges@*domain-name*.onmicrosoft.com**](mailto:rdesforges@domain-name.onmicrosoft.com)where ***domain-name*** is the name you assigned to the Azure Active Directory tenant in the first task of this exercise

First Name: **Remi**

Last Name: **Desforges**

* + - * 1. Note the new password.
        2. Create another user in the Adatum directory with the following settings:

Name: **Karen Gruber**

User name: [**kgruber@*domain-name*.onmicrosoft.com**](mailto:kgruber@domain-name.onmicrosoft.com)where ***domain-name*** is the name you assigned to the Azure Active Directory tenant in the first task of this exercise

First Name: **Karen**

Last Name: **Gruber**

Directory role: **Global administrator**

* + - * 1. Note the new password.
        2. Open an InPrivate Microsoft Edge window, navigate to the Azure portal, sign in

as **Remi Desforges**, when prompted, change the password to a new value and then sign-out and close the InPrivate Microsoft Edge window. Take a note of the new password.

* + - * 1. Open an InPrivate Microsoft Edge window, navigate to the Azure portal, sign in as **Karen Gruber**, when prompted, change the password to a new value and then sign-out and close the InPrivate Microsoft Edge window. Take a note of the new password.
        2. Click **Sign out** and close the in-private session of Microsoft Edge.
      1. **Task 4: Manage groups by using the Azure portal**
         1. From the Azure portal, assign an Azure Active Directory Premium P2 license to your user account in the **Adatum** Azure AD.
         2. From the Azure portal, enable **self-service group management** and allow users to create security groups.
         3. Create the following group in the Adatum directory:

Name: **Sales**

Description: **Sales employees**

Membership type: **Assigned**

Enable Office features?: **No**

* + - * 1. Add **Remi Desforges** to the **Sales** group.
        2. Create the following group in the Adatum directory:

Name: **Marketing**

Description: **Marketing employees**

Membership type: **Assigned**

Enable Office features?: **No**

* + - * 1. Add **Karen Gruber** to the **Marketing** group.
        2. Create the following group in the Adatum directory:

Name: **Sales and Marketing**

Description: **Sales and Marketing employees**

Membership type: **Assigned**

Enable Office features?: **No**

* + - * 1. Add the **Sales** and **Marketing** groups to the **Sales and Marketing** group.
      1. **Task 5: Manage users and groups by using Azure PowerShell**
         1. Start **Windows PowerShell ISE** as an administrator.
         2. Open **F:\Labfiles\Lab09\Starter\Set-20553D0901Lab.ps1**.
         3. In the PowerShell ISE, in the command prompt pane, enter the following command, and then press Enter:

Connect-MsolService

* + - * 1. When prompted, sign in as Karen Gruber.
        2. In the PowerShell ISE, in the script pane, locate the following code:



* + - * 1. Replace ***<#Copy your Azure Directory domain name here#>*** with the unique name you used to specify the DNS domain name of the Adatum Azure AD tenant. In the Windows PowerShell ISE, in the script pane, select the code that you just edited. On the toolbar, click the **Run Selection** button and wait for the script to complete.
        2. In the PowerShell ISE, in the command prompt pane, run the following command to list all the users:

Get-MsolUser

* + - * 1. Create a new group by running the following command:

New-MsolGroup -DisplayName "Azure team" -Description "Adatum Azure team users"

* + - * 1. In the PowerShell ISE, in the command prompt pane, enter the following command, and then press Enter to list all the groups:

Get-MsolGroup

* + - * 1. In the PowerShell ISE, in the script pane, locate the following code, and then select it:

$group = Get-MsolGroup | Where-Object {$\_.DisplayName -eq "Azure team"}

* + - * 1. On the toolbar, click the **Run Selection** button and wait for the script to complete.
        2. In the PowerShell ISE, in the Script pane, locate the following code and select it:

$user = Get-MsolUser | Where-Object {$\_.DisplayName -eq "Mario Ledford"}

* + - * 1. On the toolbar, click the **Run Selection** button, and wait for the script to complete.
        2. In the PowerShell ISE, in the Script pane, locate the following code and select it:



* + - * 1. On the toolbar, click the **Run Selection** button, and wait for the script to complete.
        2. In the PowerShell ISE, in the script pane, locate the following code and select it:

Get-MsolGroupMember -GroupObjectId $group.ObjectId

* + - * 1. On the toolbar, click the **Run Selection** button, and wait for the script to complete.
        2. Switch to Microsoft Edge displaying the Azure portal.
        3. From the **adatum** blade, verify that **Mario Ledford** appears in the list of users.
        4. From the **adatum** blade, verify that **Azure team** appears in the list of groups.

**Result**: After completing this exercise, you should have created some pilot users and groups in Azure AD by using the Azure portal and Microsoft Azure Active Directory Module for Windows PowerShell. You will also enable the Azure AD Premium functionality.

 **Exercise 2: Configuring Application SSO**



##### Scenario

Because A. Datum is planning to deploy cloud-based applications, and requires users to use SSO for these applications, you now want to install and configure a test application, and then validate the SSO experience.

The main tasks for this exercise are as follows:

* + - 1. Add directory applications and configure SSO
      2. Test SSO
      3. **Task 1: Add directory applications and configure SSO**
         1. In the **Adatum** directory, add the **Microsoft Account (Windows Live)** application from the gallery:
         2. Configure single sign-on for the application with the **Pasword-based Sign-on** setting.
         3. Assign the application to **Mario Ledford**.
         4. Select the option that allows you to enter the Microsoft account credentials on behalf of the user.
         5. In the **Email Address** box, type the name of your Microsoft account you are using for this lab. In the **Password** box, type the corresponding password, and then click the check mark.
         6. In the **Adatum** directory, add the **Skype** application from the gallery:
         7. Configure single sign-on for the application with the **Pasword-based Sign-on** setting.
         8. Assign the application to **Mario Ledford**
         9. Sign out from the Azure portal.
      4. **Task 2: Test SSO**
         1. Open an Microsoft Edge window and browse to [**https://myapps.microsoft.com**.](https://myapps.microsoft.com/) When prompted, sign in by using specify the full user name (including the @*domain name*.onmicrosoft.com suffix) of **Mario Ledford's account** and the corresponding password **Pa55w.rd**.
         2. On the applications page, click the ellipsis next to **Skype**. Note the option to update the credentials.
         3. On the applications page, click the ellipsis next to **Microsoft Account**. Note that there is no option to update the credentials.
         4. Switch to the Windows PowerShell ISE window and use the Windows PowerShell **Set- Service** cmdlet to set the startup of the Windows Update service to manual.
         5. Switch back to the Microsoft Edge window, click **Skype** and, when prompted, install the My Apps Secure Sign-in Extension Microsoft Store app with the default settings and enable the extension once the installation completes.
         6. Restart Microsoft Edge and browse to [**https://myapps.microsoft.com**.](https://myapps.microsoft.com/) When prompted, sign in as **Mario Ledford**.
         7. From the Application Access Panel, start **Skype**. Note that you are now prompted for credentials, because you did not enter any credentials on behalf of the user when configuring SSO.
         8. Click **Cancel** in the **Skype** dialog box.
         9. Sign out from the Application Access Panel and close Microsoft Edge.
         10. Switch to the Windows PowerShell ISE window and use the Windows PowerShell **Set- Service** cmdlet to disable the Windows Update service.

**Result**: After completing this exercise, you should have installed and configured a test application and validated the SSO experience.

 **Exercise 3: Configuring Multi-Factor Authentication**



##### Scenario

Because A. Datum requires applications to use Multi-Factor Authentication, you now want to configure and test Multi-Factor Authentication for Global Administrators.

The main tasks for this exercise are as follows:

* + - 1. Configure Multi-Factor Authentication
      2. Test Multi-Factor Authentication
      3. **Task 1: Configure Multi-Factor Authentication**
         1. Start Internet Explorer and sign in to the Azure portal by using the Microsoft account that is the Service Administrator of your subscription.
         2. Enable Multi-Factor Authentication for the Adatum Azure AD user account of **Karen Gruber**.
         3. Close Microsoft Edge.
      4. **Task 2: Test Multi-Factor Authentication**
         1. Open Microsoft Edge, browse to [**https://myapps.microsoft.com**,](https://myapps.microsoft.com/) and sign in

as **Karen Gruber**. You will be presented with the message stating ***More information is required. Your organization needs more information to keep your account secure***.

* + - * 1. Click **Next**.
        2. On the **Additional security verification** page, in the first drop-down list, ensure that **Authentication phone** is selected. Enter your phone number and select the option **Call me**.
        3. Answer the call and follow instructions in the recorded message.
        4. Close Microsoft Edge

**Result**: After completing this exercise, you should have configured Multi-Factor Authentication for a Global Admin account.

 **Exercise 4: Configuring SSO from a Windows 10-based computer that is joined to Azure AD**



##### Scenario

A. Datum has an increasing demand to provide its remote and mobile users, who are using Windows 10-based devices, with secure access to the cloud resources. The company wants to join Windows 10 devices to Azure AD and simplify access to cloud resources by enabling SSO. Before they can implement this, you want to test this functionality by joining a WindowsÂ 10-based computer to Azure AD.

The main tasks for this exercise are as follows:

1. Join a Windows 10-based computer to Azure AD
2. Authenticate to Azure from a Windows 10 Azure-joined computer
3. Remove the lab environment
   * + 1. **Task 1: Join a Windows 10-based computer to Azure AD**
          1. Start Microsoft Edge and sign in to the Azure portal by using the user account that is a Global Administrator of the Adatum Azure AD tenant.
          2. Verify that the **Adatum** directory allows all users to join their devices to Azure AD.
          3. On MIA-CL1, click **Settings**, click **Accounts**, and then join MIA-CL1 into Azure AD by using the Adatum Azure AD credentials of Karen Gruber.
          4. In the Azure portal, verify that **MIA-CL1** is shown on the Devices blade of the **Karen Gruber** user account.
          5. Restart **MIA-CL1**.
       2. **Task 2: Authenticate to Azure from a Windows 10 Azure-joined computer**
          1. Sign in to MIA-CL1 by using the Karen Gruber's Adatum Azure AD account and the password you set for this account in exercise 1.
          2. Start Microsoft Edge and browse to the Azure portal.
          3. Verify that you are automatically signed in as Karen Gruber by using SSO.
          4. Sign out from MIA-CL1
       3. **Task 3: Remove the lab environment**
          1. On MIA-CL1, close all open windows without saving any files.
          2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
          3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
          4. If you have multiple Azure subscriptions, select the one you want the script to target.
          5. If prompted, specify the current lab number.
          6. When prompted for confirmation, type **y**.
          7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
          8. In the Azure portal, reset the dashboard to the default state.
          9. Close all open windows.

**Result**: After completing this exercise, you should have joined the MIA-CL1 computer to Azure AD and tested the SSO access to the resources in the cloud.

**Question** What is the major benefit of joining Windows 10-based devices to Azure AD?

**Question** What is the requirement for Delegated Group Management in Azure AD?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 10: Managing an Active Directory infrastructure in a hybrid environment

**Lab: Implementing and managing Azure AD synchronization**



##### Scenario

Adatum Corporation users access on-premises applications by authenticating once, during initial sign-in to their client computers. While evaluating Azure for Adatum, you must verify that Adatum users can continue using their existing credentials to access Azure resources. In addition, you must verify that attribute changes to Active Directory user and group accounts will automatically replicate to Azure AD.

##### Objectives

After completing this lab, you will be able to:

* + - * Configure directory synchronization.
      * Synchronize on-premises Active Directory with Azure Active Directory.

##### Lab Setup

Estimated Time: 60 minutes Virtual machine: **20533E-MIA-CL1** User name: **Student**

Password: **Pa55w.rd**

 **Exercise 1: Configuring directory synchronization**



##### Scenario

Adatum plans to integrate its AD DS with Azure AD. To test this plan, you need to deploy and configure Azure AD Connect to synchronize your test Active Directory environment with a test Azure AD tenant. To eliminate the need to verify a custom DNS domain, you will be using the default DNS name of the test Azure AD domain.

The main tasks for this exercise are as follows:

* + - 1. Sign in to the Azure VM hosting an Active Directory domain controller and create test Active Directory objects.
      2. Create a new Azure AD tenant and a Global Admin account
      3. Install Azure AD Connect with custom settings
      4. **Task 1: Sign in to the Azure VM hosting an Active Directory domain controller and create test Active Directory objects.**
         1. Sign in to MIA-CL1 as **Student** with the password **Pa55w.rd**.
         2. Open the file **F:\Labfiles\Lab10\Starter\Set-20533E1001Lab.ps1** in WordPad and copy all lines to Clipboard.
         3. In the Microsoft Edge window, navigate to the Azure portal and, when prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If necessary, in the Azure portal, switch to the Azure Active Directory tenant associated with the Azure subscription that you chose when running the provisioning script at the beginning of this module.
         5. Initiate a Remote Desktop Protocol (RDP) session to **20533E1001-vm1**, and then sign in as **ADATUM\Student** with the password **Pa55w.rd1234**.
         6. In the Remote Desktop session, start Windows PowerShell ISE as administrator, paste the content of Clipboard into the script pane and run the pasted commands.
         7. From the Windows PowerShell ISE window, run the **Get-ADUser** cmdlet to verify that the list of Active Directory user accounts includes **Beverly Beach** in

the **AccountsToSync** organizational unit and **Darwin Shivers** in the **AccountsNotToSync** organizational unit.

* + - * 1. Close the **Administrator: Windows PowerShell ISE** window.
      1. **Task 2: Create a new Azure AD tenant and a Global Admin account**
         1. Within the Remote Desktop session, from Server Manager, disable IE Enhanced Security Configuration for administrators.
         2. Open Internet Explorer and navigate to the Azure portal. If prompted to set up Internet Explorer 11, ensure that the **Use recommended security, privacy, and compatibility settings** option is selected and click **OK**.
         3. From the Azure portal, create a new Azure AD tenant with the following settings:

Organization name: **AdatumSync**

Initial domain name: a unique, valid name

Country or region: **United States**

1. Refresh the Microsoft Edge window, in the Azure portal, switch to the newly created Azure AD tenant, and create a new Global Admin user with the following settings:
   * Name: **SyncAdmin**
   * User name: [**syncadmin@*domain-name*.onmicrosoft.com**](mailto:syncadmin@domain-name.onmicrosoft.com)where ***domain-name*** is the unique name you assigned to the AdatumSync Azure AD tenant earlier in this task
   * First name: **Sync**
   * Last name: **Admin**
   * Directory role: **Global administrator**
   * Show Password: enabled
2. Take the note of the autogenerated temporary password.
3. Open an Internet Explorer InPrivate Browsing session, sign in to the Azure portal as **SyncAdmin**, and change the password to a new value. Take a note of the new value.
4. Sign out as SyncAdmin and close the InPrivate Microsoft Edge session.
   * + 1. **Task 3: Install Azure AD Connect with custom settings**
          1. From the Remote Desktop session to 20533E1001-vm1, switch to the Internet Explorer displaying the Azure portal. Use the **Download Azure AD Connect** link on the **Azure AD Connect** page of locate the download page of Azure AD Connect MSI file and download it the **Downloads** folder.
          2. Install the Azure AD Connect tool, select custom settings, and then ensure that **Password Hash Synchronization** is selected as the user sign-in method.
          3. When prompted, connect to the AdatumSync Azure AD tenant by using the **SyncAdmin** Global Administrator account.
          4. When prompted, connect to the Adatum AD forest by using

the **ADATUM\Student** Enterprise Admin account with the password **Pa55w.rd1234**.

* + - * 1. When prompted, acknowledge that you intend to continue without matching all UPN suffixes to verified domains.
        2. On the **Domain and OU filtering** page, limit synchronization to the **AccountsToSync** organization unit only.
        3. Accept the default values in the remaining wizard pages, and then start the synchronization process. Close the wizard once the configuration is completed.

**Note:** Installation will take a few minutes.

1. In the Azure portal, on the **AdatumSync** blade, navigate to the **All Users** blade of

the **AdatumSync** Azure AD tenantn and confirm that the list of users includes **Beverly Beach** from the **AccountsToSync** OU but does not include **Darwin Shivers** from the **AccountsNotToSync** OU.

**Result**: After completing this exercise, you should have installed and configured Azure AD Connect, and have performed initial synchronization.

 **Exercise 2: Managing synchonization**



##### Scenario

Adatum wants to test Azure AD synchronization by changing a few attributes of a synchronized user account and then performing manual synchronization.

The main tasks for this exercise are as follows:

* + - 1. Modify attributes of an Active Directory user and Initiate manual synchronization
      2. Remove the lab environment
      3. **Task 1: Modify attributes of an Active Directory user and initiate delta synchronization**
         1. On 20533E1001-vm1, start **Windows PowerShell ISE** as Administrator.
         2. From the Windows PowerShell ISE console, check the value of

the **Title** and **Department** attributes of the user **bbeach** by using the **Get- ADUser** cmdlet.

* + - * 1. From the Windows PowerShell ISE console, use the **Set-ADUser** cmdlet to set the value of the **Title** and **Department** attributes of the

user **bbeach** to **VP** and **Marketing**, respectively.

* + - * 1. From the Windows PowerShell ISE console, import the module **'C:\Program Files\Microsoft AZure AD Sync\Bin\ADSync\ADSync.psd1'** and check the current synchronization settings by running the **Get-ADSyncScheduler** cmdlet.
        2. From the Windows PowerShell ISE console, start delta synchronization by running **Start-ADSyncSyncCycle -PolicyType Delta**.
        3. From the the **Users and groups - All Users** blade in the Azure portal, verify that the changes that you made to the user accounts have been synchronized to Azure AD. If you do not see any changes, wait for a few minutes, and then refresh the page.
        4. Close the 20533E1001-vm1 Remote Desktop session.
      1. **Task 2: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Start **Windows PowerShell** as Administrator.
         10. Use the **Connect-MsolService** cmdlet to authenticate to the AdatumSync Azure Active Directory tenant by using the SyncAdmin credentials.
         11. Use the **Set-MsolDirSyncEnabled** cmdlet to disable directory synchronization to the AdatumSync Azure Active Directory tenant.
         12. Close all open windows.

**Result**: After completing this exercise, you should have changed attributes on a user account, and then forced synchronization.

**Question** How do you configure organizational unit (OU)-level filtering for directory synchronization?

**Question** When do you use Azure AD Connect custom setup?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

#### Module 11: Implementing Azure-based management and automation

**Lab: Implementing Automation**



##### Scenario

Adatum Corporation wishes to minimize administrative overhead as much as possible, especially for tasks that involve management of VMs. For this reason, as part of Adatum’s evaluation of Microsoft Azure, you have been asked to configure an Automation account and use its features to automate the most common VM management tasks.

##### Objectives

After completing this lab, you will be able to:

* Configure Automation accounts.
* Create runbooks.

##### Lab Setup

Estimated Time: 40 minutes

Virtual Machine: **20533E-MIA-CL1**

User Name: **Student**

Password: **Pa55w.rd**

Before starting this lab, ensure that you have performed the "Preparing the Azure environment" demonstration tasks at the beginning of the first lesson in this module and that the **Setup-Azure** script has completed.

 **Exercise 1: Configuring Automation accounts**



##### Scenario

Administrators at Adatum Corporation spend considerable time managing VMs. You want to increase administrator productivity by using Automation to manage VMs.

The main tasks for this exercise are as follows:

* + - 1. Create an Automation account
      2. Create and review Automation assets
      3. **Task 1: Create an Automation account**
         1. Ensure that you are signed in to MIA-CL1 as **Student** with the password **Pa55w.rd**, and that the setup script you ran in the previous demonstration to prepare the environment has completed.
         2. Start Microsoft Edge and sign in to the Azure portal by using the Microsoft account that is the Service Administrator of your Azure subscription. If necessary, in the Azure portal, switch to the Azure Active Directory tenant associated with the Azure subscription that you chose when running the provisioning script at the beginning of this module.
         3. From the **Virtual machines** blade, note that **20533E1101-vm0** and **20533E1101- vm1** virtual machines are currently running.
         4. Create a new Azure Automation account with the following settings:

Name: **AutomationAccount-20533E11**

Subscription: the name of your Azure subscription

Resource group: **20533E1102-LabRG**

Location: the same Azure region that you chose when running **Add- 20533EEnvironment** script at the beginning of this module or, if not available, another region close to it

Create Azure Run As account: **Yes**

* + - * 1. Wait for the Automation account to be provisioned. This should take less than a minute.
      1. **Task 2: Create and review Automation assets**
         1. In the Azure portal, from the **AutomationAccount-20533E11** blade, create the following Azure Automation non-encrypted string variables

Name: **VM0**

Description: **the first VM**

Type: **String**

Value: **20533E1101-vm0**

Encrypted: **No**

Name: **VM1**

Description: **the second VM**

Type: **String**

Value: **20533E1101-vm1**

Encrypted: **No**

Name: **ResourceGroup**

Description: **VM resource group**

Type: **String**

Value: **20533E1101-LabRG**

Encrypted: **No**

* + - * 1. In the same Automation account, create the following Schedule asset:

Name: **EndOfDay**

Description: **End of day**

Starts: tomorrow's date at **6:00:00 PM** with the time zone of the Azure region containing the Automation account

Recurrence: **Recurring**

Recur every: **1 Day**

Set expiration: **No**

* + - * 1. In the list of assets, note two precreated

connections **AzureClassicRunAsConnection** and **AzureRunAsConnection**. They were created automatically during provisioning of the Automation account since you selected the option to create the Azure Run As account.

**Result**: After completing this exercise, you should have configured a new Azure Automation account, created Automation variable assets and Automation schedule asset, and reviewed the precreated Azure Automation connection assets

 **Exercise 2: Creating and executing runbooks**



##### Scenario

As part of your tests of the new Automation features, you will stop an Azure VM by using an Automation runbook.

The main tasks for this exercise are as follows:

* + - 1. Import a runbook
      2. Publish and execute a runbook
      3. Remove the lab environment
      4. **Task 1: Import a runbook**
         1. From the Azure portal, import the PowerShell workflow

script **F:\Labfiles\Lab11\Starter\Stop-20533E1101VMs.ps1** into your Automation account.

* + - * 1. Review the content of the runbook.
      1. **Task 2: Publish and execute a runbook**
         1. Publish the **Stop-AzureVMs-Workflow** runbook.
         2. Start the newly published runbook.
         3. View the progress of the runbook execution. Wait until the job completes.
         4. From the Azure portal, verify that the of **20533E1101-vm0** and **20533E1101- vm1** virtual machines have been stopped.
      2. **Task 3: Remove the lab environment**
         1. On MIA-CL1, close all open windows without saving any files.
         2. Start **Windows PowerShell** as Administrator and, from the **Administrator: Windows PowerShell** window, run **Remove-20533EEnvironment**.
         3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         4. If you have multiple Azure subscriptions, select the one you want the script to target.
         5. If prompted, specify the current lab number.
         6. When prompted for confirmation, type **y**.
         7. Start Microsoft Edge, browse to the Azure portal, and sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
         8. In the Azure portal, reset the dashboard to the default state.
         9. Close all open windows.

**Result**: After completing this exercise, you should have imported, published, and executed a PowerShell workflow-based runbook that deploys two virtual machines in parallel.

**Question** What mechanism did you use to authenticate when accessing the Azure subscription when running the Azure Automation runbook in the lab?

**Question** What should you consider when testing the execution of an Automation runbook?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License,](https://creativecommons.org/licenses/by/3.0/legalcode) additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.

### Assessment Rubrics for EE432: Computer Networks Lab 15

Student Name: Roll Number:

**Method:**

Lab report evaluation and instructor observation during lab sessions.

**Outcomes Assessed:**

1. Ability to condut experiments as well as to analyze and interpret data
2. Ability to adhere to safety and disciplinary rules
3. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

| **Performance** | **Exceeds expectation (5-4)** | **Meets expectation (3-2)** | **Does not meet expectation (1)** | **Marks** |
| --- | --- | --- | --- | --- |
| **Realization of experiment (a)** | Downloads and installs required software and sets up the system according to the experiment requirements | Needs guidance to set up the system according to the experiment requirements | Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools |  |
| **Conducting experiment (a, c)** | Carries out each procedural step in a satisfactory manner and studies outputs of the software application  rigorously | Needs assistance or guidance to proceed through experiment steps, studies outputs with minor  errors in interpretation | Unable to carry out procedural steps and make any useful observations of outputs |  |
| **Laboratory safety and**  **disciplinary rules (b)** | Observes lab safety rules; adheres to the lab disciplinary guidelines aptly | Observes safety rules and disciplinary guidelines with minor deviations | Disregards lab safety and disciplinary rules |  |
| **Data collection (c)** | Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered  in the lab manual according to the specified instructions | Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines | Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly |  |
| **Data analysis (a, c)** | Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy  in data from theory with sound explanation | Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any  discrepancy in data from theory | Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data |  |