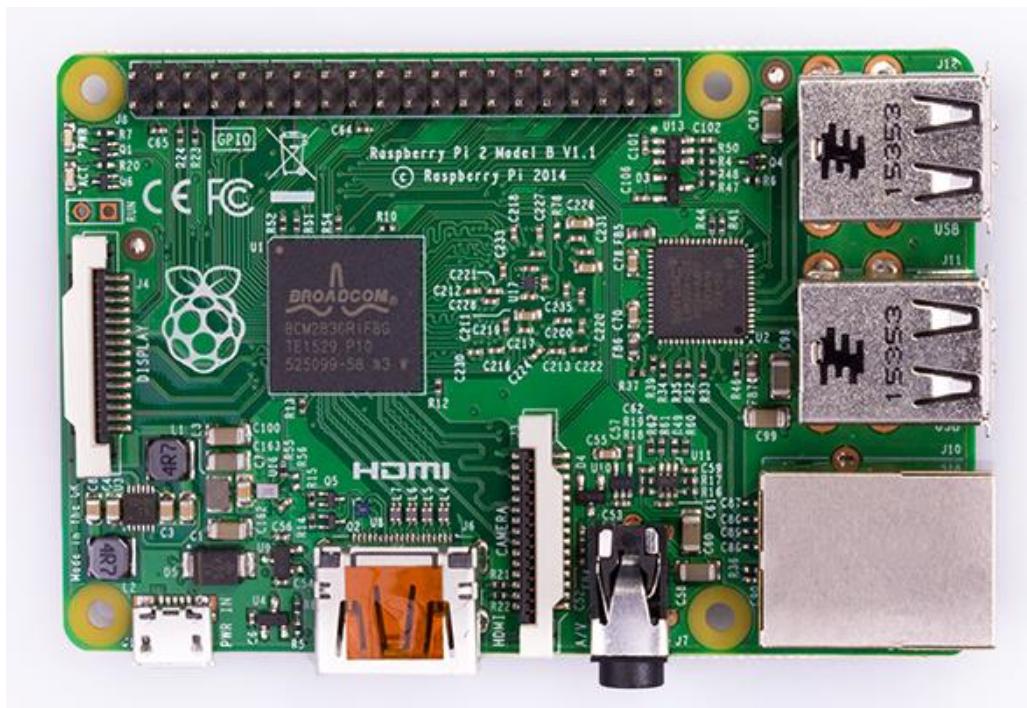


Assignment 1

Title: Study of Embedded Linux / Microkernel

a. Study of Raspberry Pi Model B



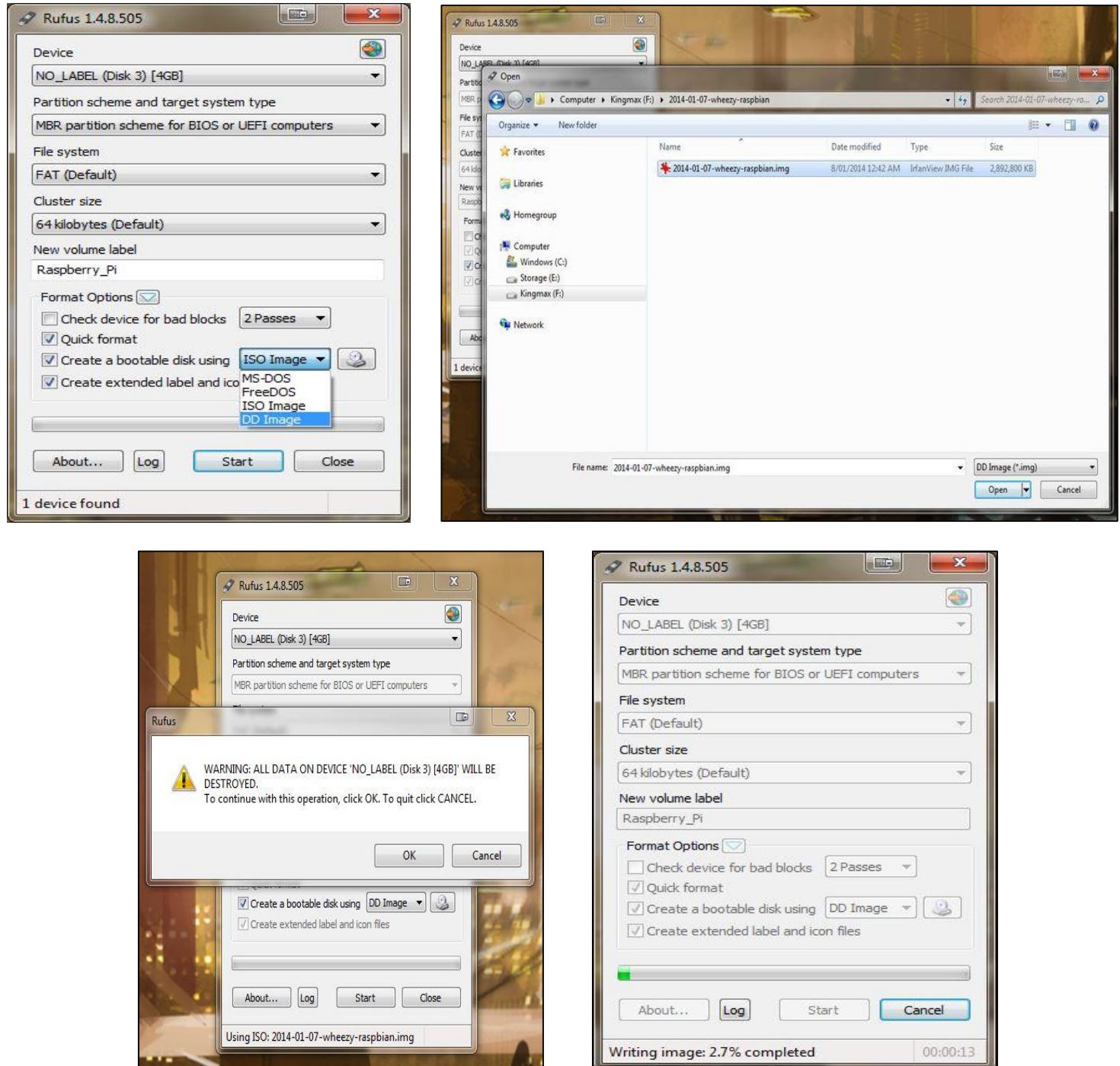
The Raspberry Pi 2 Model B is the second generation Raspberry Pi. It replaced the original Raspberry Pi 1 Model B+ in February 2015. Compared to the Raspberry Pi 1 it has:

- A 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

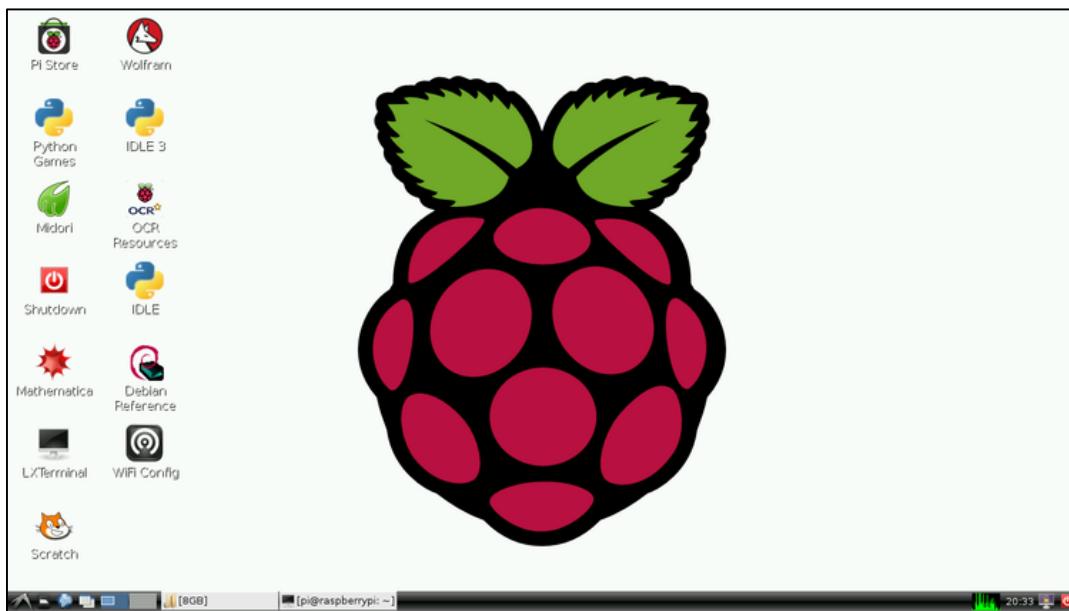
Because it has an ARMv7 processor, it can run the full range of ARM GNU/Linux distributions, including Snappy Ubuntu Core, as well as Microsoft Windows 10. The Raspberry Pi 2 has an identical form factor to the previous (Pi 1) Model B+ and has complete compatibility with Raspberry Pi 1.

b. Downloading existing image and burn on SD card

- Raspbian OS Download URL:
<https://www.raspberrypi.org/downloads/raspbian/>
- Download Rufus Tool to burn SD card with OS image file
URL: <http://rufus.akeo.ie/>

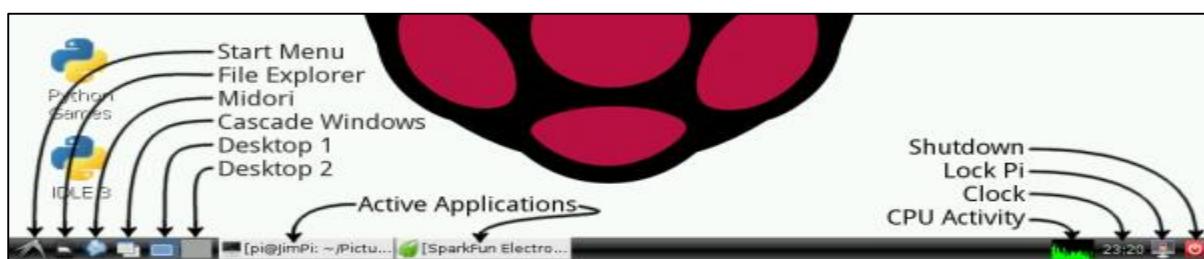


c. Setup and demonstrate as thin client



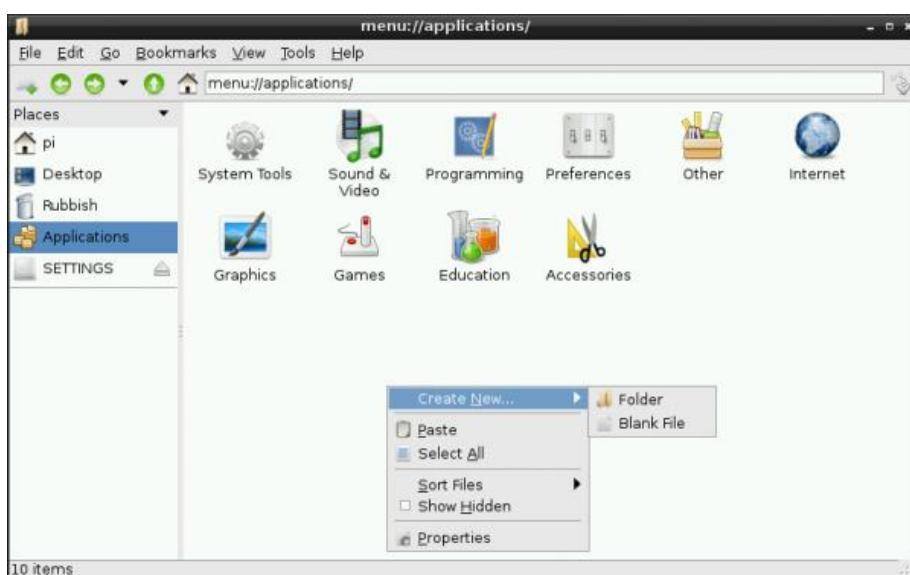
d. Demonstrate its working:

Raspbian comes with a variety of useful software tools. The UI should feel generally similar to Windows, OS X, and other Linux OS'es you may have used.



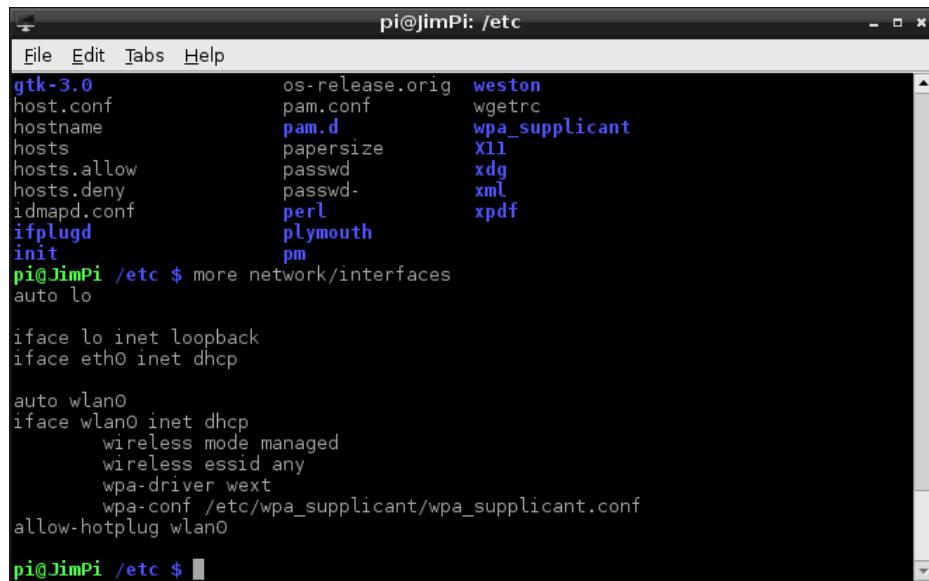
- **File Manager**

If you've done any sort of file manipulation, creation, or deletion, you've probably encountered a file browser before. Open the File Manager by clicking the second icon from the left on the bottom taskbar. *Using the file manager to browse around the Applications folder*. You can use the File Manager to look around at what files, applications, and tools are included with Raspbian. Or you can be a little more Linux-y by using the terminal.



- **LXTerminal**

LXTerminal is Raspbian's default terminal program. If thinking of "Linux" conjures images of hackers typing furiously into a text-filled black-and-white terminal, LXTerminal is the actualization of that.



```
pi@JimPi: /etc
File Edit Tabs Help
gtk-3.0          os-release.orig  weston
host.conf        pam.conf       wgetrc
hostname         pam.d          wpa_supplicant
hosts            papersize      X11
hosts.allow      passwd         xdg
hosts.deny       passwd-       xml
idmapd.conf     perl          xpdf
ifplugd          Plymouth
init             pm
pi@JimPi /etc $ more network/interfaces
auto lo

iface lo inet loopback
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet dhcp
    wireless mode managed
    wireless essid any
    wpa-driver wext
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
allow-hotplug wlan0

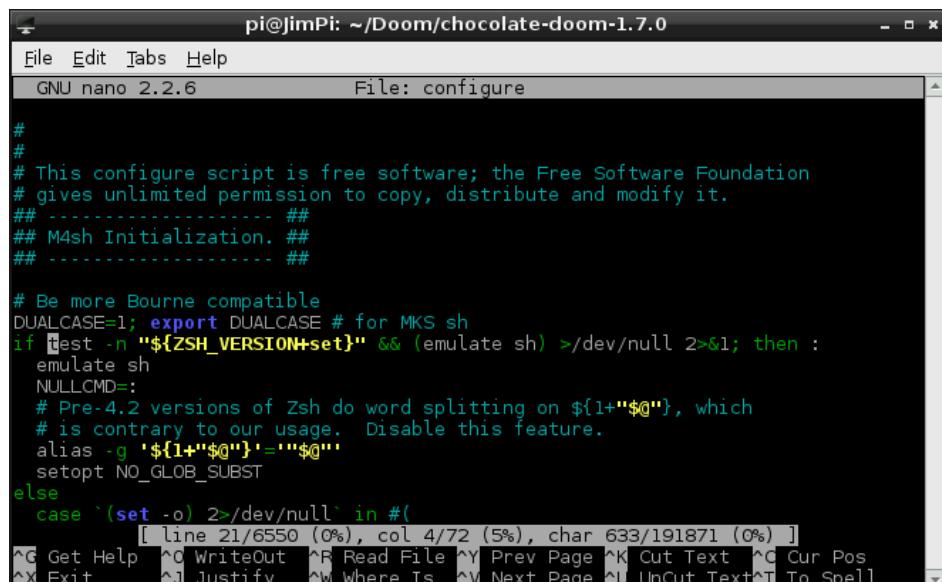
pi@JimPi /etc $
```

If you've never used Linux, or are unfamiliar with the command line interface (CLI), there are plenty of resources to help get you started. Begin with navigating around using cd, list some directory contents with ls, read some command manuals with man, and you'll be a command line ninja in no time.

- **Text Editors: Nano & Leafpad**

Both of these text editors can be used to modify text files, which is necessary to **configure** many applications. The difference between the two is the GUI. Leafpad is a few-frills, graphical text editor like Notepad (Windows) orTextEdit (OS X). Leafpad can be found from the "Accessories" menu under the Start menu.

Nano is Raspbian's terminal-based text editor. It's fast, lightweight, and more technical than Leafpad. Once you get into the habit of using Nano, you'll never go back to Leafpad, because you can edit stuff so much faster. *Editing a makefile config file with Nano.*



```
pi@JimPi: ~/Doom/chocolate-doom-1.7.0
File Edit Tabs Help
GNU nano 2.2.6           File: configure

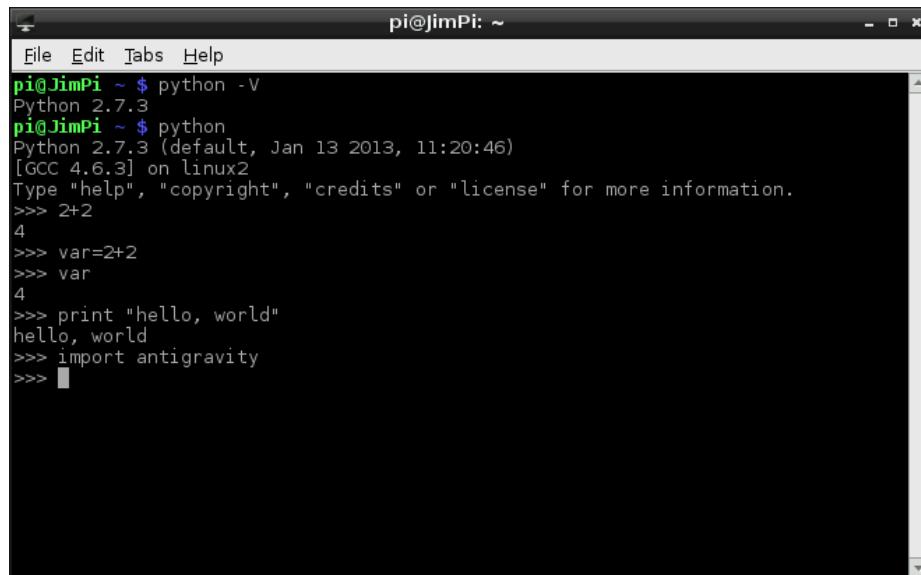
#
#
# This configure script is free software; the Free Software Foundation
# gives unlimited permission to copy, distribute and modify it.
## -----
## M4sh Initialization. ##
## ......

# Be more Bourne compatible
DUALCASE=1; export DUALCASE # for MKS sh
if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&l; then :
    emulate sh
    NULLCMD=
    # Pre-4.2 versions of Zsh do word splitting on ${1+"$@"}, which
    # is contrary to our usage. Disable this feature.
    alias -g '${1+"$@"}'=""$@"
    setopt NO_GLOB_SUBST
else
    case `(`set -o` 2>/dev/null` in
        [l]ine [2]1/6550 ([%], col [4]72 ([%]), char [6]33/191871 ([%]) ]
        ^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
        ^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
    esac
fi
```

Nano is perfect for editing short configuration files, especially those you need superuser privileges for. To open a file for editing with nano, use the terminal command nano filename.abc. Or, to open it with root privileges, use sudo nano filename.abc. If you *really* want to take your terminal text editing a step further, Raspbian does include Vi

- **Python**

Raspbian also includes Python (both version 2.7.3 and 3.2.3), a scripting language of which we're quite enamoured with. Python is an amazingly powerful programming language, thanks to all of the amazing libraries that have already been written with it and its cross-platformability.



```
pi@JimPi ~ $ python -V
Python 2.7.3
pi@JimPi ~ $ python
Python 2.7.3 (default, Jan 13 2013, 11:20:46)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+2
4
>>> var=2+2
>>> var
4
>>> print "hello, world"
hello, world
>>> import antigravity
>>> 
```

Python is useful for simple hardware control (toggling I/O pins, interfacing with I²C sensors, etc.), and it can even be used to create fully-functional projects.

Conclusion:

Studied Raspberry Pi Model B and setup Raspbian image and burn on SD card. Also demonstrated it as thin client and its various applications.

References:

<https://www.raspberrypi.org/downloads/raspbian/>

<http://rufus.akeo.ie/>

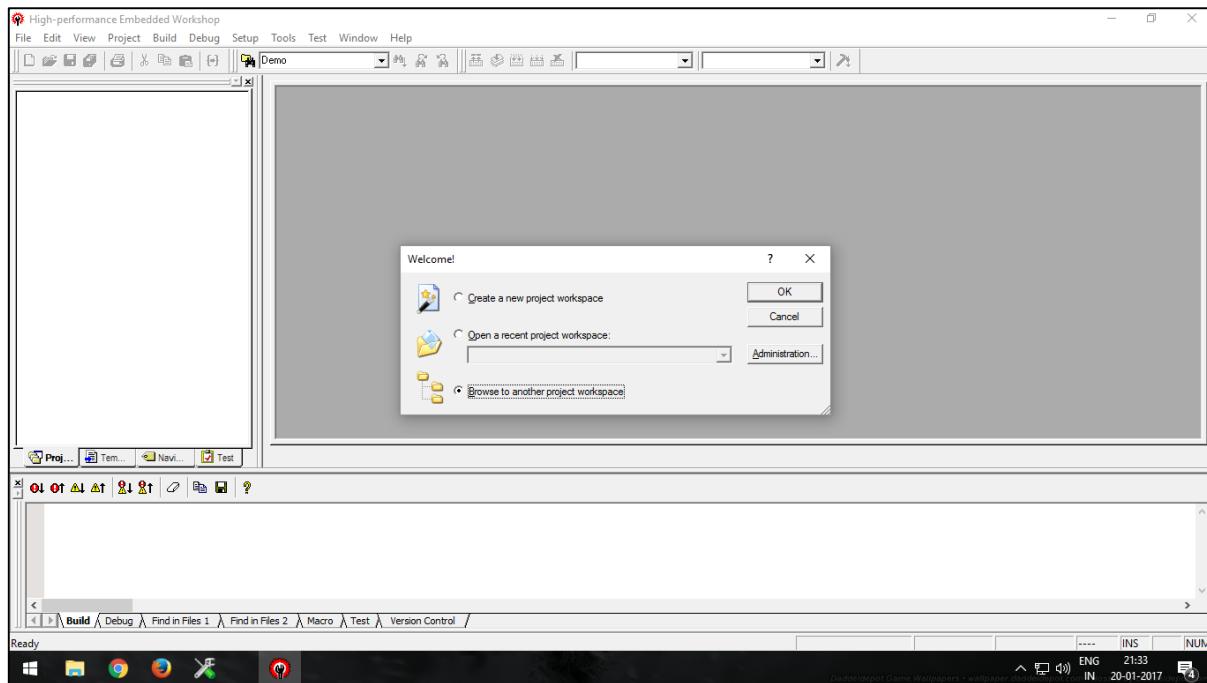
Assignment 2

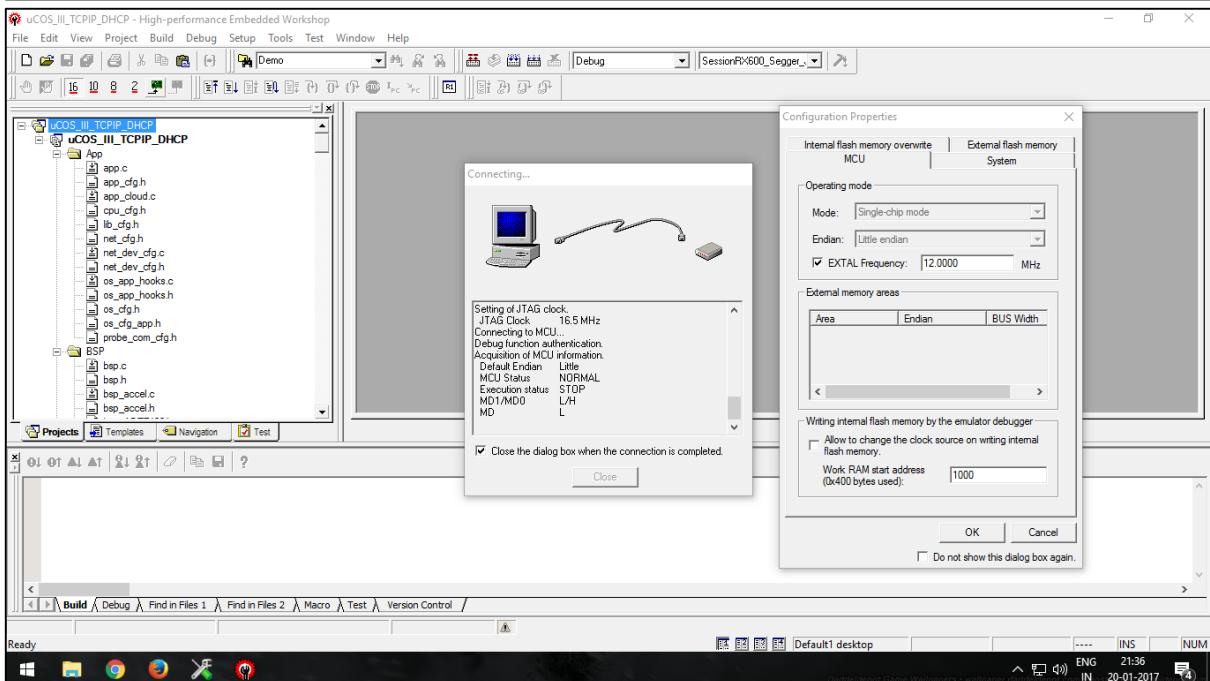
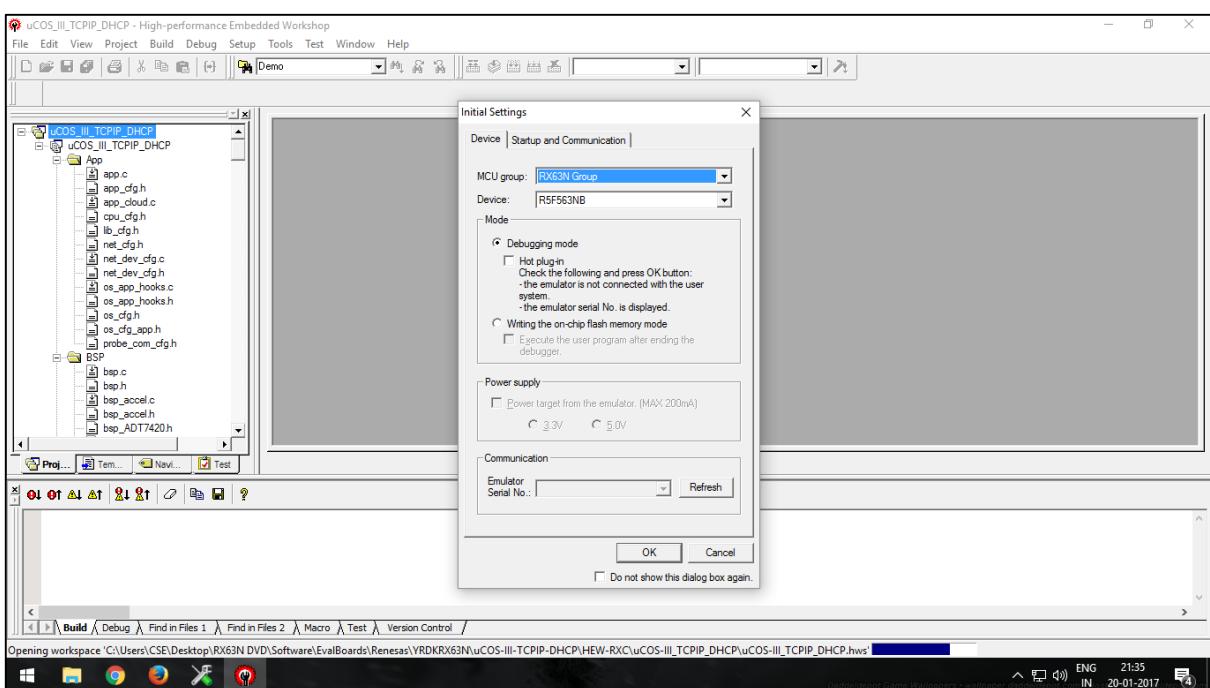
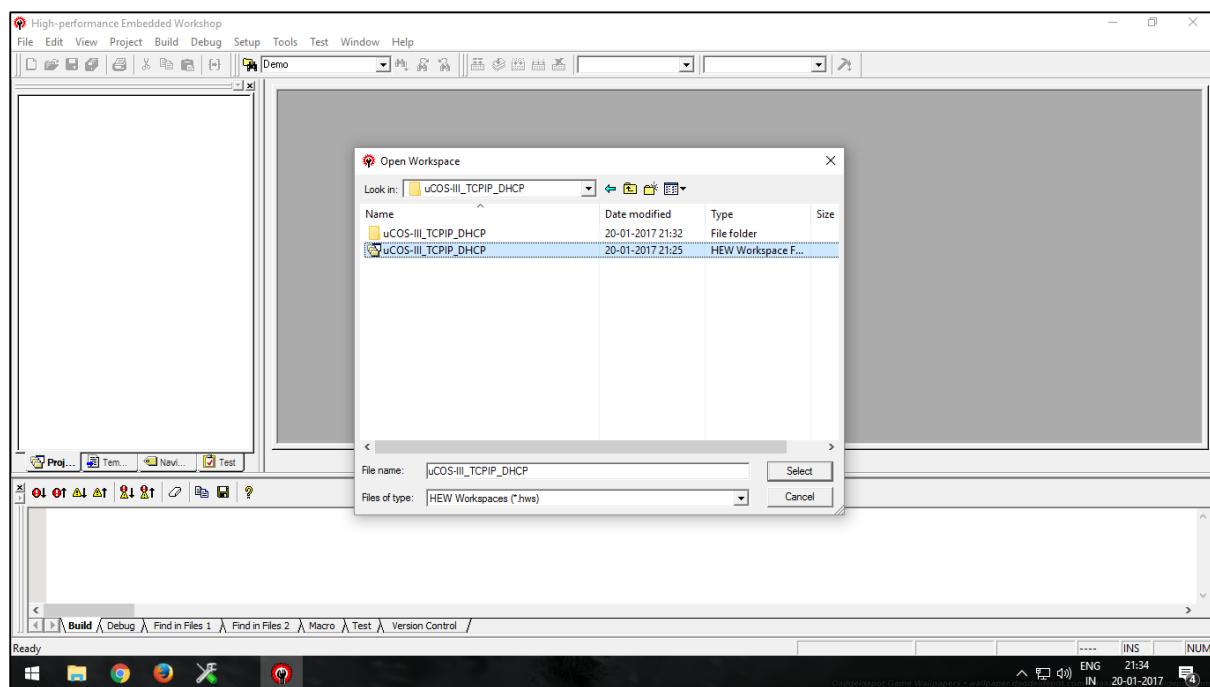
Title: Study of Renesas RX63N Demonstration Kit (YRDKRX63N) and Micro-C OS-II

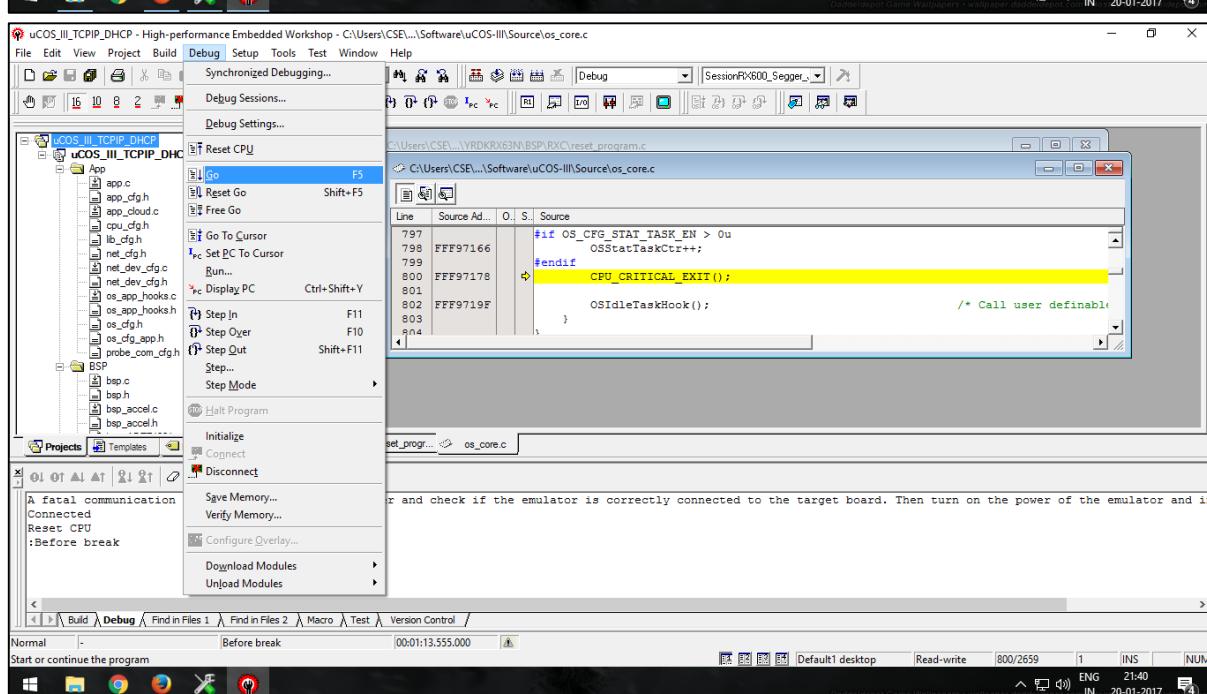
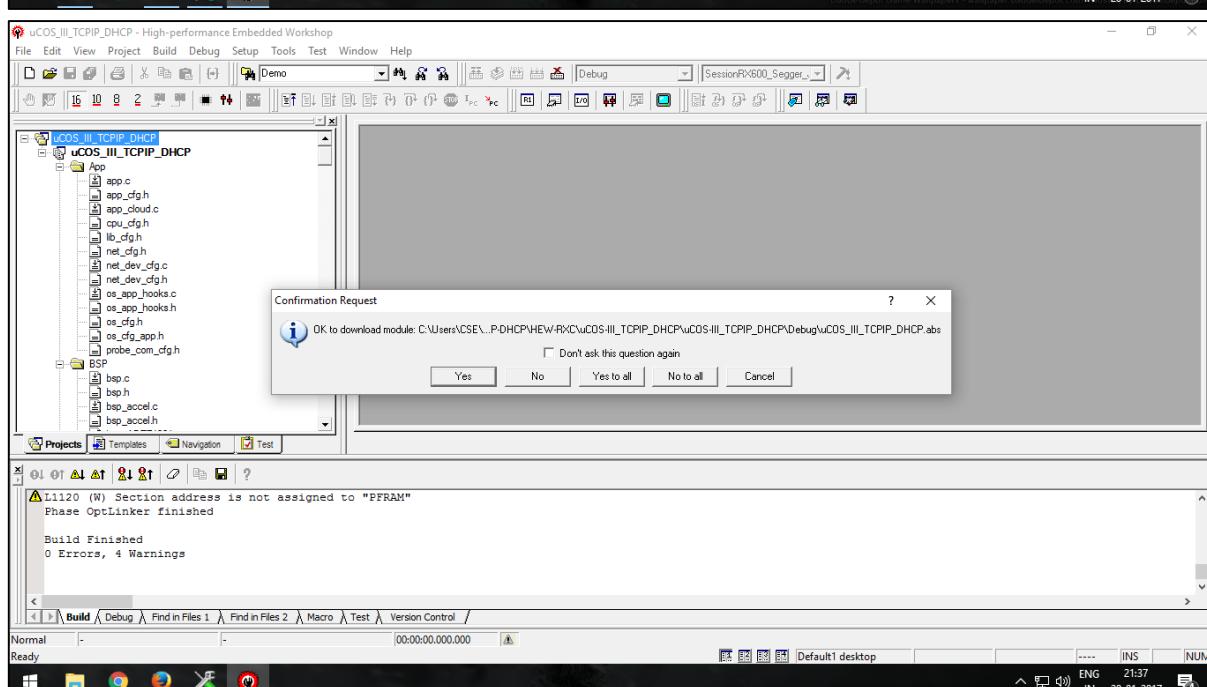
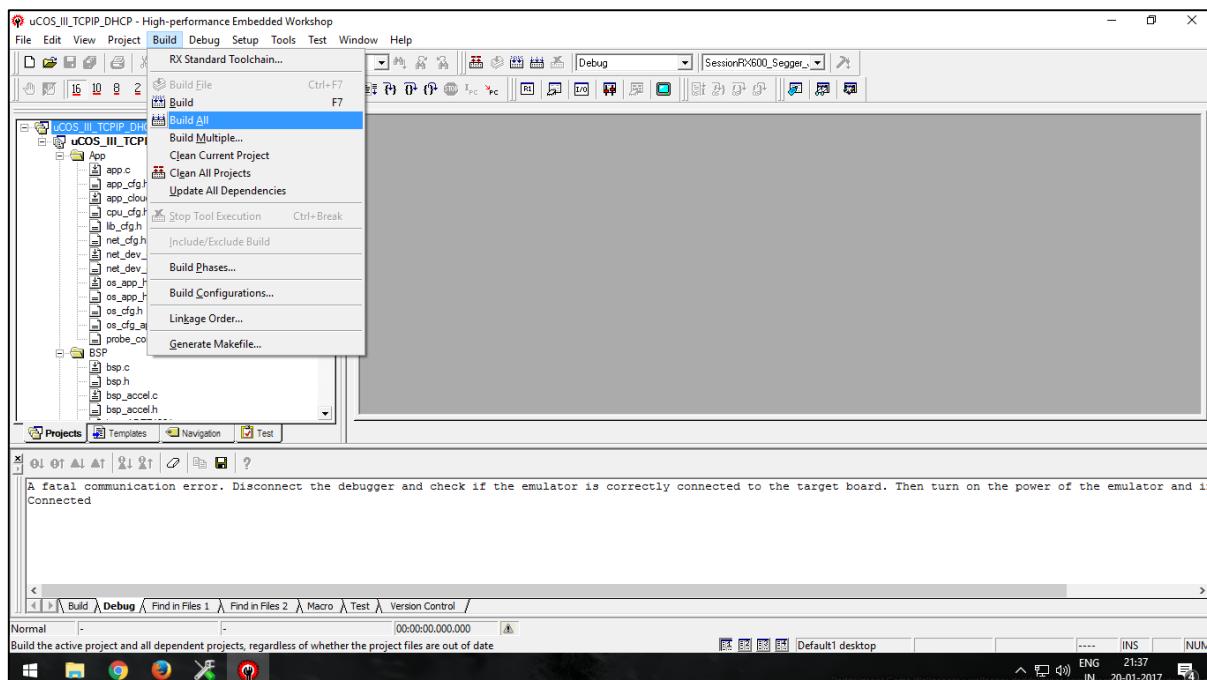
a. Testing the Renesas RX63N Demonstration Kit (YRDKRX63N)

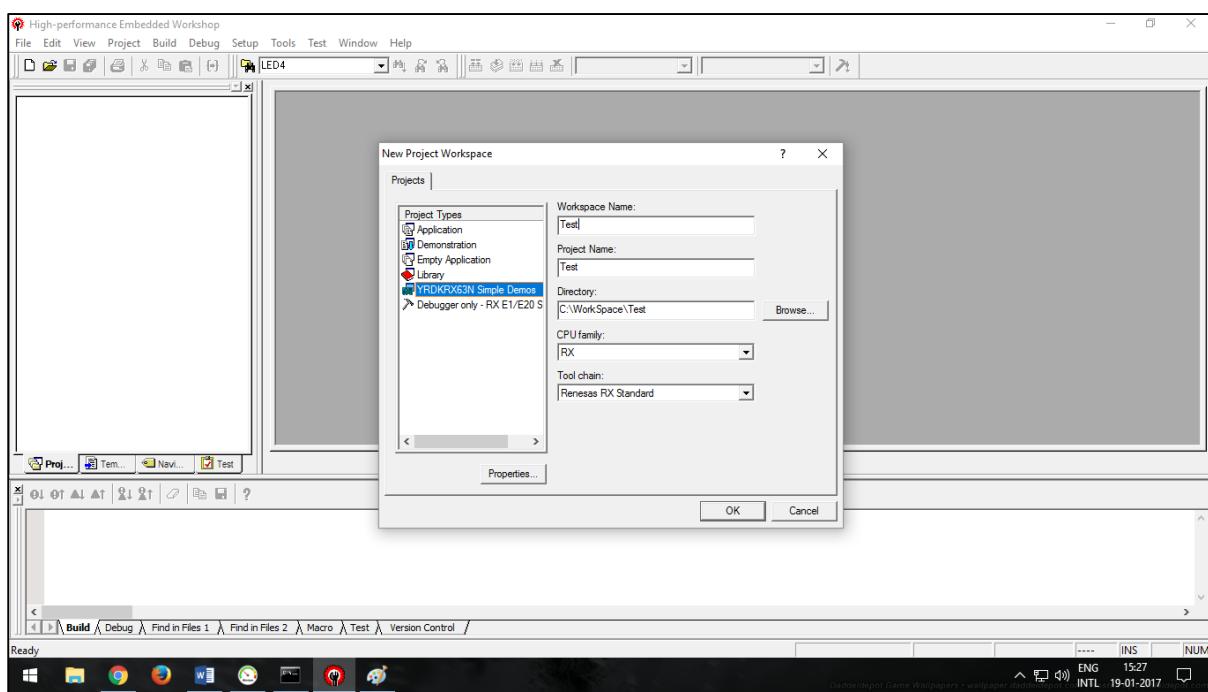
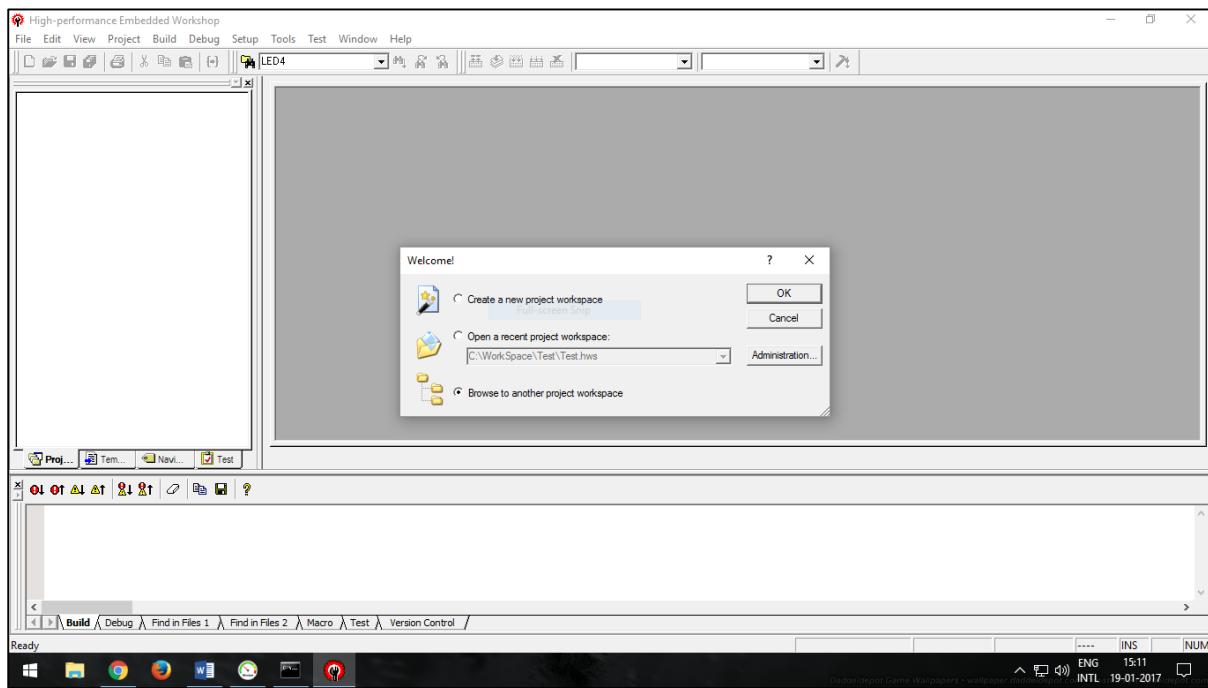
- Install High-Performance Embedded Workshop (HEW) and JLink using setup files provided with YRDKRX63N in DVD.
- On the RDK, ensure the MCU Mode DIP switch, SW5, is set to “DEBUG” (OFF, OFF, OFF, ON) as shown on the right.
- Connect one end of the included USB cable to your computer and the other end to J-Link connector of the RDK.
- The J-Link connector is located above the display between the DB9 connector for the serial port and the power jack.
- This is the debug interface, but for now we are only using it to power the board. Ignore any “Driver not found” errors.
- Press the “RESET” button, SW4.
- You should see the RDK Welcome Screen and version number on the LCD Screen.
- Tilt the board and observe changes to the “Roll” and “Pitch” variables at the bottom of the screen. Turn the
- Potentiometer VR1 and observe changes to the “Pot” variable at the bottom of the screen.

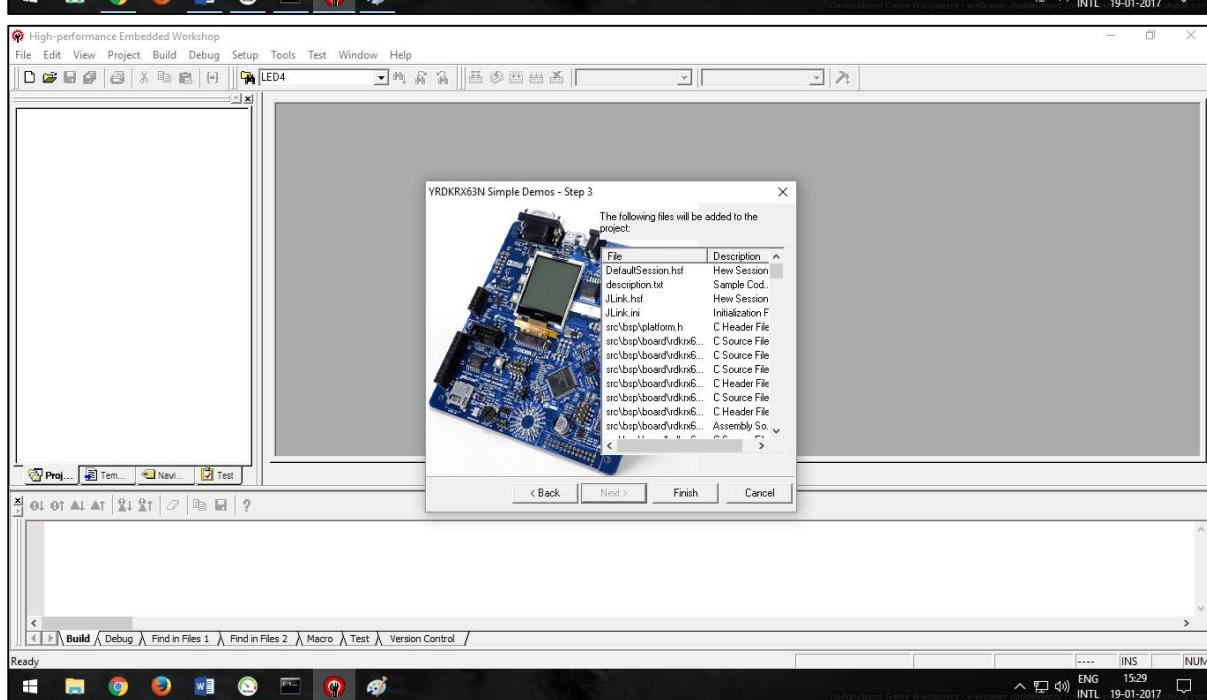
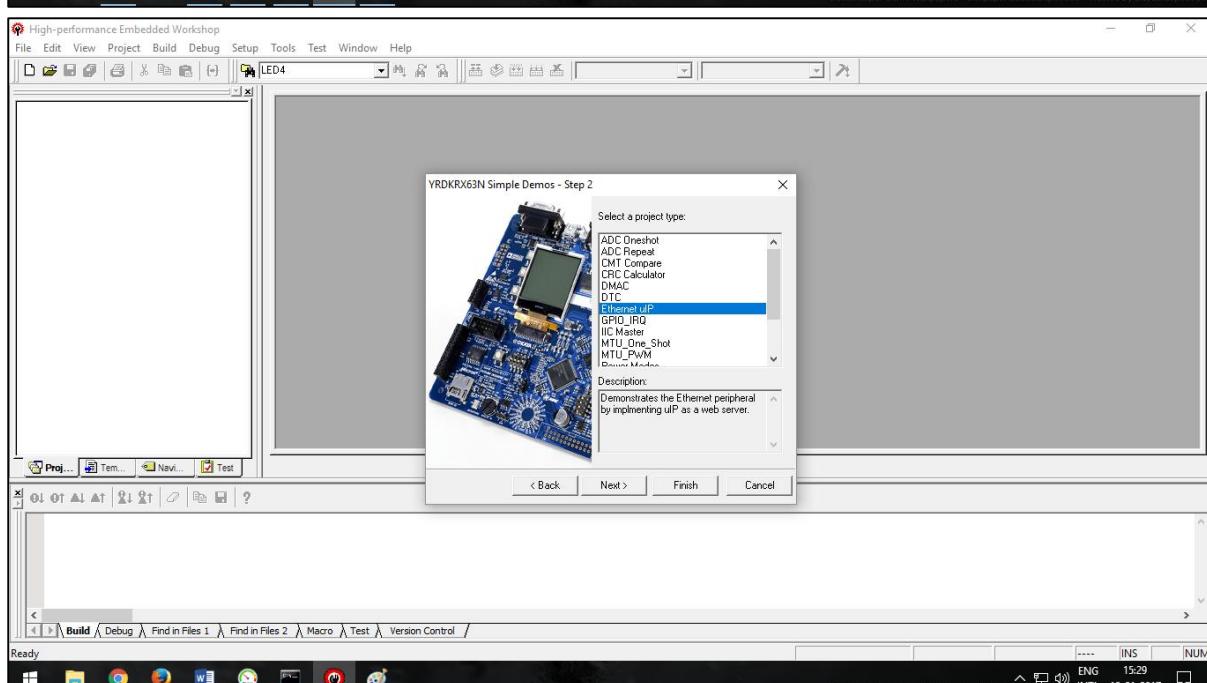
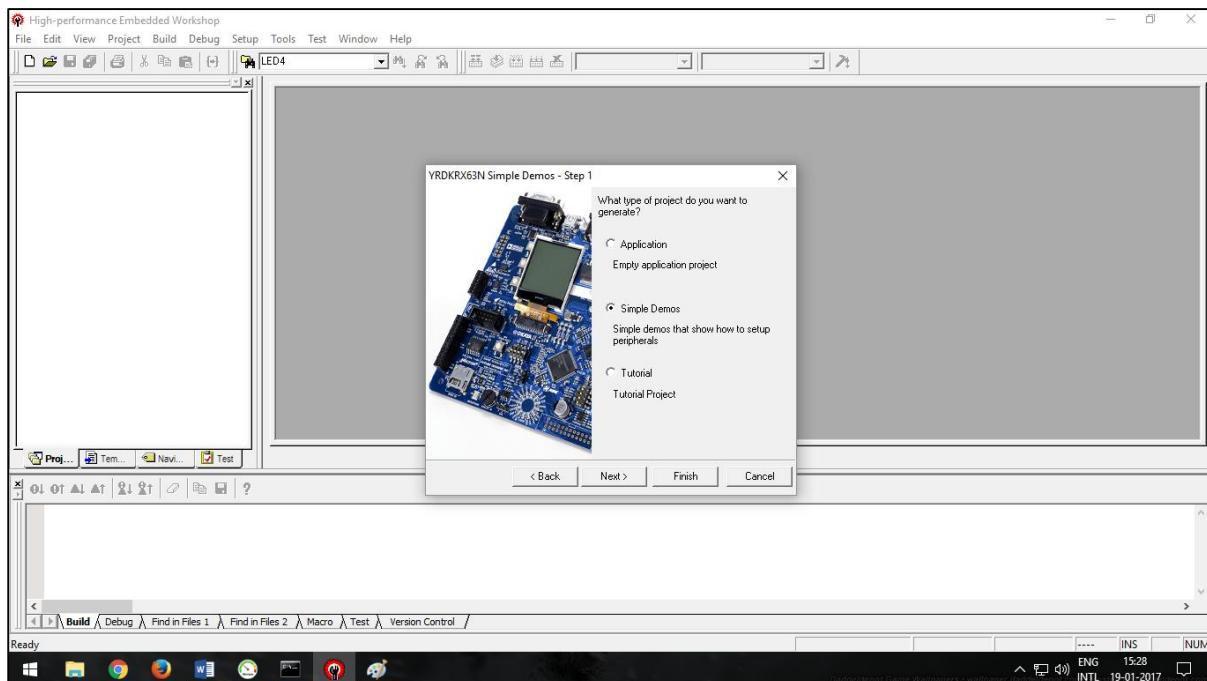
b. Download and burn Micro-C OS kernel on Renesas Kit

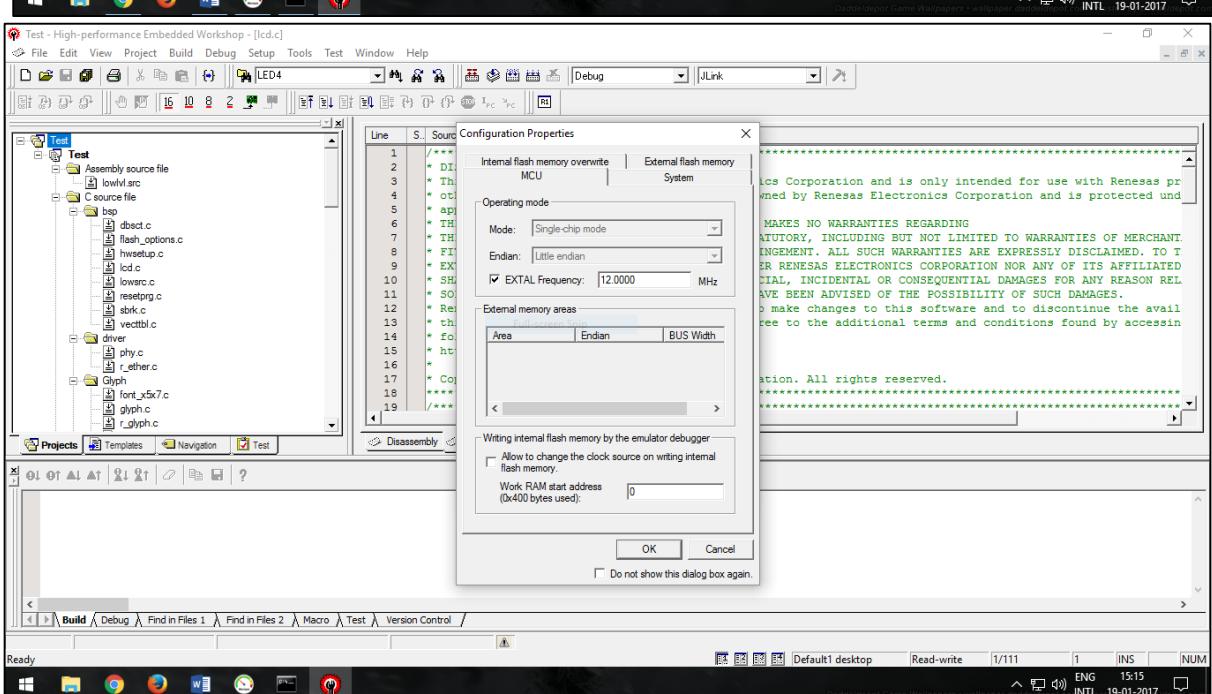
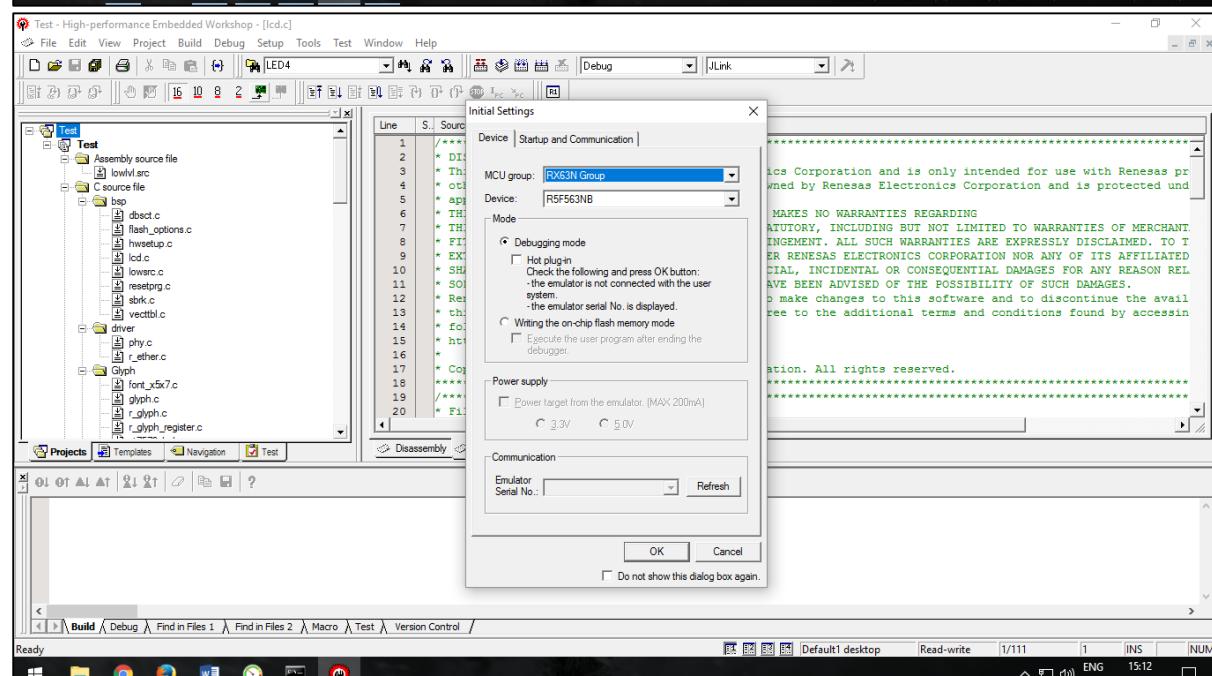
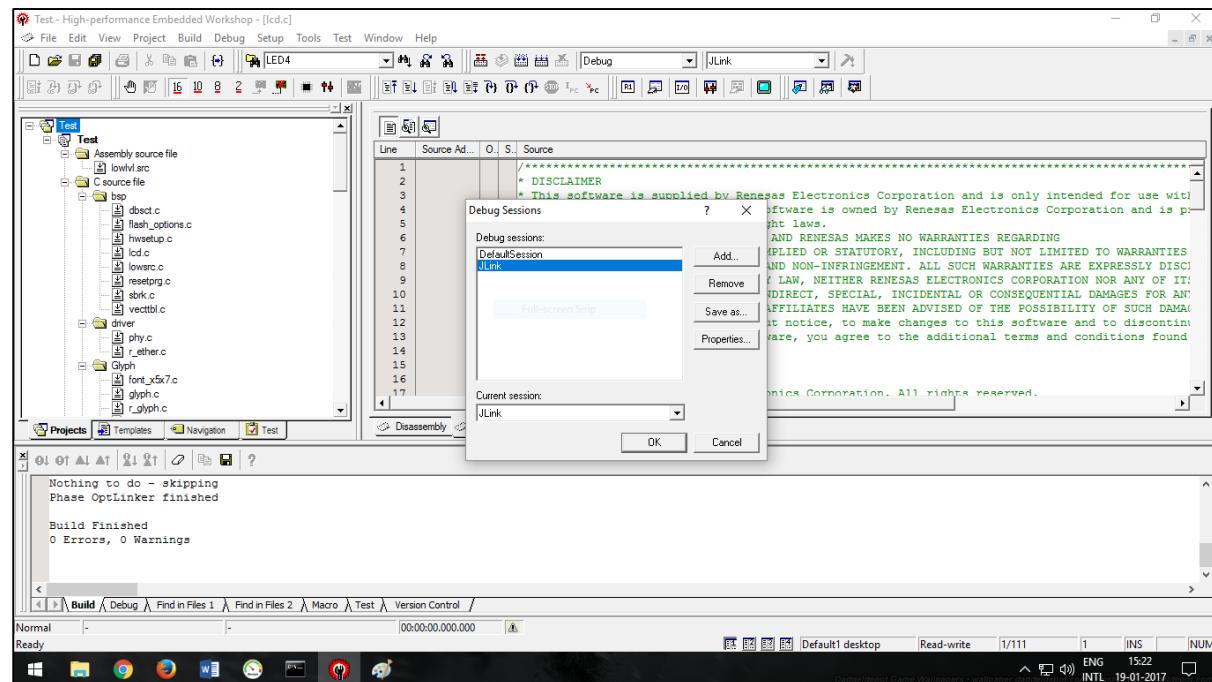


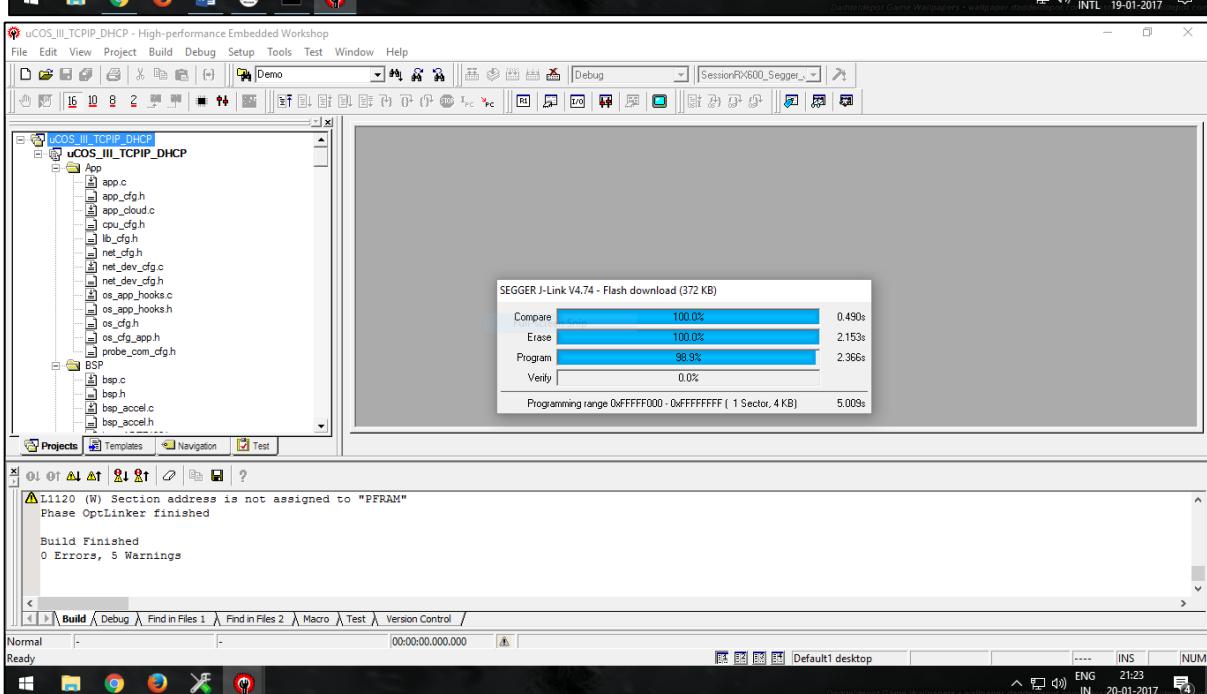
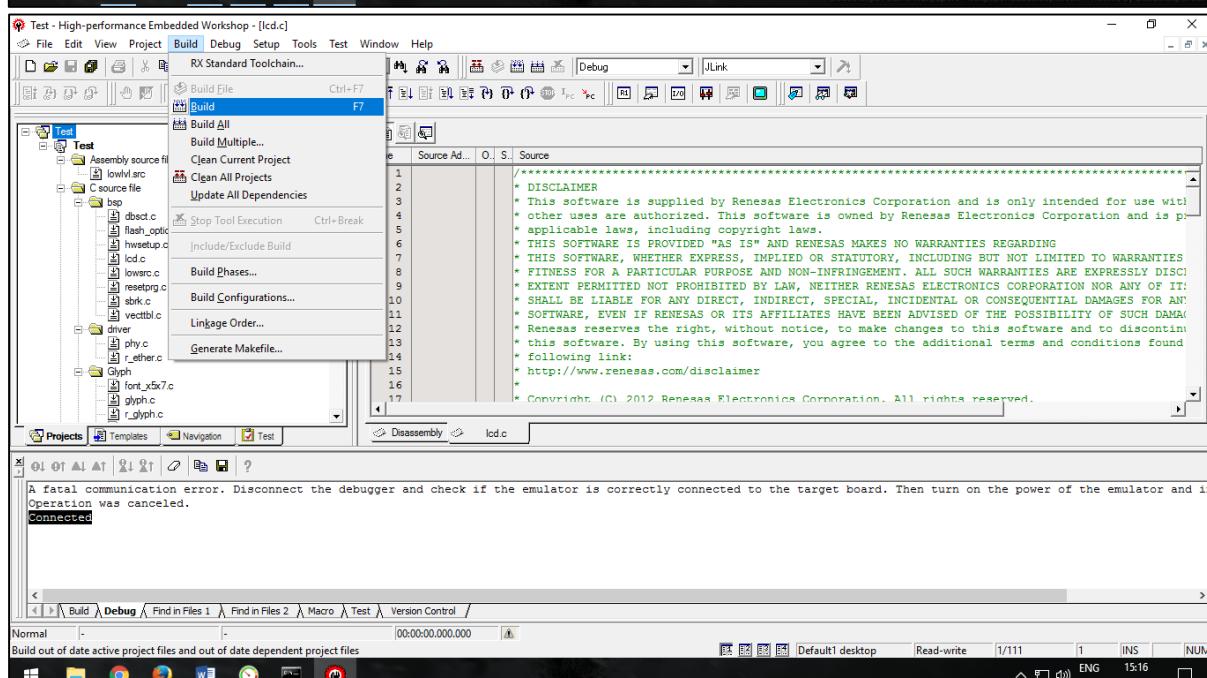
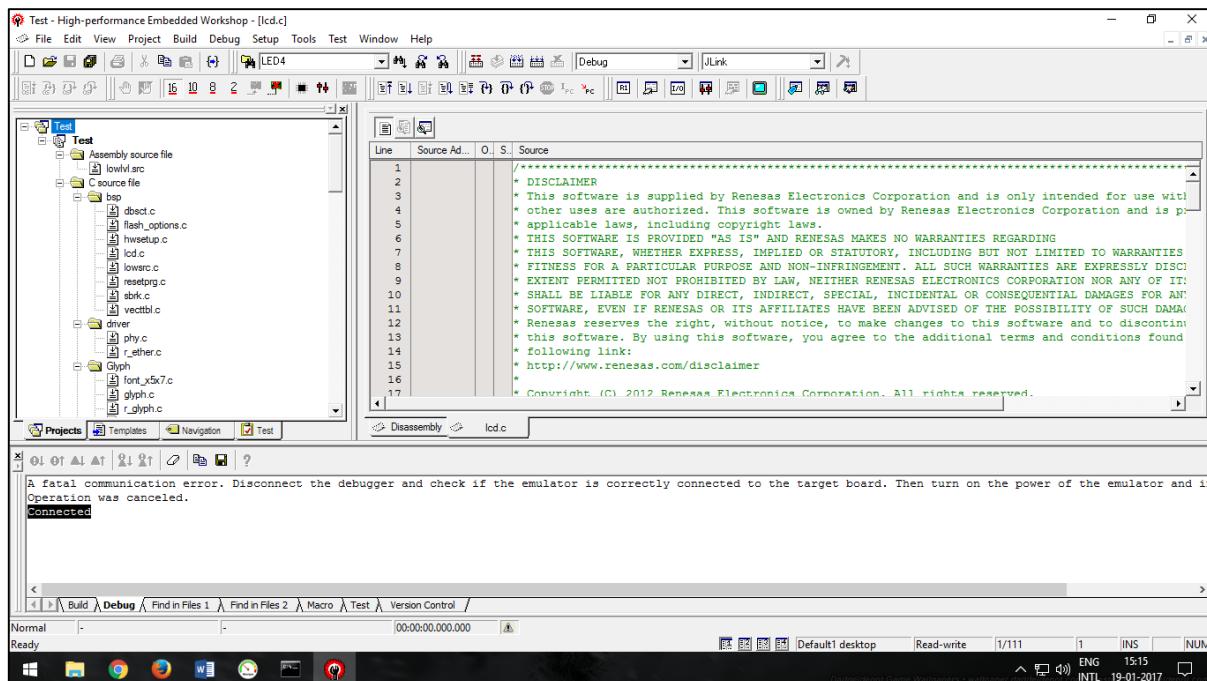




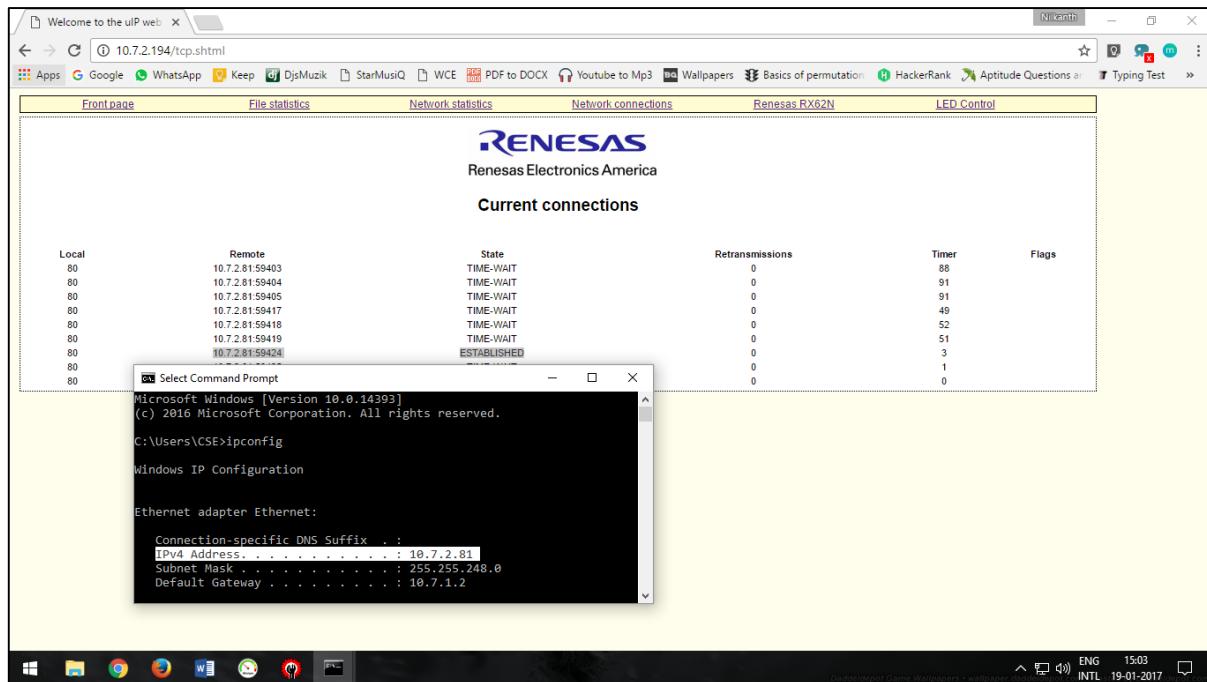
c. Run the demo programs on kit







Connect LAN cable to board. Note down IP address from LCD screen to enter into browser.



According to signals sent via web interface LEDs will turn ON or OFF on board.

Conclusion:

Studied Renesas RX63N Demonstration Kit (YRDKRX63N) and setup Micro-C OS on board. Also, demonstrated sample demo program on kit.

References:

RX63N Demonstration Kit (YRDKRX63N) User Manual and DVD

Assignment 3

Title: Study of Real Time OS and RTLinux Kernel / source code.

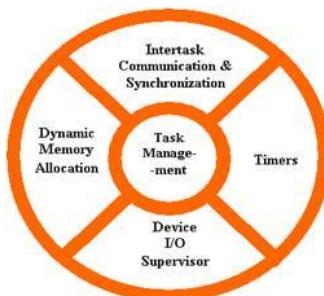
Theory:

A real-time operating system (RTOS; commonly pronounced as "are-toss") is a multitasking operating system designed for real-time applications. Such applications include embedded systems, industrial robots, scientific research equipment and others. An RTOS simplifies the creation of a real-time applications, but does not guarantee the final result will be real-time; this requires good development of the software. Real-time operating systems use specialized scheduling algorithms in order to provide the real-time applications. An RTOS can respond more quickly and/or predictably to an event than other operating systems. The main features of an RTOS are minimal interrupt latency and a minimal thread switching latency.

Learning the difference between real-time and standard operating systems is as easy as imagining yourself in a computer game. Each of the actions you take in the game is like a program running in that environment. A game that has a real-time operating system for its environment can feel like an extension of your body because you can count on a specific "lag time:" the time between your request for action and the computer's noticeable execution of your request. A standard operating system, however, may feel disjointed because the lag time is unreliable. To achieve time reliability, real-time programs and their operating system environment must prioritize deadline actualization before anything else. In the gaming example, this might result in dropped frames or lower visual quality when reaction time and visual effects conflict.

The basic two designs for RTOS are:

- Event-driven (priority scheduling) designs: switch tasks only when an event of higher priority needs service, called pre-emptive priority.
- Time-sharing designs: switch tasks on a clock interrupt, and on events, called round robin.



Examples: These are the best known, most widely used real-time operating systems. See List of real-time operating systems for a full list. Also, see List of operating systems for all types of operating systems.

- Windows CE
- QNX
- RTLinux
- VxWorks
- TRON

RTLinux Kernel / source code

RT Linux is a hard real time RTOS microkernel that runs the entire Linux operating system as fully pre-emptive process. It is one of the hard real-time variants of Linux, among several, that makes it possible to control robots, data acquisition systems, manufacturing plants, and other time-sensitive instruments and machines.

It was developed by Victor Yodaiken, Michael Barabanov, Cort Dougan and others at the New Mexico Institute of Mining and Technology and then as a commercial product at FSMLabs. Wind River Systems acquired FSMLabs embedded technology in February 2007 and made a version available as Wind River Real-Time Core for Wind River Linux. As of August 2011, Wind River has discontinued the Wind River Real-Time Core product line, effectively ending commercial support for the RTLinux product.

The key RTLinux design objective was to add hard real-time capabilities to a commodity operating system to facilitate the development of complex control programs with both capabilities. For example, one might want to develop a real-time motor controller that used a commodity database and exported a web operator interface. Instead of attempting to build a single operating system that could support real-time and non-real-time capabilities, RTLinux was designed to share a computing device between a real-time and non-real-time operating system so that (1) the real-time operating system could never be blocked from execution by the non-real-time operating system and (2) components running in the two different environments could easily share data.

As the name implies RTLinux was the first computer designed to use Linux as the non-real-time system but it eventually evolved so that the RTCore real-time kernel could run with either Linux or BSD UNIX. Multi-Environment Real-Time (MERT) was the first example of a real-time operating system coexisting with a UNIX system. MERT relied on traditional virtualization techniques: the real-time kernel was the host operating system (or hypervisor) and Bell Systems UNIX was the guest. RTLinux was an attempt to update the MERT concept to the PC era and commodity hardware. It was also an attempt to also overcome the performance limits of MERT, particularly the overhead introduced by virtualization.

The technique used was to only virtualize the guest interrupt control. This method allowed the real-time kernel to convert the guest operating system into a system that was completely pre-emptible but that could still directly control, for example, storage devices. In particular, standard drivers for the guest worked without source modification although they needed to be recompiled to use the virtualization "hooks". See also Para virtualization. The UNIX "pipe" was adapted to permit real-time and non-real-time programs to communicate although other methods such as shared memory were also added.

From the programmer's point of view, RTLinux originally looked like a small threaded environment for real-time tasks plus the standard Linux environment for everything else. The real-time operating system was implemented as a loadable kernel module which began by virtualizing guest interrupt control and then started a real-time scheduler.

Tasks were assigned static priorities and scheduling was originally purely priority driven. The guest operating system was incorporated as the lowest priority task and essentially acted as the idle task for the real-time system. Real-time tasks ran in kernel mode. Later development of RTLinux adopted the POSIX threads application programming interface (API) and then permitted creation of threads in user mode with real-time threads running inside guest processes.

In multiprocessor environments threads were locked to processor cores and it was possible to prevent the guest thread from running on designated core (effectively reserving cores for only real-time processing). RTLinux provides the capability of running special real-time tasks and interrupt handlers on the same machine as standard Linux. These tasks and handlers execute when they need to execute no matter what Linux is doing.

Conclusion:

Studied Real Time OS and RTLinux Kernel / source code.

References:

https://en.wikibooks.org/wiki/Embedded_Systems/Real-Time_Operating_Systems
<https://en.wikipedia.org/wiki/RTLinux>

Assignment 4

Title: Virtual Network Computing

a. Set up or enable VNC Server on the device

- Sometimes it is not convenient to work directly on the Raspberry Pi. Maybe you would like to work on it from another device by remote control.
- VNC is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer (running VNC Server) from another computer or mobile device (running VNC Viewer). VNC Viewer transmits the keyboard and either mouse or touch events to VNC Server, and receives updates to the screen in return.
- You will see the desktop of the Raspberry Pi inside a window on your computer or mobile device. You'll be able to control it as though you were working on the Raspberry Pi itself.
- Enabling VNC Server: On your Raspberry Pi, run the following commands to make sure you have the latest version of VNC Connect:
sudo apt-get update
sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer

Enabling VNC Server graphically

- On your Raspberry Pi, boot into the graphical desktop.
- Select Menu > Preferences > Raspberry Pi Configuration > Interfaces.
- Ensure VNC is enabled.

Enabling VNC Server at the command line

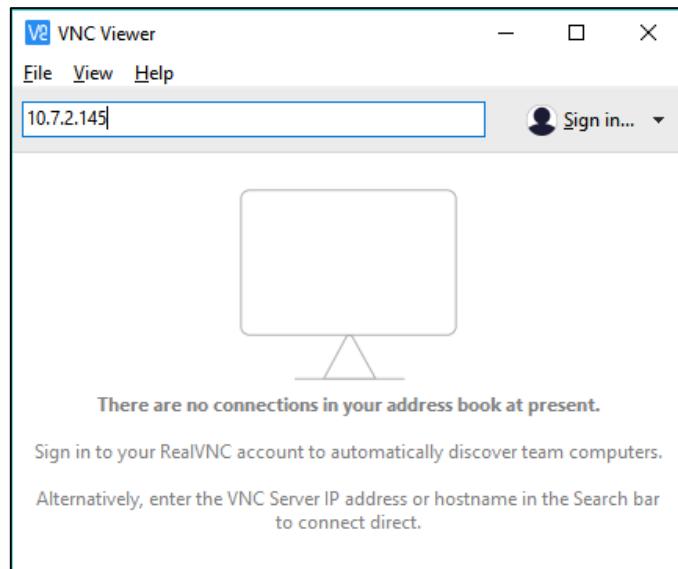
- You can enable VNC Server at the command line using raspi-config:
sudo raspi-config

Now, enable VNC Server by doing the following:

- Navigate to **Interfacing Options**.
- Scroll down and select **VNC > Yes**.

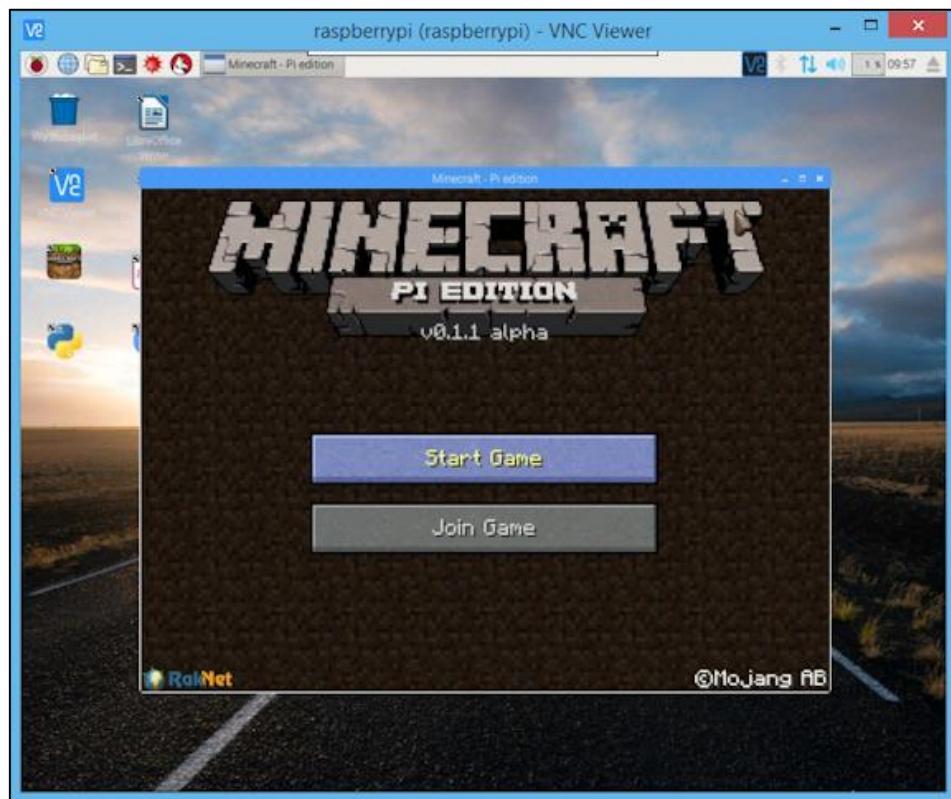
b. Install VNC viewer on a remote computer

- On your Raspberry Pi run ifconfig to discover your private IP address.
- On the device you'll use to take control, download VNC Viewer. For best results, use the compatible app from RealVNC.
- Enter your Raspberry Pi's private IP address into VNC Viewer:



c. Run VNC viewer and verify that it connects to raspberry pi 2

Enter the user name and password you normally use to log in to your user account on the Raspberry Pi. By default, these credentials are pi and raspberry.



Conclusion:

Studied and implemented Virtual Network Computing on Raspberry Pi kit.

References: <https://www.realvnc.com/>

Assignment 5

Title: Distributed Task Execution using WCF

- a. Implement a set of WCF services that expose the "task running" functionality as operations;
- b. Deploy the WCF services on several hosts/machines hosted as Windows services (they all must use the same contract);
- c. Configure the WCF services to use the Microsoft Message Queuing binding (netMsmqBinding), all using the same queue, in transactional mode;
- d. Configure the individual management applications to send their requests to the message queue, using the same netMsmqBinding;

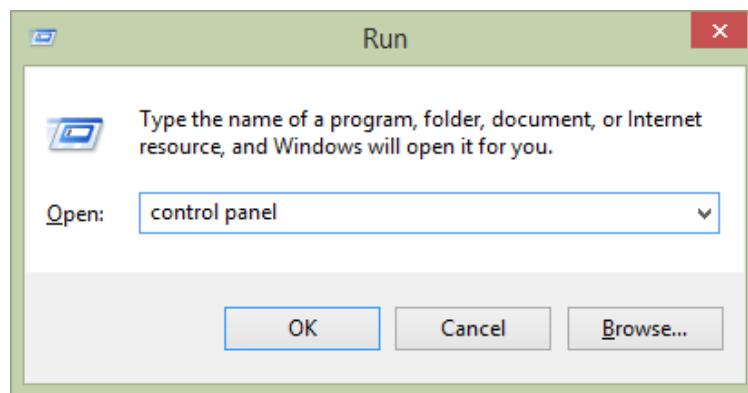
Theory:

MSMQ is Microsoft Message Queuing developed by Microsoft and deployed in windows operating system. MSMQ Queue ensures that reliable messaging between a client and a Windows Communication Foundation (WCF) service. So first we need to enable this feature in our operating system.

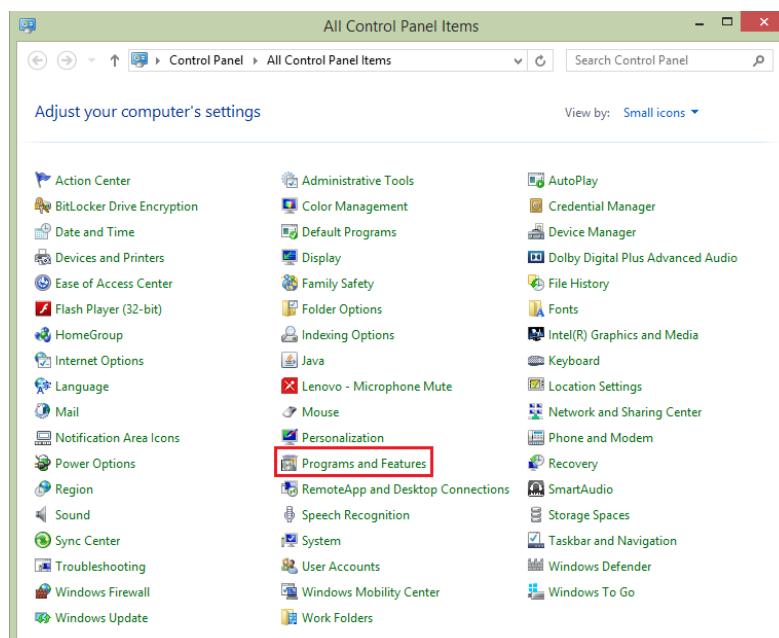
- **How to enable MSMQ**

Press Window key + R, it will open Run window.

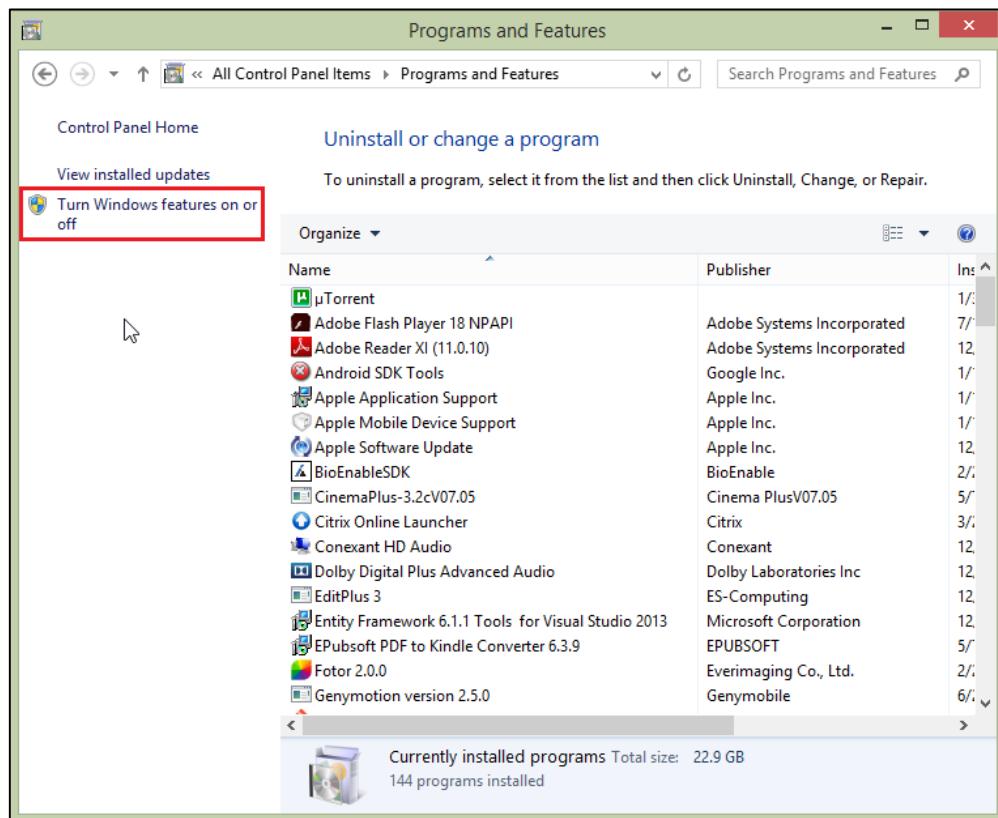
Enter 'Control Panel' and press 'OK' button, it will open control panel,



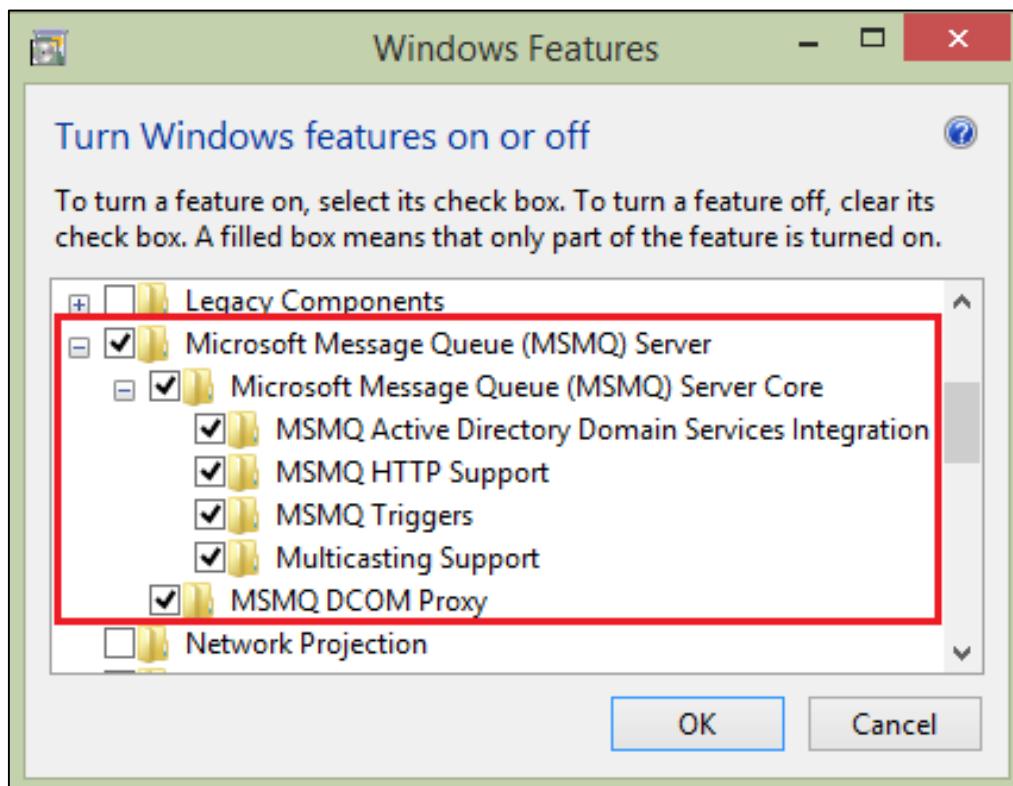
Click on 'Programs and Features',



From the left panel click on 'Turn Windows features on or off',



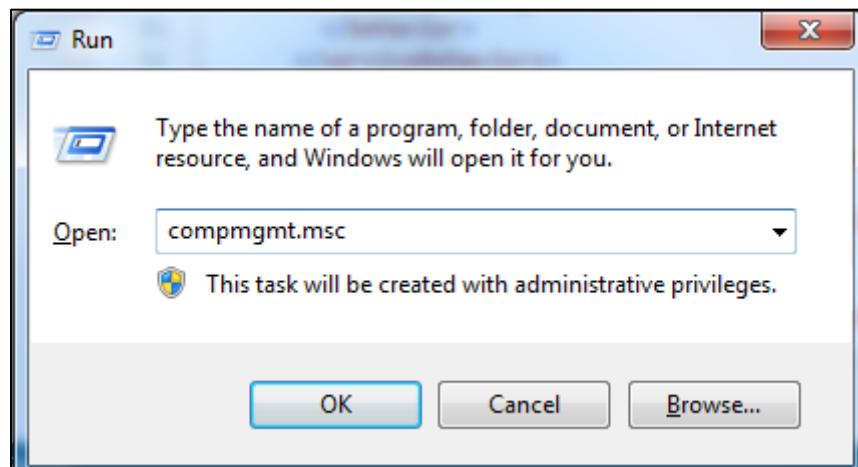
Select 'Microsoft Message Queue (MSMQ) Server' and all its child options.
Press 'OK' button, it will enable MSMQ for your operating system,



- **How to create a queue?**

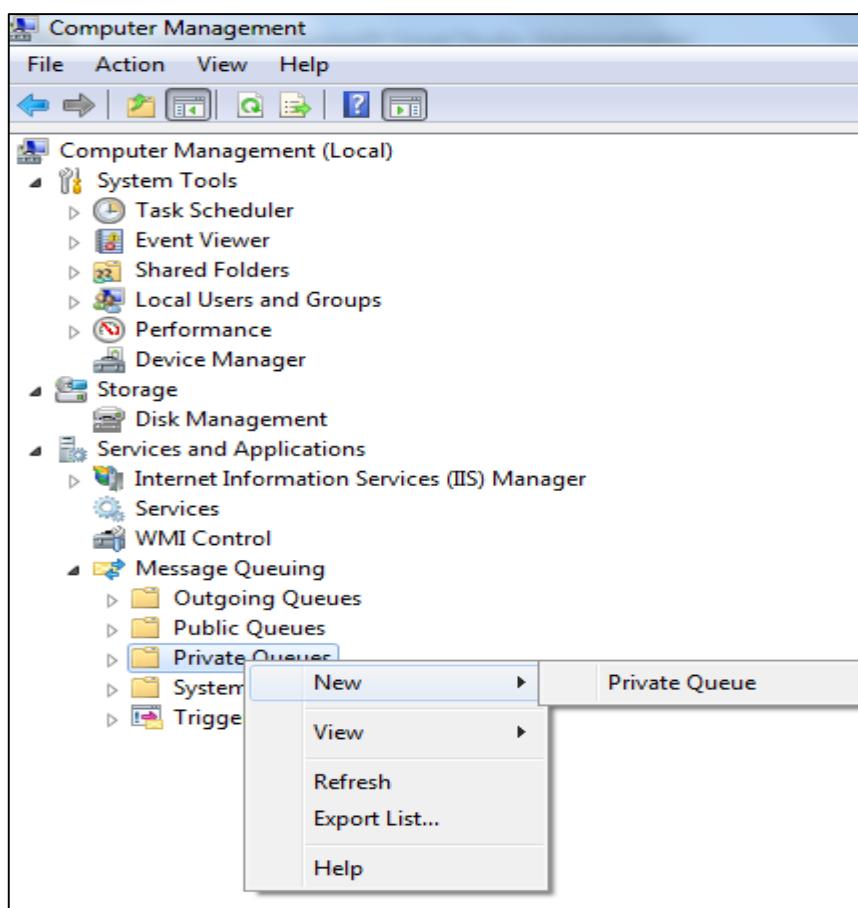
Press Window key + R, it will open Run window.

Enter 'compmgmt.msc' and press 'OK' button, it will open Computer Management window.

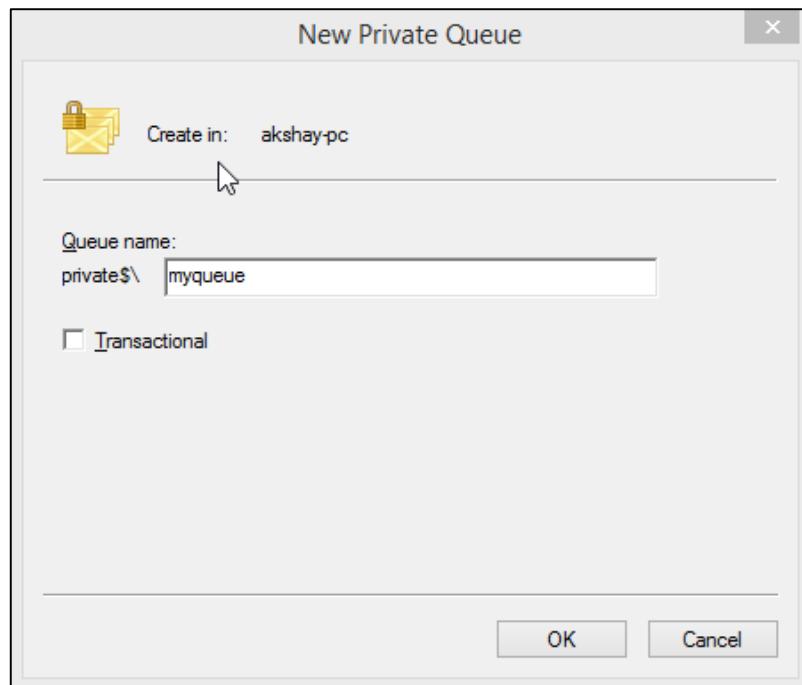


Under 'Services and Applications' expand 'Message Queuing'.

Right click on 'Private Queueus', select 'New' and click on 'Private Queue'.



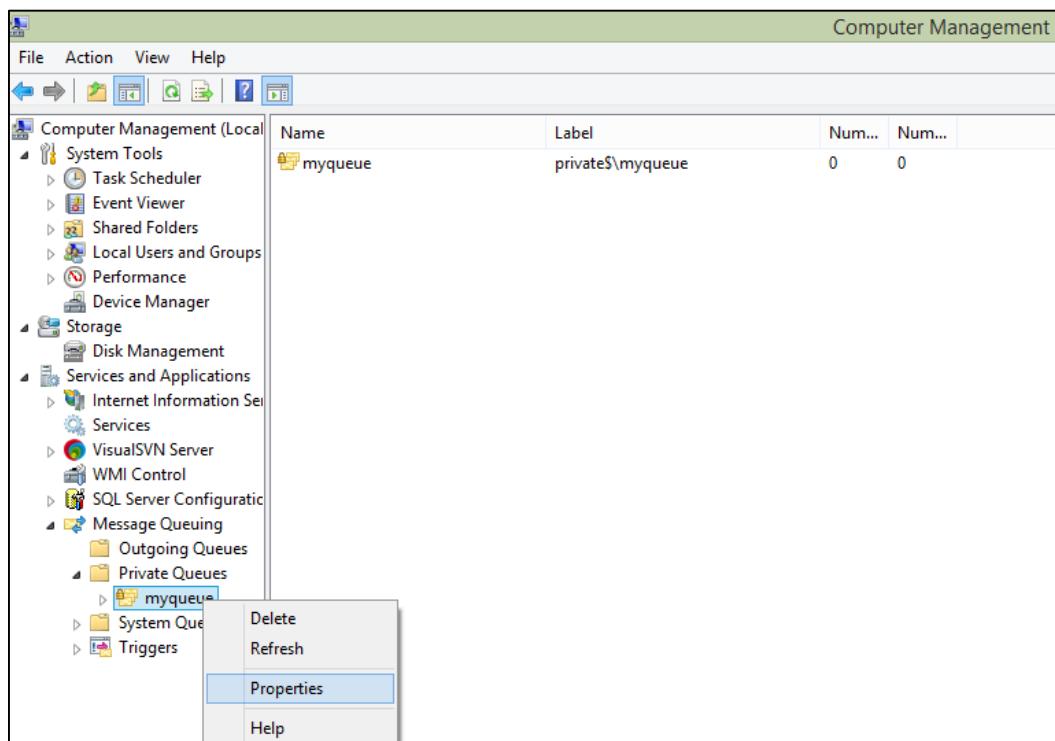
Enter queue name, here enter 'myqueue', which we will use in our application later.



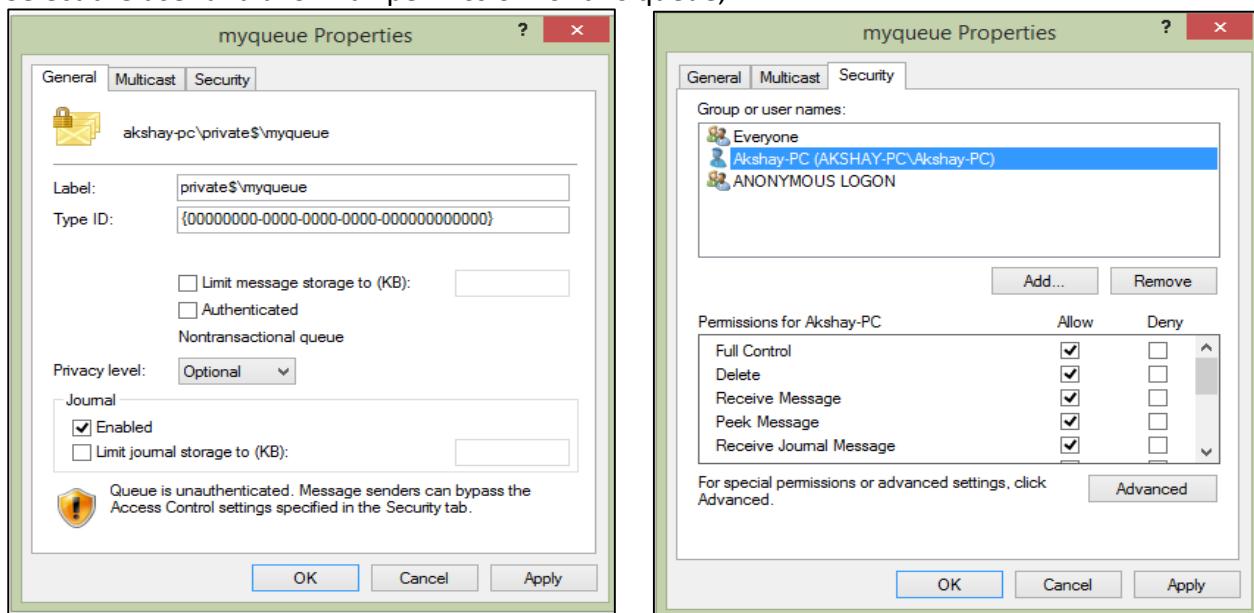
You can find preceding queue under Private Queues.

Select 'Private Queues' and you will have list of all queues with details like Name, Label, Number of Messages and Number of Journal Messages.

Right click on 'myqueue' and select 'Properties',

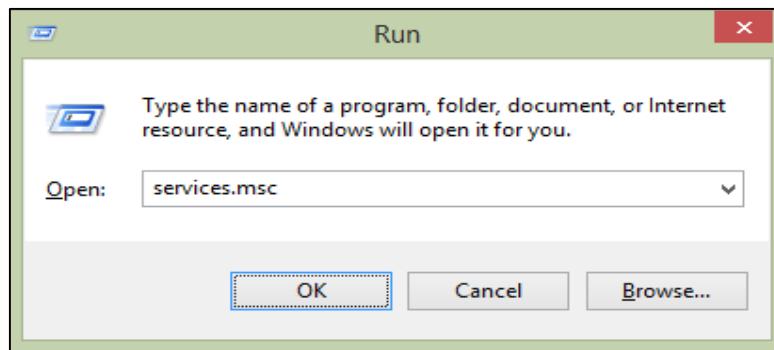


Check 'Journal Enabled' checkbox,
Select the user and allow full permission for this queue,



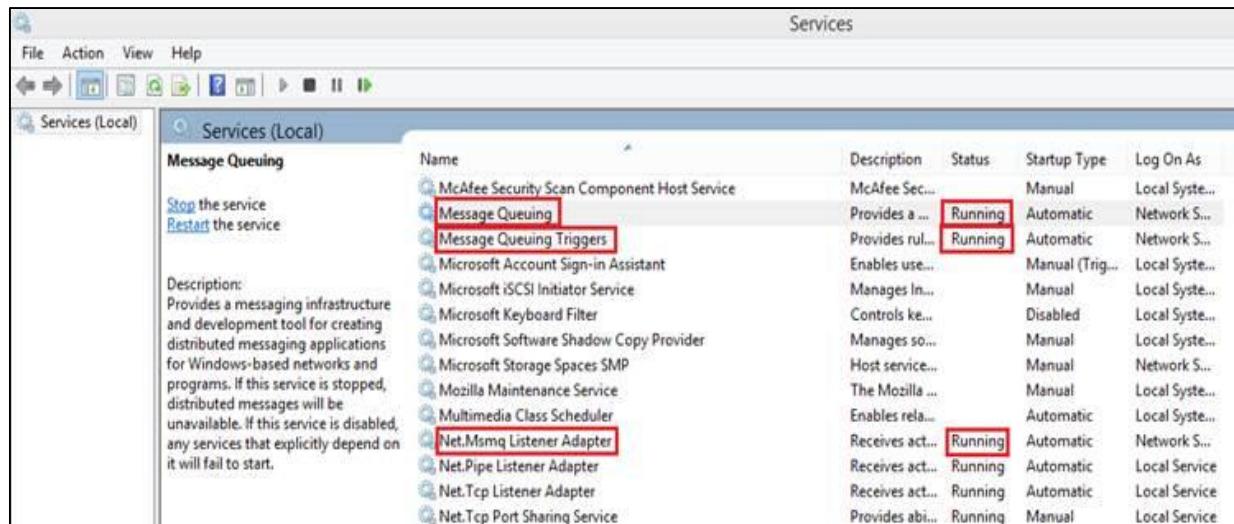
- Verify Queue Service is started**

After creating and setting permissions for the queue, we need to verify if queue service is running or not. Press Window key + R, it will open Run window, Enter 'services.msc' and press 'OK' button, it will open Services window.



Change below services status to 'Running',

- Message Queuing
- Message Queuing Triggers
- Net.Msmq Listener Adapter



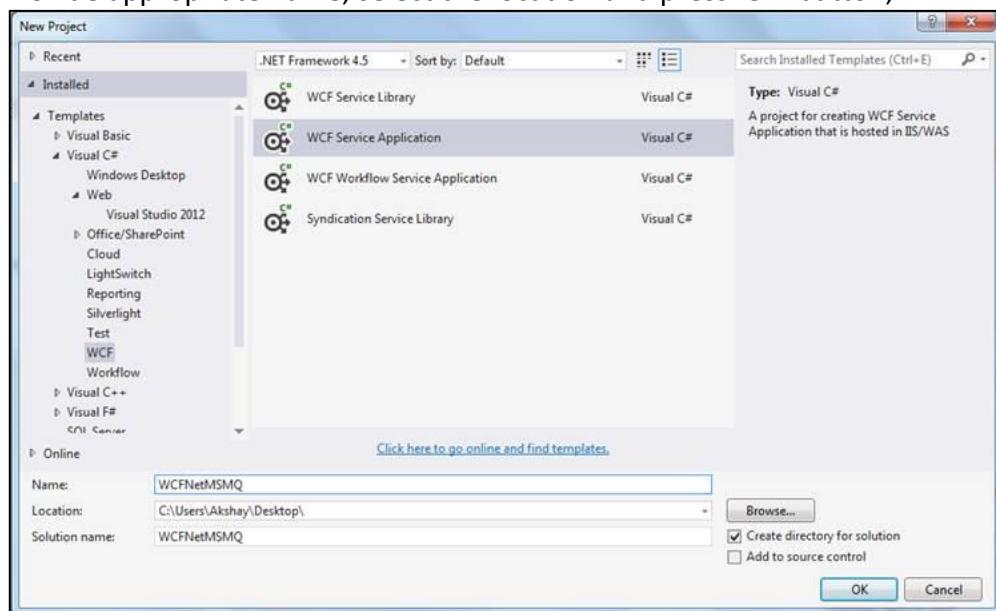
- **Create WCF Service Application**

Open Visual Studio Editor,

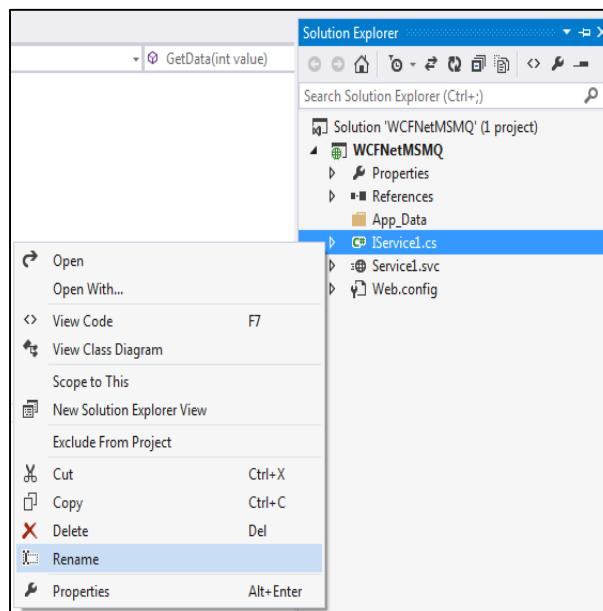
Select ‘File’ menu, expand ‘New’ and click on ‘Projects’,

Select ‘WCF’ from left panel, select ‘WCF Service Application’,

Provide appropriate name, select the location and press ‘OK’ button,

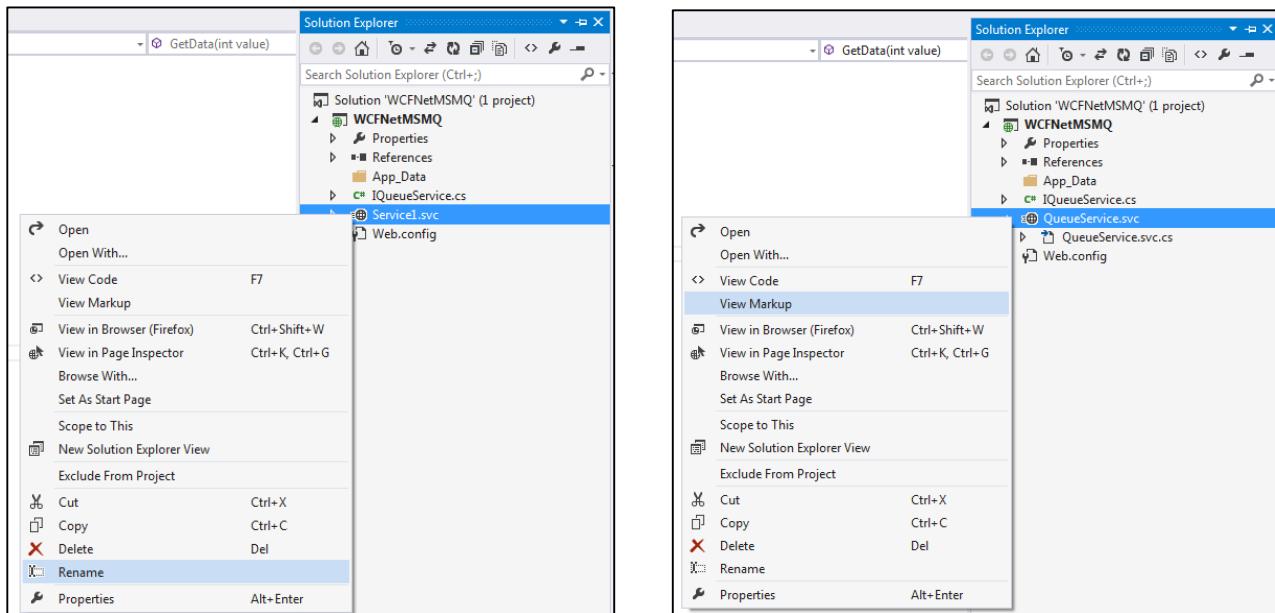


Right click on interface IService1 and rename it to IQueueService,



Right click on Service1.svc and rename it to 'QueueService.svc',

Right click on 'QueueService.svc' and click on 'View Markup',



Change the name of service to 'WCFNetMSMQ.QueueService',

1. <%@ ServiceHost Language="C#" Debug="true" Service="WCFNetMSMQ.QueueService" CodeBehind="QueueService.svc.cs" %>

Configure service and binding in the web.config under Service Model section.

Under service we need to set endpoint with address, binding, contract and binding configuration,

- Address – set queue address with net.msmq, use queuename which we have created previously in the last.
- Binding – binding name which we will configure under bindings section.
- Contract – Provide the interface name

1. <services>
2. <service name="WCFNetMSMQ.QueueService">
3. <endpoint address="net.msmq://localhost/private/myqueue"
4. binding="netMsmqBinding"
5. bindingConfiguration="MsmqBindingConfig"
6. contract="WCFNetMSMQ.IQueueService" />
7. </service>
8. </services>

Configure binding under netMsmqBinding section. Set security as none.

1. <bindings>
2. <netMsmqBinding>
3. <binding name="MsmqBindingConfig" exactlyOnce="false">

```
4.      <security mode="None" />
5.    </binding>
6.  </netMsmqBinding>
7. </bindings>
```

In the interface, we have operation contract namely 'GetData', Set 'IsOneWay' attribute as true and make return type as void.

```
1. namespace WCFNetMSMQ
2. {
3.   [ServiceContract]
4.   public interface IQueueService
5.   {
6.     [OperationContract(IsOneWay=true)]
7.     void GetData(int value);
8.   }
9. }
```

At the implementation part, trace the value which you are going to pass as parameter in the operation contract.

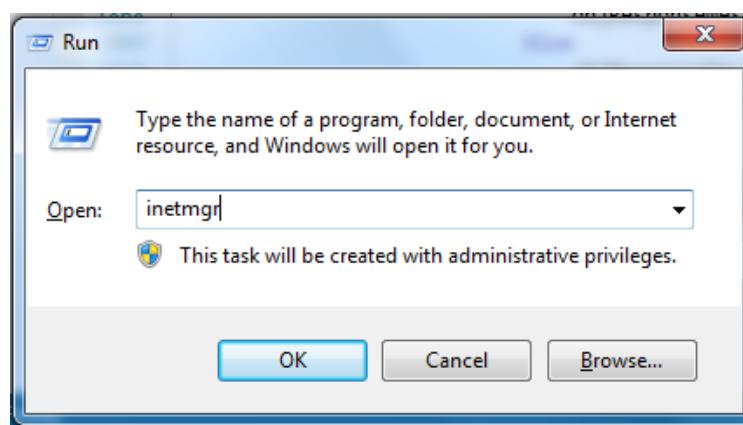
```
1. namespace WCFNetMSMQ
2. {
3.   public class QueueService : IQueueService
4.   {
5.     public void GetData(int value)
6.     {
7.       System.Diagnostics.Trace.WriteLine(string.Format("You entered: {0}", value));
8.     }
9.   }
10. }
```

Now we are done with the service application having operation contract which accepts integer number and log it. In the next step we need to host our application in IIS.

- **Host WCF Service Application in IIS**

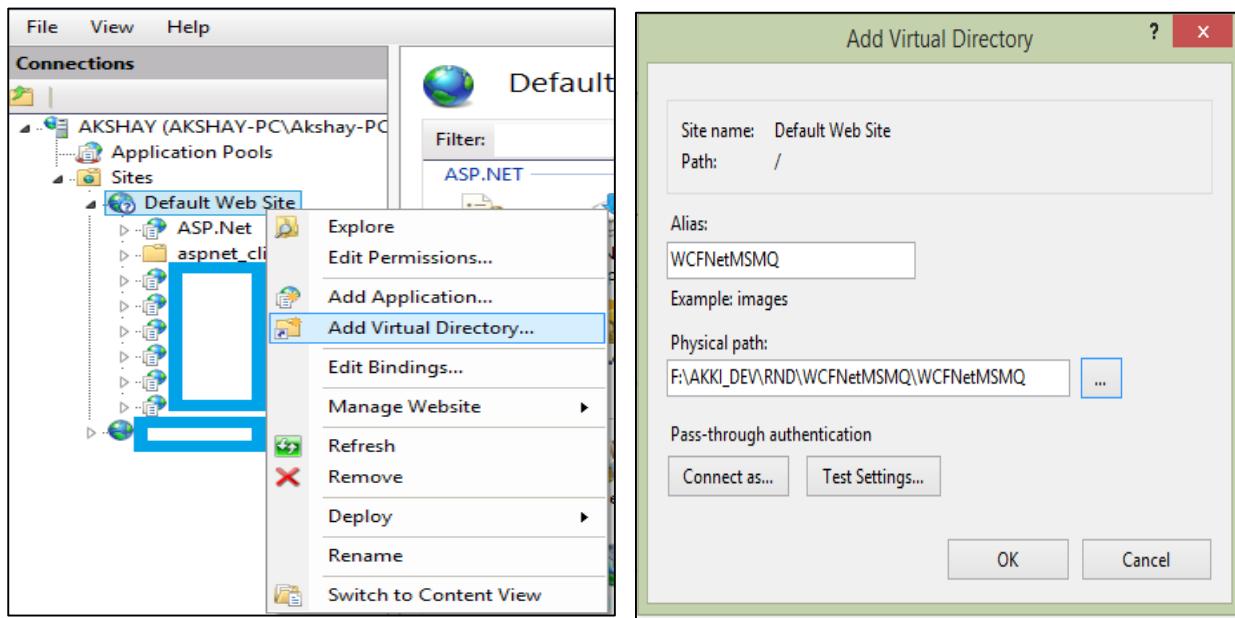
Press Window key + R, it will open Run window.

Enter 'inetmgr' and press 'OK' button, it will open IIS window.



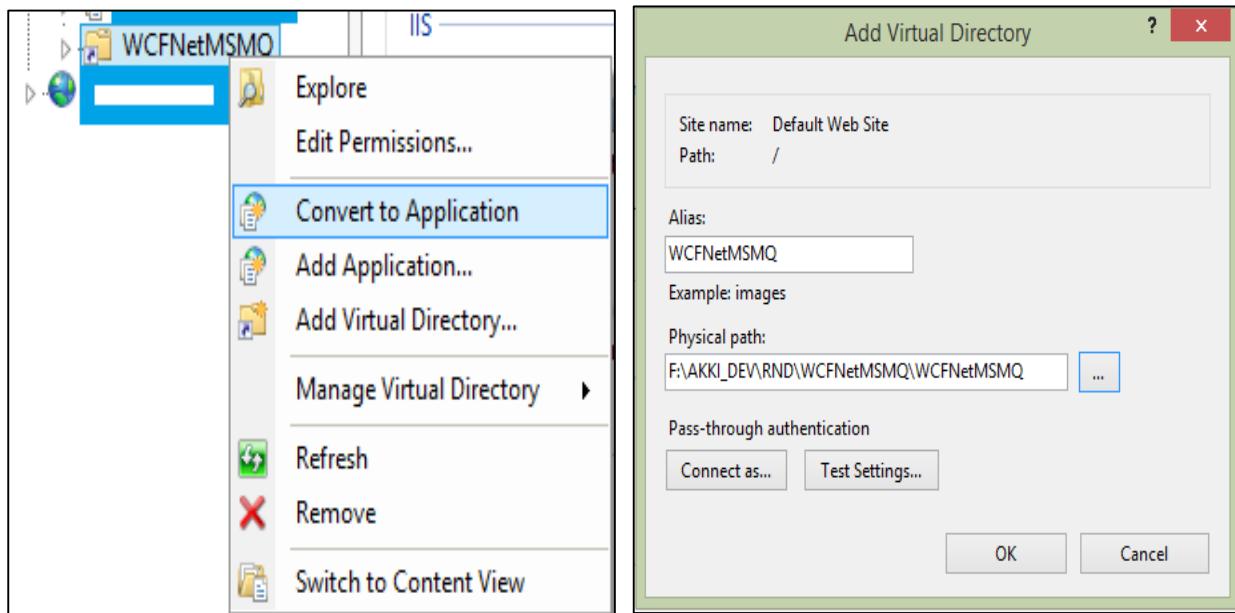
Expand the 'Sites' and you will find 'Default Web Site'.

Right click on 'Default Web Site' and click on 'Add Virtual Directory...' in order to host the application, provide alias name and select our application physical path. Press 'OK' button,



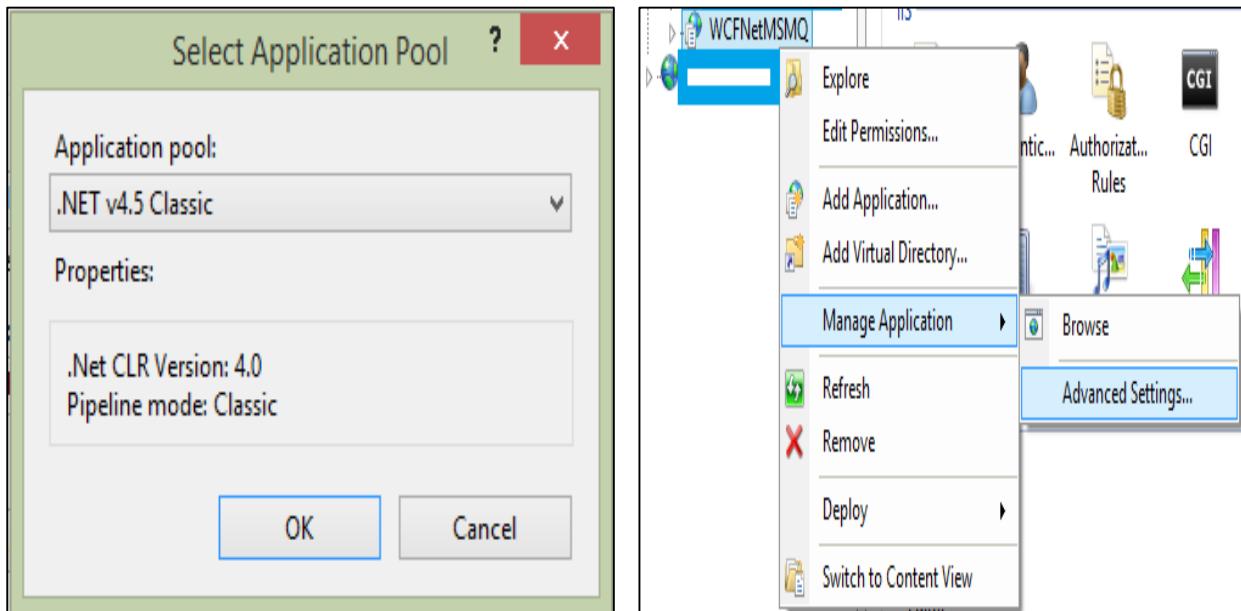
Right click on 'WCFNetMSMQ' and click on 'Convert to Application'.

Click on 'Select...' button and select the application pool.



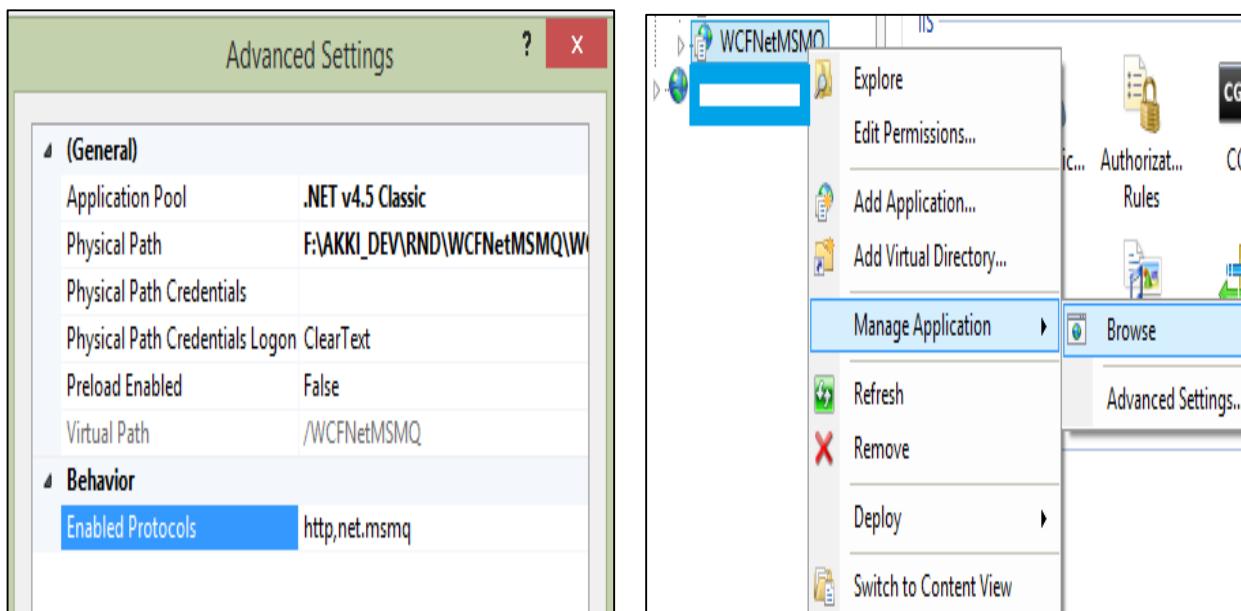
Select '.NET v4.5 Classic' application pool and press 'OK' button.

In order to enable net.msmq protocol for this application, Right click on 'WCFNetMSMQ', expand 'Manage Application' and click on 'Advanced Settings...'.

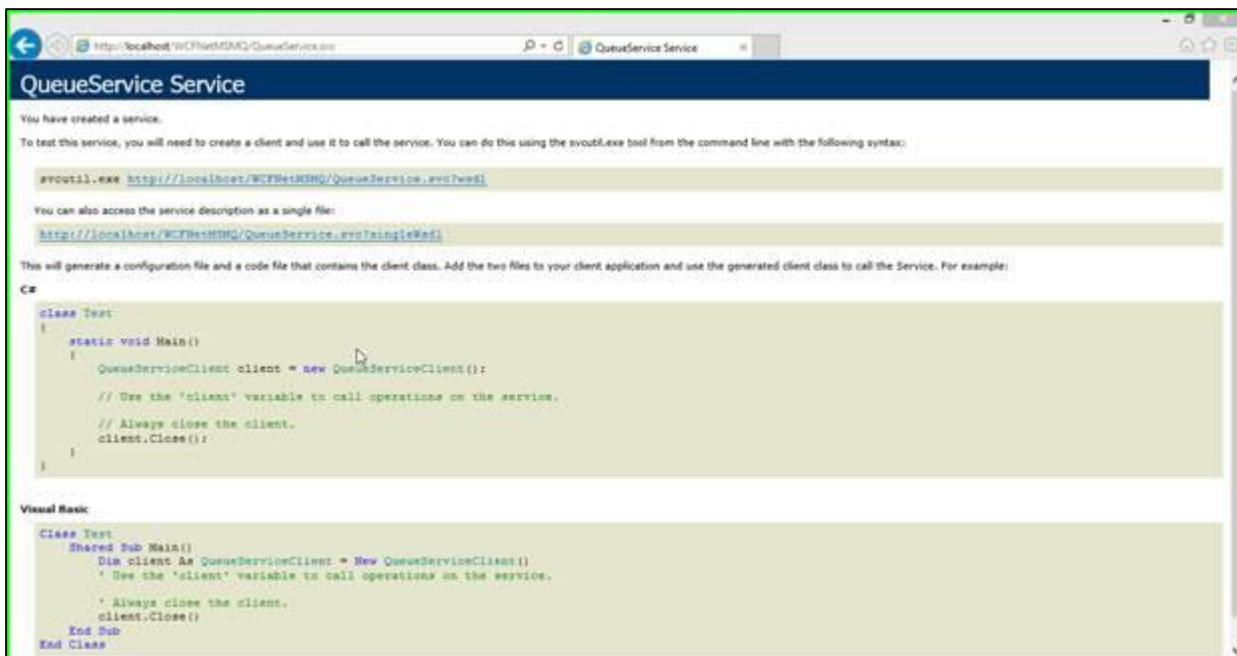


In Enabled Protocols add net.msmq along with http, It will enable net.msmq protocol for our application.

Now check whether configured site is working or not, Right click on 'WCFNetMSMQ', expand 'Manage Application' and click on 'Browse'.



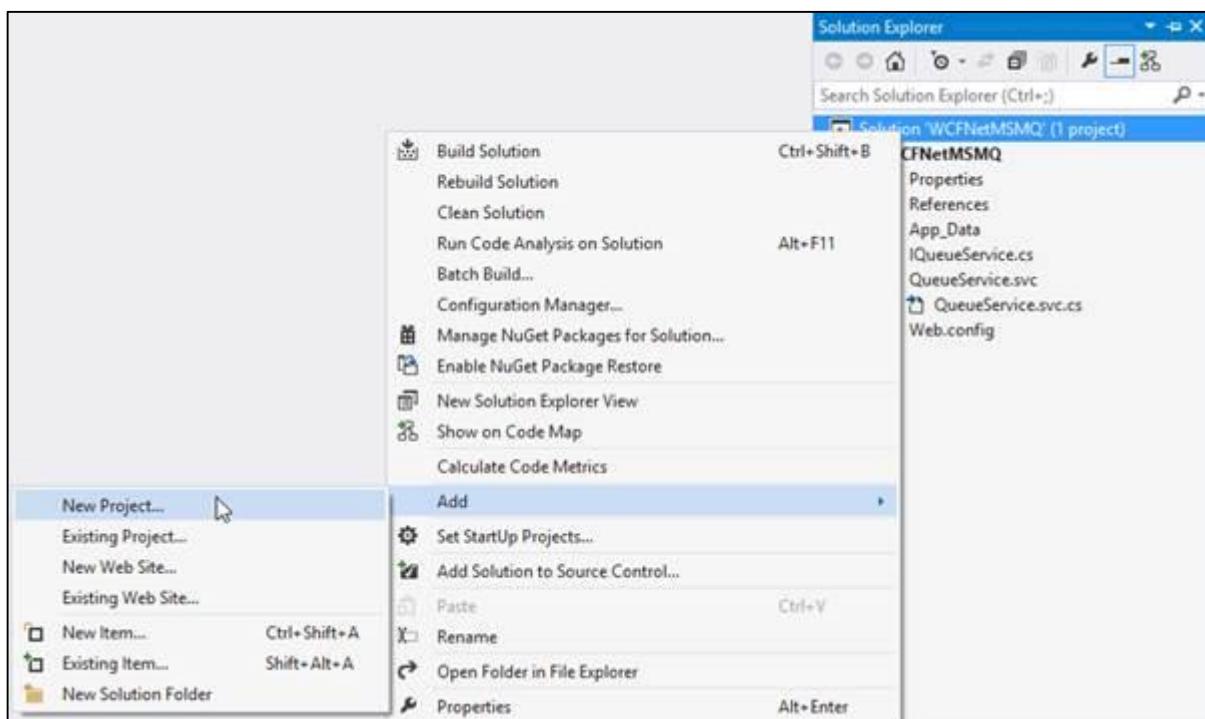
In the browser we can see that service is running fine.



Now it's time to create client application to consume service.

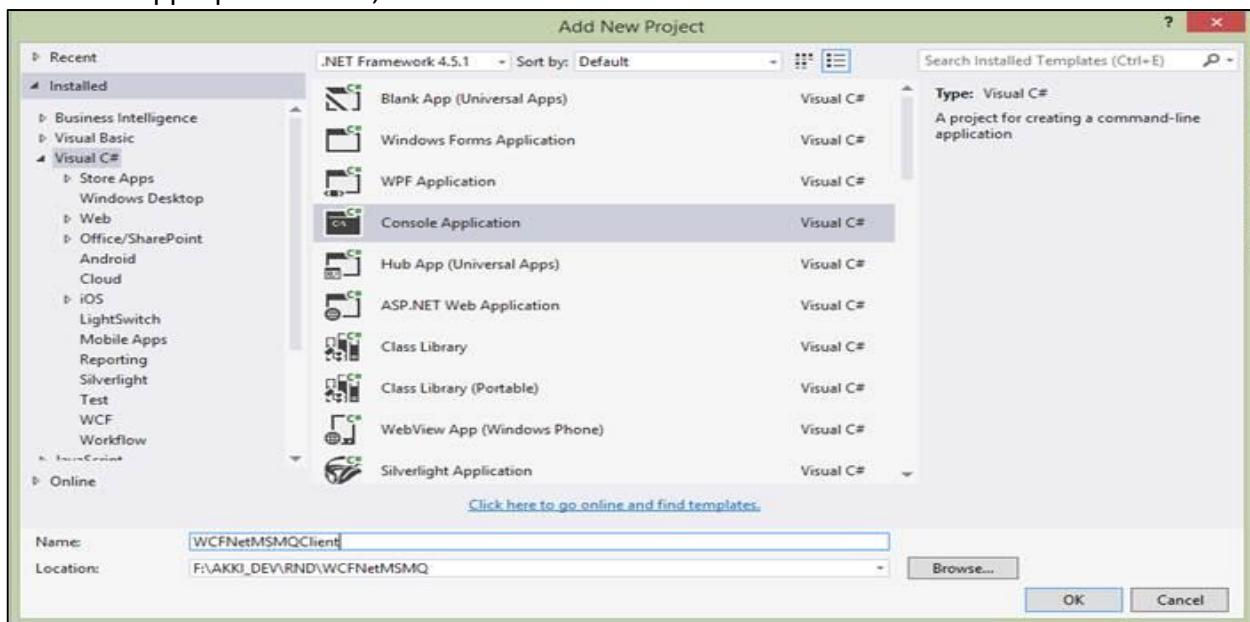
- **Create Client Application**

Right click on Solution, expand ‘Add’ and click on ‘New Project...’

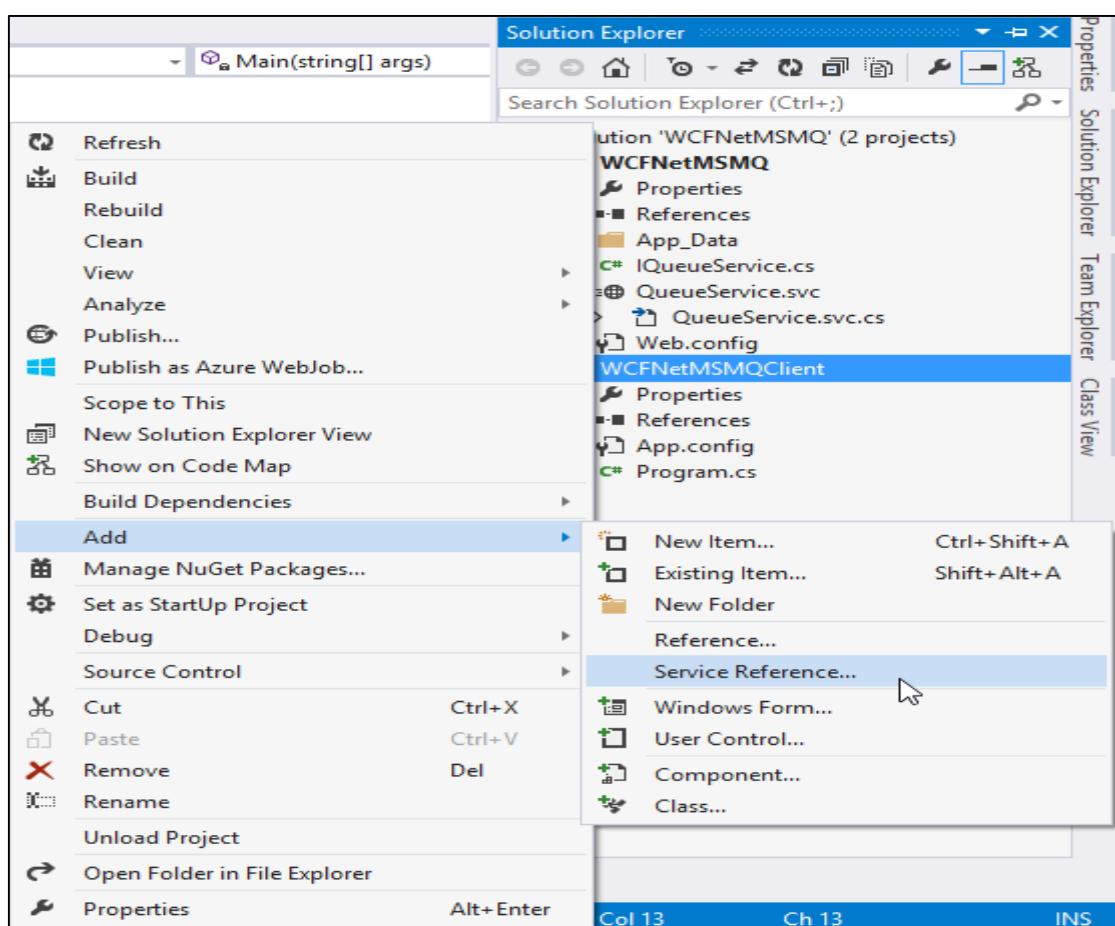


Select ‘Visual C#’ from the left panel and select ‘Console Application’.

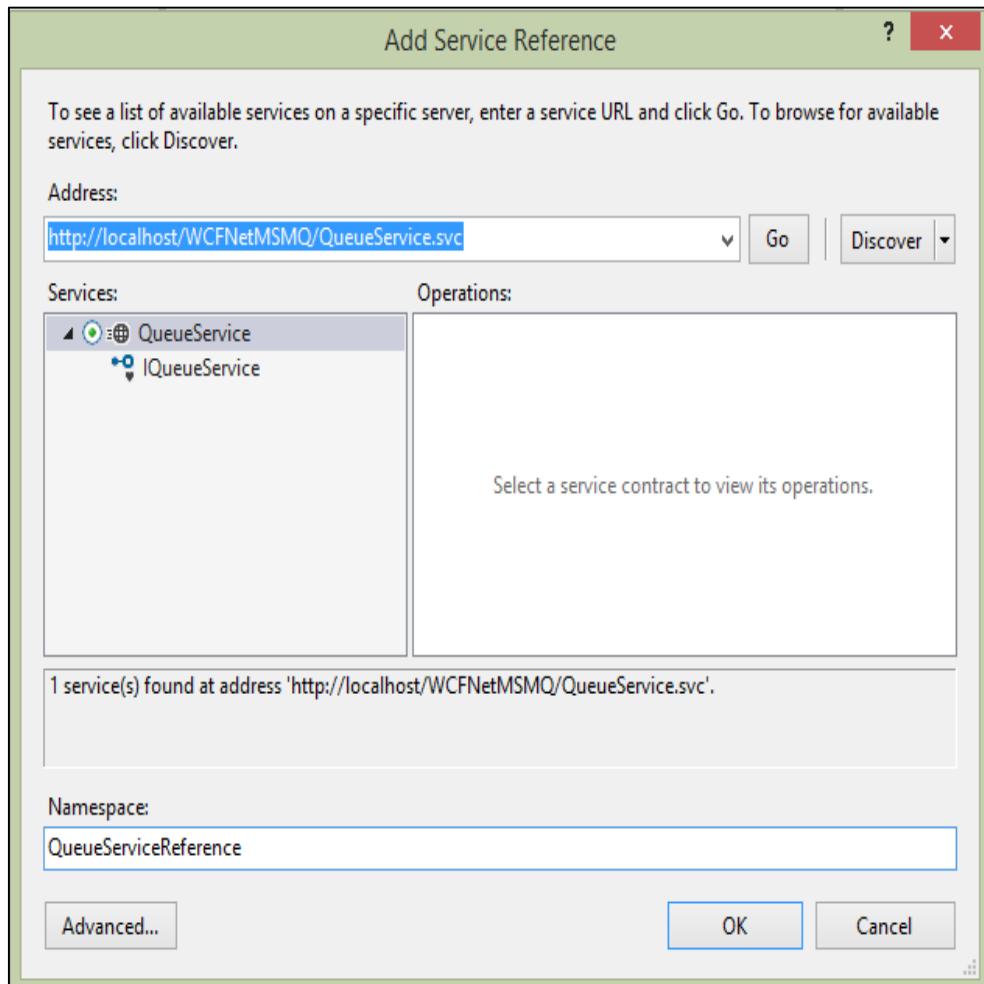
Provide appropriate name, select location and click on 'OK' button.



Add service reference in our client application, Right click on the project, expand 'Add' and click on 'Service Reference...'.



Browse application from IIs and copy address from the address bar of the browser.
Paste same address in the 'Add Service Reference' window and press 'Go' Button.
Select 'Queue Service' from Services list.
Give the namespace and press 'OK' button.

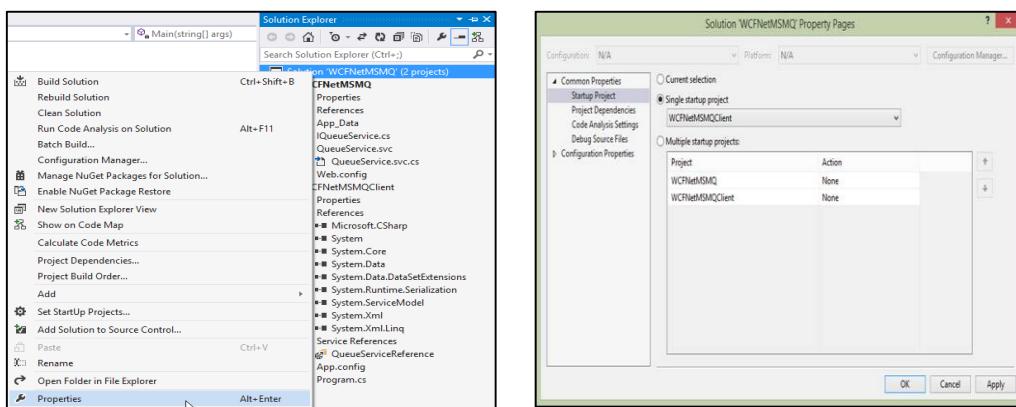


- **Create an instance of QueueService client.**

Call GetData operation contract using this instance,

1. namespace WCFNetMSMQClient
2. {
3. class Program
4. {
5. static void Main(string[] args)
6. {
7. QueueServiceReference.QueueServiceClient serviceClient = new QueueServiceReference.QueueServiceClient();
8. serviceClient.GetData(5677);
9. }
10. }
11. }

In order to set client application as startup project, right click on solution and click on 'Properties', Select client application under 'Single Startup Project',



Now run the project and check 'myqueue' queue.

You can see that one message is added in Journal Messages.

Name	Label	Number of Messages	Number of Journal Messages
myqueue	private\$myqueue	0	1

Select Journal Messages under myqueue and you can see the message details like label, priority, class, and size and message id.

Conclusion: The main benefit of using queue mechanism is if the destination server is offline when the client sends a message, the message will be queued on the client until the server comes back online.

References:

<https://www.codeproject.com/Articles/326909/Creating-a-WCF-Service-with-MSMQ-Communication-and>

Assignment 6

Title: Study of Android 7.0 Nougat & RTAndroid

Real-Time Android (RTAndroid)

RTAndroid makes real-time Android possible. Our platform provides support for applications with real-time requirements while preserving all benefits of Android. Extended system components augment the new platform with a predictable and deterministic real-time behaviour. By creating a reliable execution environment for general purpose mobile devices, RTAndroid can be used as a low-cost platform for data monitoring and visualization or in the field of industrial automation and control.

The platform has been carefully designed based on more than 5 years of research experience in Android and real-time systems. Started with Android 2, our latest update package is based on the latest Android 7.1.1 release, preserving the standard Android application programming model, as well as the full compatibility to already existing Android components and third-party software.

	Android 7.0 Nougat	RTAndroid
Support for devices	Wide variety of devices including mobile phones, tablets, TVs, etc. from different hardware manufacturers	Limited Devices are supported: Samsung Galaxy S4, ODROID XU3 / XU4, Raspberry Pi 3, Google Nexus 10
Features	<ul style="list-style-type: none"> • WIFI • Bluetooth • Audio Jack • Various Sensors Support 	<ul style="list-style-type: none"> • Audio Jack • HDMI Audio • Ethernet • WIFI • Bluetooth
Google Play Support	Google Play comes preinstalled	No, packaging Google Play Services by default is not allowed. But we include an additional script "gapps.sh", which can be used to install OpenGApps.
Known Issues	<ul style="list-style-type: none"> • performance is known to be periodically slow • Battery life may be regressed • Some apps may not function normally • System UI crash after toggling screen brightness, then going to Overview and selecting a new app. • Taps in Quick Settings may be registered in another location. 	<ul style="list-style-type: none"> • Video playback • Music application • Gallery application

Installing RTAndroid on Raspberry Pi

Download RTAndroid OS image file from official site.

URL: <https://rtandroid.embedded.rwth-aachen.de/downloads/raspberry-pi/>

Flashing RTAndroid

This section describes how to flash RTAndroid. Please use a Linux distribution for the installation process. Windows users should use a Live-CD or VirtualBox.

Once you have downloaded the package, it can be flashed as follows.

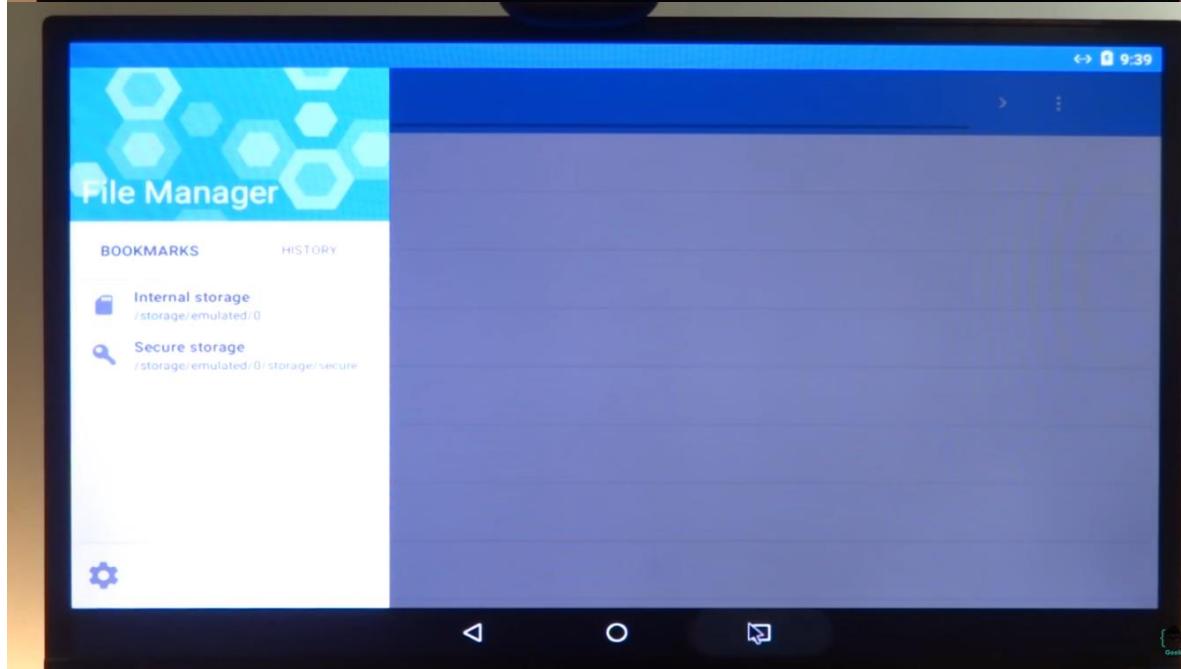
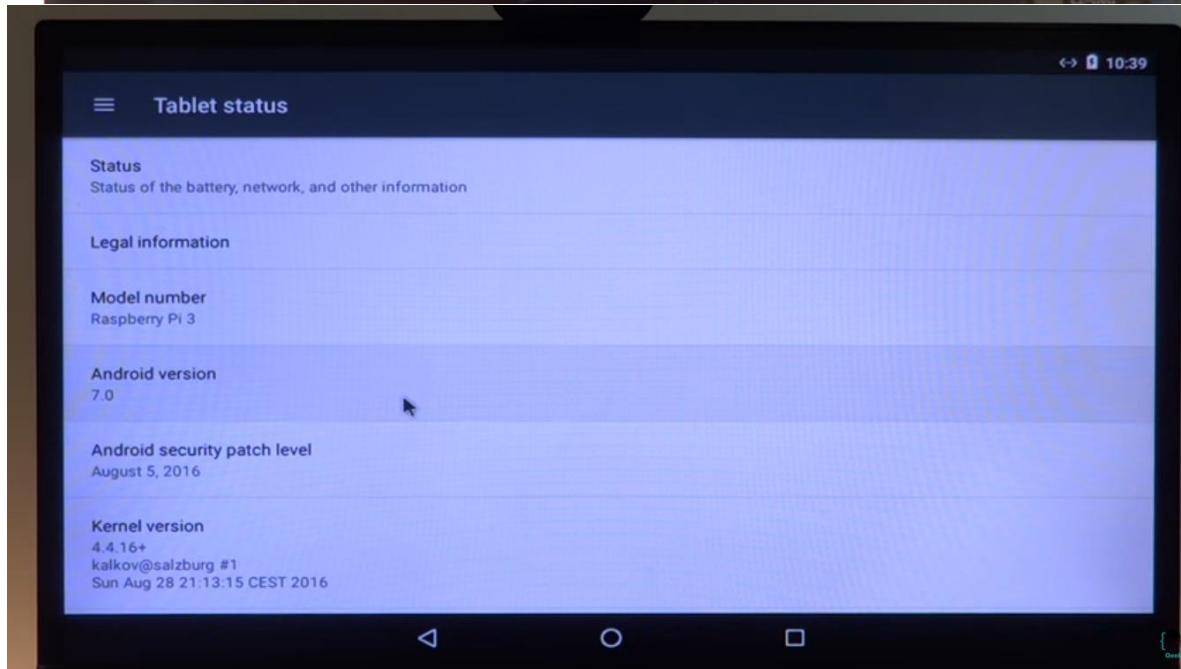
1. Make sure you have a micro SD card with UHS speed class of 4 or higher.
Slow memory cards will slow down the system significantly.
2. Unpack the downloaded zip package.

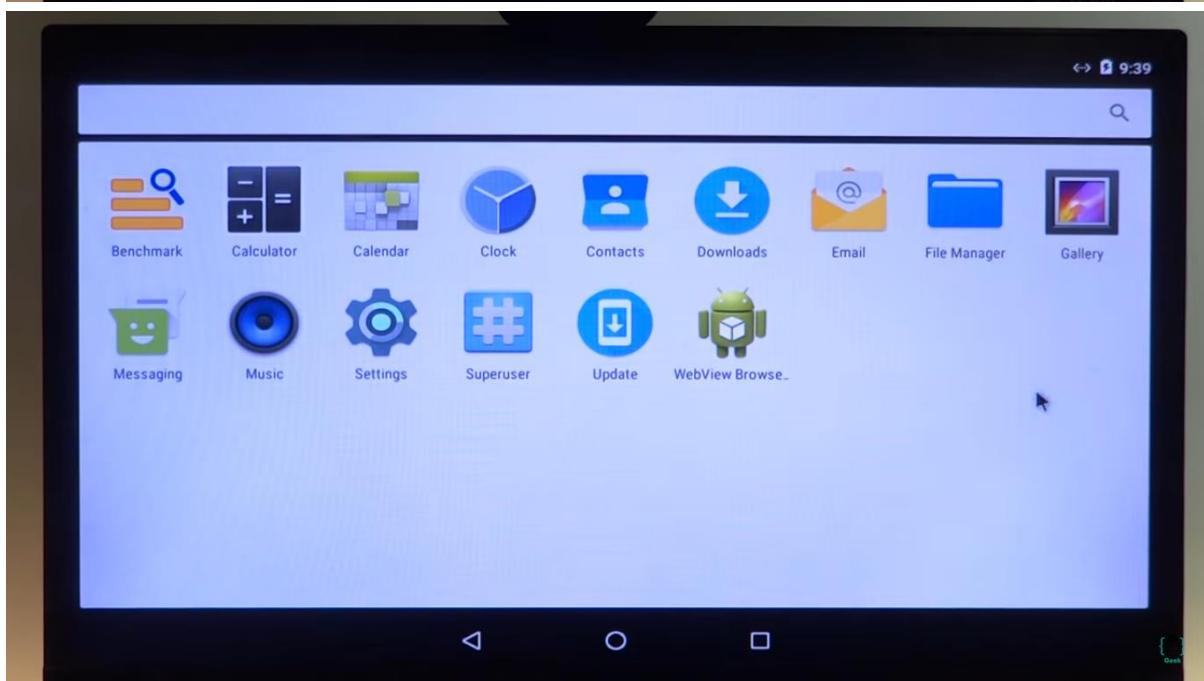
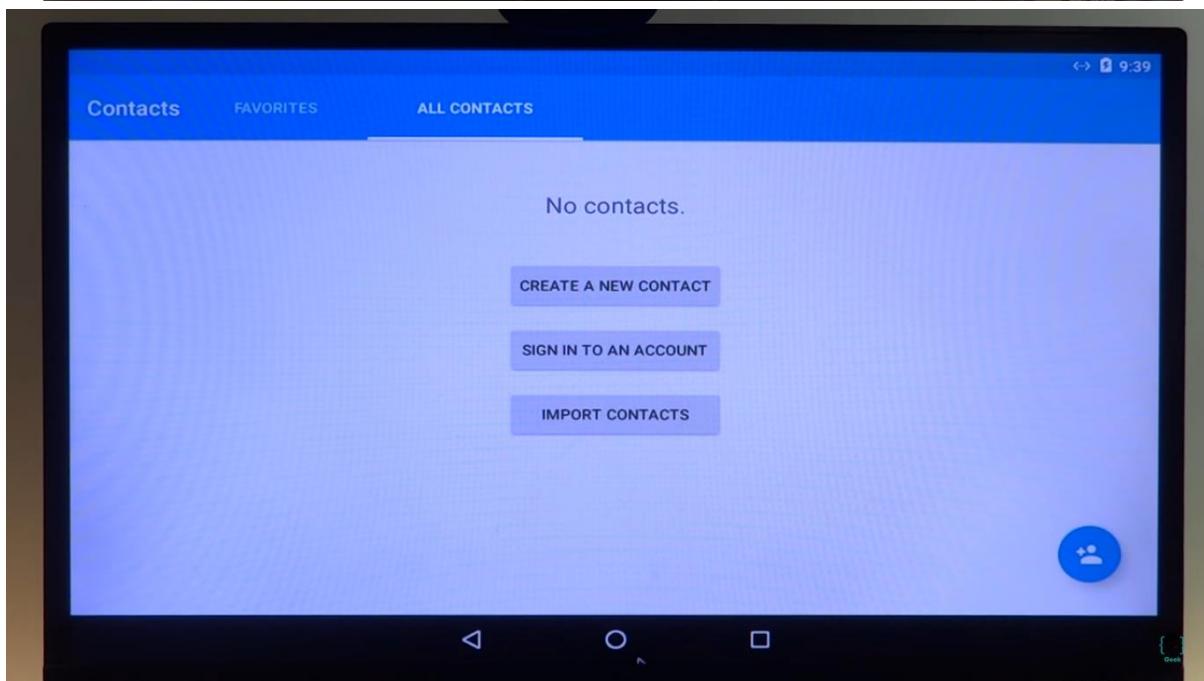
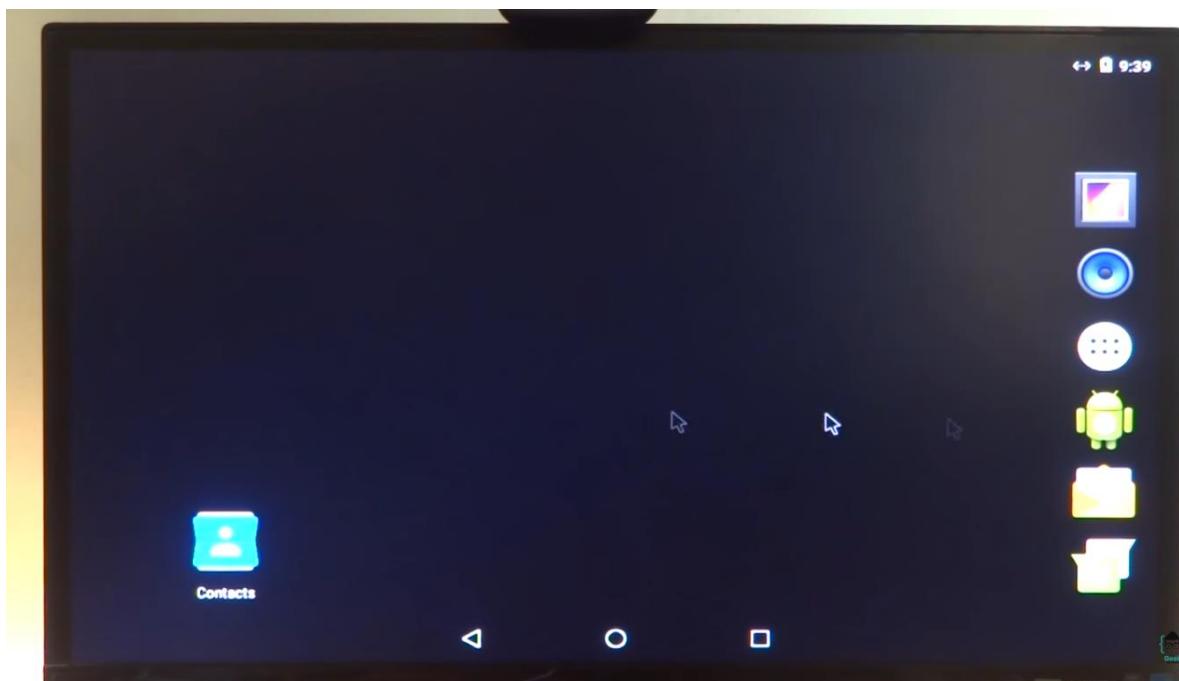
```
unzip rtandroid-[VERSION]-[DATE]-rpi3.zip -d
```

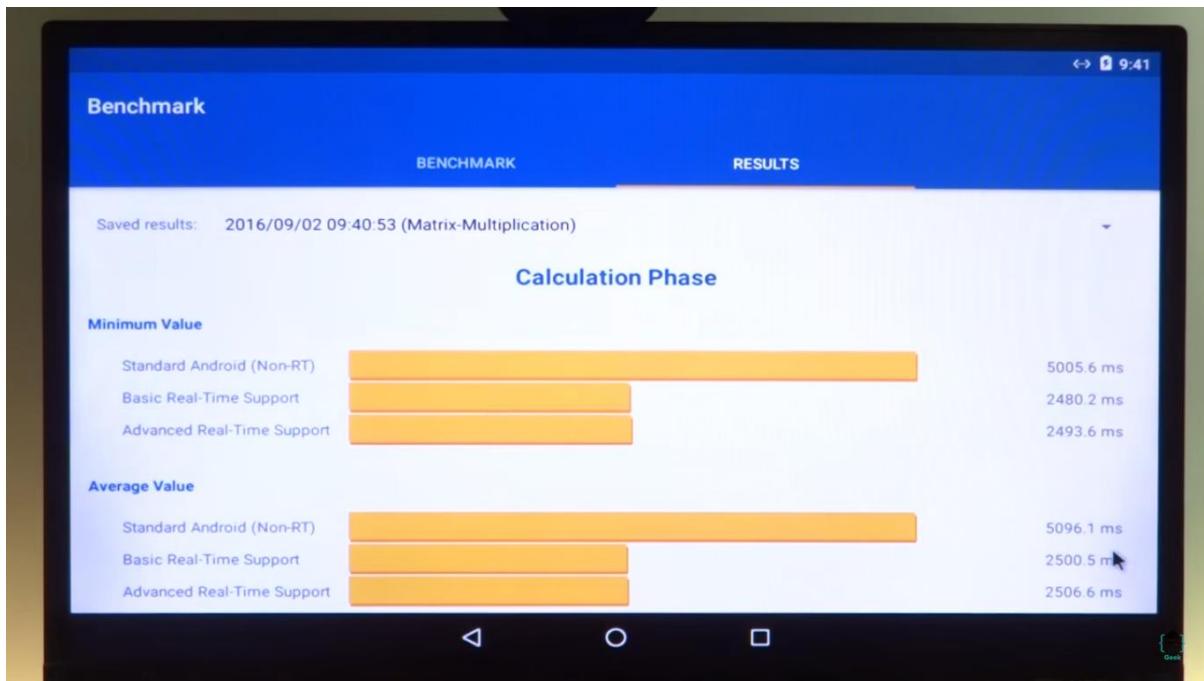
3. Insert the memory card into your computer. All data on this card will be lost.
4. Execute the install script from the downloaded package:
 - Find the device name where the card is mounted (example: /dev/sdc)
`sudo fdisk -l`
 - Use the "-p" flag to partition your card.
 - Use the "-f" flag to format your card.
`./install.sh -p -f /dev/sdc`

5. Execute the gapps script from the downloaded package, if desired.

Depending on the memory card the first system boot can take up to 15 minutes until RTAndroid will be fully functional.

Snapshots:



**Conclusion:**

Studied Android 7.0 Nougat & RTAndroid also demonstrated RTAndroid on Raspberry Pi kit.

References:

<https://rtandroid.embedded.rwth-aachen.de/downloads/raspberry-pi/>

Assignment 7

Title: Development of Android Application using Android Studio.

Create an Android Project

1. In Android Studio, create a new project:
 - If you don't have a project opened, in the Welcome to Android Studio window, click Start a new Android Studio project.
 - If you have a project opened, select File > New Project.
2. In the New Project screen, enter the following values:
 - Application Name: "My First App"
 - Company Domain: "example.com"
3. Click Next.
4. In the Target Android Devices screen, keep the default values and click Next.
5. In the Add an Activity to Mobile screen, select Empty Activity and click Next.
6. In the Customize the Activity screen, keep the default values and click Finish.

After some processing, Android Studio opens the IDE. Now take a moment to review the most important files.

First, be sure the Project window is open (select View > Tool Windows > Project) and the Android view is selected from the drop-down list at the top of that window.

Run on an emulator

Before you run your app on an emulator, you need to create an Android Virtual Device (AVD) definition. An AVD definition specifies the characteristics of an Android phone, tablet, Android Wear, or Android TV device that you want to simulate in the Android Emulator.

Create an AVD Definition as follows:

1. Launch the Android Virtual Device Manager by selecting **Tools > Android > AVD Manager**, or by clicking the AVD Manager icon  in the toolbar.
2. In the **Your Virtual Devices** screen, click **Create Virtual Device**.
3. In the **Select Hardware** screen, select a phone device, such as Pixel, and then click **Next**.
4. In the **System Image** screen, click **Download** for one of the recommended system images. Agree to the terms to complete the download.
5. After the download is complete, select the system image from the list and click **Next**.
6. On the next screen, leave all the configuration settings as they are and click **Finish**.
7. Back in the **Your Virtual Devices** screen, select the device you just created and click **launch this AVD in the emulator** .

While the emulator starts up, close the Android Virtual Device Manager window and return to your project so you can run the app:

1. Once the emulator is booted up, click the **app** module in the **Project** window and then select **Run > Run** (or click **Run**  in the toolbar).
2. In the **Select Deployment Target** window, select the emulator and click **OK**.

Android Studio installs the app on the emulator and starts it.

A Simple Android Application for Adding Two Numbers

Open up the activity java file from `src/com.example.addition`. Declare a few variables before the `onCreate` function.

```
EditText firstNumber;
EditText secondNumber;
TextView addResult;
Button btnAdd;

double num1,num2,sum;
```

In the `onCreate` function once the content view has been set, we'll read the values entered in the Text Views using an id which we have set in the XML code above.

```
firstNumber = (EditText)findViewById(R.id.txtNumber1);
secondNumber = (EditText)findViewById(R.id.txtNumber2);
addResult = (TextView)findViewById(R.id.txtResult);
btnAdd = (Button)findViewById(R.id.btnAdd);
```

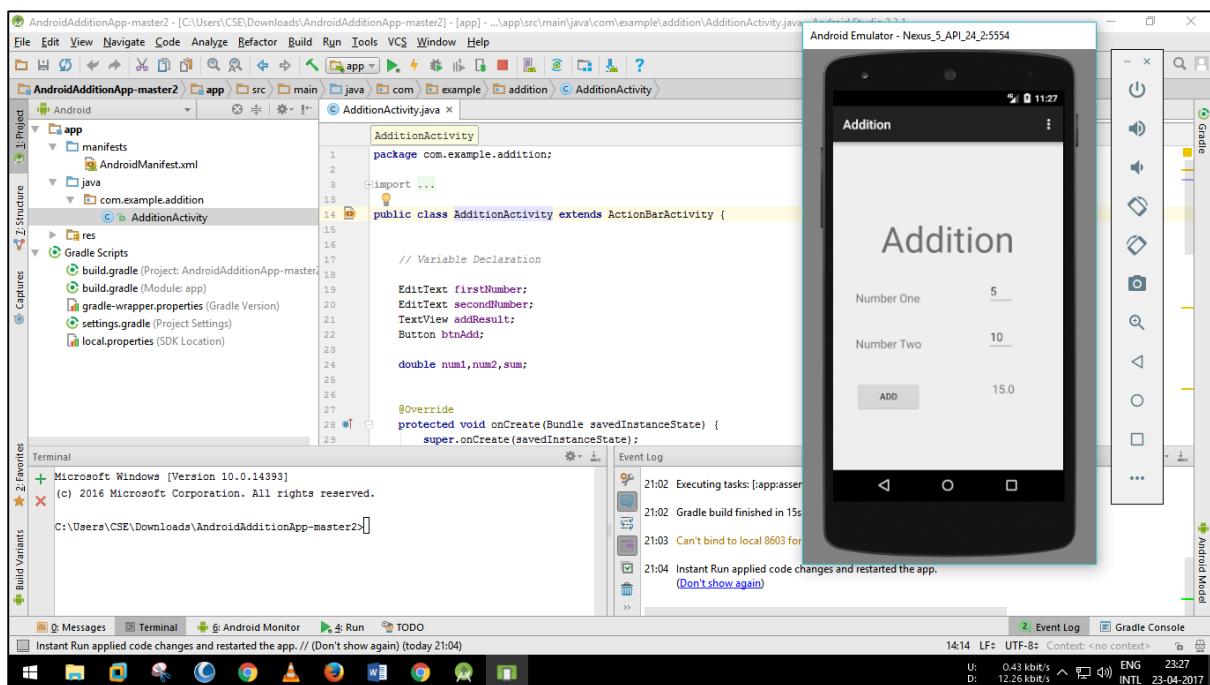
When the Add button has been clicked we need to add the values entered in the text boxes, sum it up and render the output in the `txtResult` text view. So, first we need to add a click listener to the Add button.

```
btnAdd.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // code will be here
    }
});
```

Inside the on click listener, we'll add the numbers and set the sum to the `txtResult` Text View.

```
btnAdd.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        num1 = Double.parseDouble(firstNumber.getText().toString());
        num2 = Double.parseDouble(secondNumber.getText().toString());
        sum = num1 + num2;
        addResult.setText(Double.toString(sum));
    }
});
```

Save the above changes and run the application. You should have the addition app running. Input the two numbers and you should be able to view the result in the Text View.

Output:**Conclusion:**

Successfully developed an Android Application using Android Studio.

References:

- <http://codehandbook.org/simple-android-application/>
- <https://www.tutorialspoint.com/android/>

Assignment 8

Title: Design and implement Web Service Manager for following service:

1. Web service StringTool: The multistring operation that returns the input string repeated as a sequence of a specifiable length. The operation has 2 input parameters: a string, S and an integer, x that specifies how often the input string should be repeated as it is in the output string, O along with other symbols of the string in between. The output parameter is output string, O that contains the at least x occurrences of S.
2. Matrix Inversion: Service to accept a matrix in the form of vectors of its rows and return its inverse.

- **Web service StringTool**

```

using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service" in code, svc and config file together.
public class Service : IService
{
    System.Text.StringBuilder sb = new System.Text.StringBuilder();

    public string SampleMethod(string Name, int Count)
    {
        for (int i = 1; i <= Count; i++)
        {
            sb.Append(Name).Append(" |");
        }
        return sb.ToString();
    }
}

Output
Show output from: Build
Validating web site
Building directory '/App_Code/'.
Building directory ''.

Validation Complete
=====
Rebuild All: 1 succeeded, 0 failed, 0 skipped =====

Web Publish Activity: Output

```

Rebuild All succeeded

```

using System.Threading.Tasks;

namespace ConsoleApplication2016MTECSCO020
{
    class Program
    {
        static void Main(string[] args)
        {

            ServiceReference1.ServiceClient objService = new ConsoleApplication2016MTECSCO020.ServiceReference1.ServiceClient();

            Console.WriteLine("Please Enter String: ");
            string Message = Console.ReadLine();
            Console.WriteLine("How many times to repeat: ");
            int Repeat = (int)Convert.ToDouble(Console.ReadLine());

            string output = objService.SampleMethod(Message,Repeat);

            Console.WriteLine(output);
            Console.ReadLine();
        }
    }
}

Output
Show output from: Debug
The thread 'vshost.RunParkingWindow' (0x2fb8) has exited with code 0 (0x0).
The thread '' (0x1780) has exited with code 0 (0x0).
The program '[964] ConsoleApplication2016MTECSCO020.vhost.exe: Program Trace' has exited with code 0 (0x0).
The program '[964] ConsoleApplication2016MTECSCO020.vhost.exe: Managed (v4.0.30319)' has exited with code 0 (0x0).

Item(s) Saved

```

Ln 26 Col 28 Ch 28 INS

Output of Web Service String Tool:

```
file:///C:/Users/CSE/Documents/Visual Studio 2012/Projects/Count Return String/ConsoleApplication2016MTECSCO020/ConsoleApplication2016MTEC...
Please Enter String:
Hello
How many times to repeat:
5
Hello Hello Hello Hello Hello
```

• Matrix Inversion:

localhost_10451 - Microsoft Visual Studio (Administrator)

FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP

Service.cs IServiceProvider Matrix.cs Complex.cs Service.svc

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService" in both code and config file together.
[ServiceContract]
public interface IService
{
    [OperationContract]
    //string SampleMethod(string Name);
    string IM(double[] arr, int Dim);
}
```

Solution Explorer

- Solution 'localhost_10451' (1 proj)
 - Matrix_service
 - App_Code
 - Complex.cs
 - IService.cs
 - Matrix.cs
 - Service.cs
 - App_Data
 - Service.svc
 - Web.config
 - Web.Debug.config
 - website.publishproj

Properties

localhost_10451 - Microsoft Visual Studio (Administrator)

FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP

Service.cs IServiceProvider Matrix.cs Complex.cs Service.svc

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using CSML;

public class Service : IService
{
    public string IM(double[] arr, int Dim)
    {
        double[,] mat = new double[Dim, Dim];
        for (int i = 0, k = 0; i < Dim; i++)
        {
            for (int j = 0; j < Dim; j++)
            {
                mat[i, j] = arr[k];
                k++;
            }
        }
        double[,] new_mat = new double[Dim, Dim];
        Matrix m = new Matrix(mat);
        m = m.Inverse();
        string buf = m.ToString();
        return buf;
    }
}
```

Solution Explorer

- Solution 'localhost_10451' (1 proj)
 - Matrix_service
 - App_Code
 - Complex.cs
 - IService.cs
 - Matrix.cs
 - Service.cs
 - App_Data
 - Service.svc
 - Web.config
 - Web.Debug.config
 - website.publishproj

Properties

```
matrixApp - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW
Program.cs
matrixApp.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using matrixApp.ServiceReference1;
namespace matrixApp
{
    class Program
    {
        static void Main(string[] args)
        {
            ServiceReference1.ServiceClient objService = new ServiceClient();
            Console.WriteLine("Enter Matrix Dimension");
            int Dim = Convert.ToInt32(Console.ReadLine());
            int MD = Dim * Dim;
            double[] arr = new double[MD];
            Random rnum = new Random();

            for (int i = 0; i < arr.Length; i++)
            {
                arr[i] = rnum.Next(1, 50);
            }
            double[,] mat = new double[Dim, Dim];
            System.Console.WriteLine("\n\n\nMatrix Input:");
            for (int i = 0, k = 0; i < Dim; i++)
            {
                for (int j = 0; j < Dim; j++)
                {
                    mat[i, j] = arr[k];
                    k++;
                }
            }
            for (int i = 0; i < Dim; i++)
            {
                for (int j = 0; j < Dim; j++)
                {
                    System.Console.Write(mat[i, j] + " ");
                }
                System.Console.Write("\n");
            }

            string sarr;
            sarr = objService.IM(arr, Dim);
            System.Console.WriteLine("\n\n\nInverse Matrix:\n");
            System.Console.WriteLine(sarr);
            Console.ReadLine();
        }
    }
}
```

Output of Matrix Inversion:

```
Enter Matrix Dimension
6

Matrix Input:
12 34 6 7 22 42
15 46 38 3 48 37
31 32 24 2 5 29
42 23 17 43 45 24
48 36 19 18 4 31
35 8 33 31 8 16

Inverse Matrix:

-0.021 -0.046 0.145 0.056 -0.079 -0.032
-0.011 0.073 -0.202 -0.066 0.165 0.003
-0.009 0.019 -0.011 -0.020 0.000 0.028
0.017 0.041 -0.178 -0.050 0.112 0.039
-0.010 -0.016 0.084 0.045 -0.068 -0.025
0.042 -0.047 0.109 0.024 -0.094 0.009
```

Conclusion:

Designed and implemented Web Service Manager for required services.

References:

<http://www.aspdotnet-suresh.com/2011/06/introduction-to-wcf-wcf-tutorial-wcf.html>

Assignment 9

Title: Study of Eucalyptus cloud with regards to its components and terminologies used.

Theory:

- **Cloud** - A federated set of physical machines that offer computing resources through virtual machines, provisioned and recollected dynamically.
- **Cloud Controller (CLC)** - Eucalyptus component that provides the web UI (an https server on port 8443), and implements the Amazon EC2 API. There should be only one Cloud Controller in an installation of UEC. This service is provided by the Ubuntu *eucalyptus-cloud* package.
- **Cluster** - A collection of nodes, associated with a Cluster Controller. There can be more than one Cluster in an installation of UEC. Clusters are sometimes physically separate sets of nodes. (e.g. floor1, floor2, floor2).
- **Cluster Controller (CC)** - Eucalyptus component that manages collections of node resources. This service is provided by the Ubuntu *eucalyptus-cc* package.
- **EBS** - Elastic Block Storage. <http://aws.amazon.com/ebs/>
- **EC2** - Elastic Compute Cloud. Amazon's pay-by-the-hour, pay-by-the-gigabyte public cloud computing offering.
- **EKI** - Eucalyptus Kernel Image.
- **EMI** - Eucalyptus Machine Image.
- **ERI** - Eucalyptus Ramdisk Image.
- **Eucalyptus** - Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems. An open source project originally from the University of California at Santa Barbara, now supported by Eucalyptus Systems, a Canonical Partner.
- **Front-end** - Physical machine hosting one (or more) of the high level Eucalyptus components (cloud, walrus, storage controller, cluster controller).
- **Node** - A node is a physical machine that's capable of running virtual machines, running a node controller. Within Ubuntu, this generally means that the CPU has VT extensions, and can run the KVM hypervisor.
- **Node Controller (NC)** - Eucalyptus component that runs on nodes which host the virtual machines that comprise the cloud. This service is provided by the Ubuntu package *eucalyptus-nc*.
- **S3** - Simple Storage Service. Amazon's pay-by-the-gigabyte persistent storage solution for EC2. <http://aws.amazon.com/s3/>
- **Storage Controller (SC)** - Eucalyptus component that manages dynamic block storage services (EBS). Each 'cluster' in a Eucalyptus installation can have its own Storage Controller. This component is provided by the '*eucalyptus-sc*' package.
- **UEC** - Ubuntu Enterprise Cloud. Ubuntu's cloud computing solution, based on Eucalyptus.
- **VM** - Virtual Machine.
- **VT** - Virtualization Technology. An optional feature of some modern CPUs, allowing for accelerated virtual machine hosting.
- **Walrus** - Eucalyptus component that implements the Amazon S3 API, used for storing VM images and user storage using S3 bucket put/get abstractions.

Steps:

STEP 1: Prerequisites

To deploy a minimal cloud infrastructure, you'll need at least two dedicated systems:

- a front end
- one or more node(s)

Front End

Use the following table for a system that will run one or more of:

- the cloud controller (clc)
- the cluster controller (cc)
- walrus (the S3-like storage service)
- the storage controller (sc)

Node(s)

The other system(s) are nodes, which will run:

- the node controller (nc)

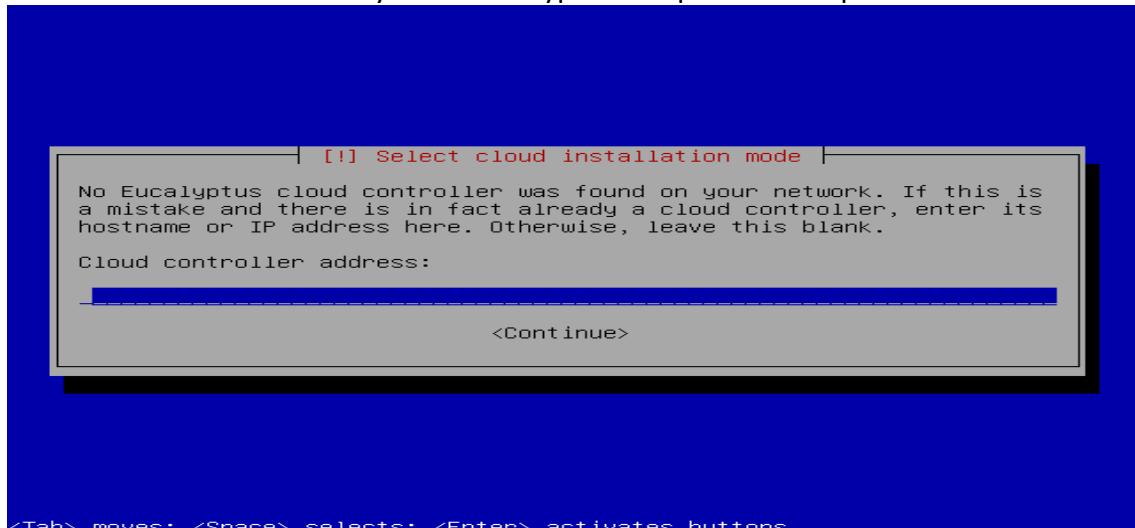
These systems will actually run the instances. You will need one or more systems.

STEP 2: Install the Cloud/Cluster/Storage/Walrus Front End Server

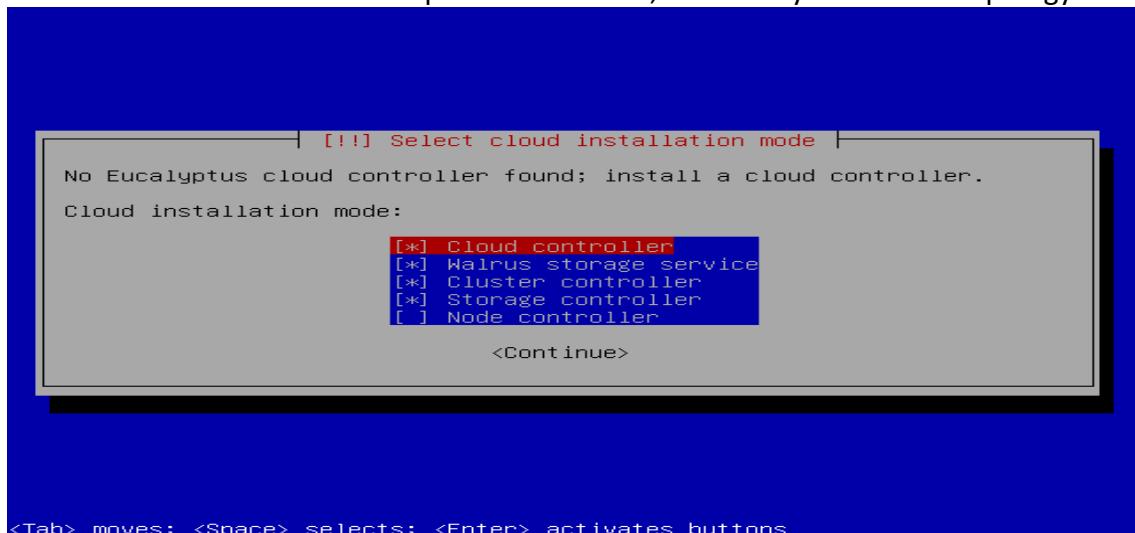
1. Download the 10.04 Server ISO
2. When you boot, select “Install Ubuntu Enterprise Cloud”.



3. The installer will detect if any other Eucalyptus components are present.

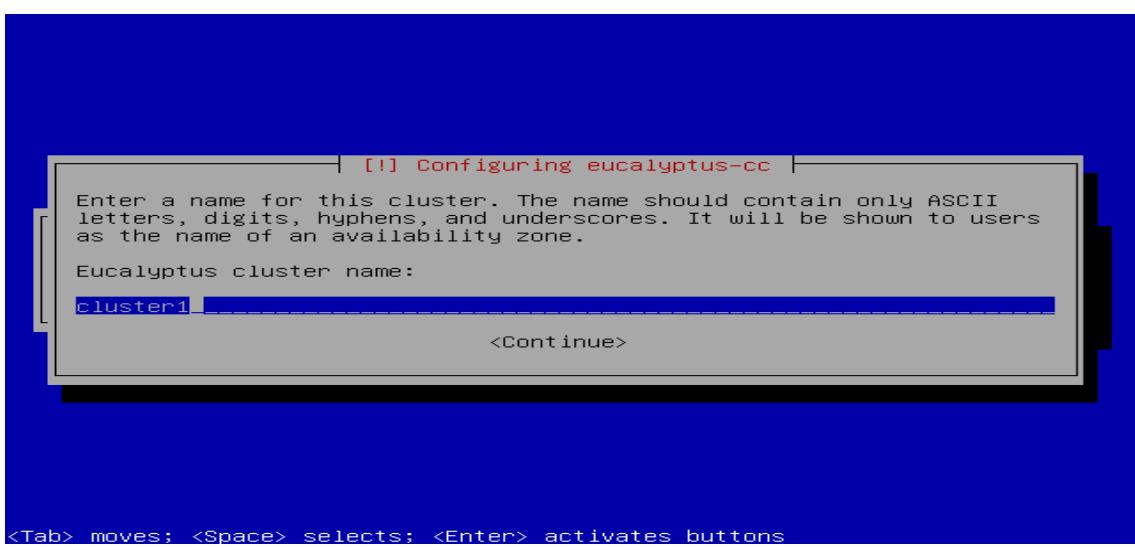


4. You can then choose which components to install, based on your chosen topology.



5. It will ask two other cloud-specific questions during the course of the install:

1. Name of your cluster, e.g. *cluster1*



2. A range of public IP addresses on the LAN that the cloud can allocate to

The screenshot shows a terminal window with the title '[!] Configuring eucalyptus-cc'. The text inside the window reads:

```
Eucalyptus requires a pool of IP addresses that can be dynamically assigned as the "public" IPs of virtual machines. These IPs must be unused within their Class C subnet, this system must have an interface configured with an address on this subnet, and your prospective users must be able to connect to these IPs from wherever they run the client tools.

Please specify one or more ranges of IP addresses, e.g.:
192.168.1.100-192.168.1.199
or
192.168.2.50-192.168.2.99 192.168.2.150-192.168.2.199

You may leave this blank if you have no IP addresses available, BUT you and your users must then request the private addressing scheme when starting a virtual machine instance. For ec2-run-instances and euca-run-instances, this is done with the option "--addressing private".
```

At the bottom of the window, there is a blue bar with the text '<Continue>' and a message at the bottom: '<Tab> moves; <Space> selects; <Enter> activates buttons'.

STEP 3: Install the Node Controller(s)

The node controller install is even simpler. Just make sure that you are connected to the network on which the cloud/cluster controller is already running.

1. Boot from the same ISO on the node(s)
2. Select “Install Ubuntu Enterprise Cloud”
3. It should detect the Cluster and preselect “Node” install for you
4. Confirm the partitioning scheme
5. The rest of the installation should proceed uninterrupted; complete the installation and reboot the node

Note: You can login in the node by using the username and password defined for the cloud controller host.

STEP 4: Obtain Credentials

After installing and booting the Cloud Controller, users of the cloud will need to retrieve their *credentials*. This can be done either through a web browser, or at the command line.

From a Web Browser

1. From your web browser (either remotely or on your Ubuntu server) access the following URL:
<https://<cloud-controller-ip-address>:8443/>
2. Use username 'admin' and password 'admin' for the first time login (you will be prompted to change your password).
3. Then follow the on-screen instructions to update the admin password and email address.
4. Once the first time configuration process is completed, click the 'credentials' tab located in the top-left portion of the screen.

5. Click the 'Download Credentials' button to get your certificates
6. Save them to `~/.euca`
7. Unzip the downloaded zipfile into a safe location (`~/.euca`)
8. `unzip -d ~/.euca mycreds.zip`

STEP 5: Install an image from the store

The following is by far the simplest way to install an image. However, advanced users may be interested in learning how to Bundle their own image.

The simplest way to add an image to UEC is to install it from the Image Store on the UEC web interface.

1. Access the web interface at the following URL (Make sure you specify https):
`https://<cloud-controller-ip-address>:8443/`
2. Enter your login and password (if requested, as you may still be logged in from earlier)
3. Click on the Store tab



The screenshot shows the UEC web interface with the 'Store' tab selected. The main content area is titled 'All Images' and displays two entries:

- Ubuntu 9.10 RC - Karmic Koala (amd64)**
Image version: 20091022
Ubuntu 9.10 Release Candidate image for amd64.
Read more... [How to run?](#)
- Ubuntu 9.10 RC - Karmic Koala (i386)**
Image version: 20091022
Ubuntu 9.10 Release Candidate image for i386.
Read more... [Install](#)

4. Browse available images
5. Click on install for the image you want

Once the image has been downloaded and installed, you can click on "How to run?" that will be displayed below the image button to view the command to execute to instantiate (start) this image. The image will also appear on the list given on the Image tab.

ID	Name	Kernel	Ramdisk	State	Actions
eri-0B4E116D	image-store-1256379456/ramdisk.manifest.xml			available	Delete
emi-3DFF123D	i-20091021151206/karmic-uec-amd64.img.manifest.xml	eki-61F31745	eri-44CC16C0	available	Delete
eki-81F31745	k-20091021151206/karmic-uec-amd64-vmlinux-virtual.manifest.xml			available	Delete
emi-E05B1075	image-store-1256379456/image.manifest.xml	eki-F6C110FD	eri-0B4E116D	available	Delete
eri-44CC16C0	r-20091021151206/karmic-uec-amd64-initrd-virtual.manifest.xml			available	Delete
eki-F6C110FD	image-store-1256379456/kernel.manifest.xml			available	Delete
emi-95800F86	nydemo/mediawiki.img.manifest.xml	eki-61F31745	eri-44cc16c0	available	Delete

STEP 6: Run an Image

There are multiple ways to instantiate an image in UEC:

- Use the command line
- Use one of the UEC compatible management tools such as Landscape
- Use the ElasticFox extension to Firefox

Conclusion:

Studied and demonstrated Eucalyptus cloud with regards to its components using Ubuntu Enterprise Cloud OS v10.04 Server.

References:

<https://help.ubuntu.com/community/UEC/CDInstall>

Assignment 10

Title: Install and configure latest version of VMWare ESXi. Demonstrate the creation of virtual machine and host OS access.

Theory:

VMware ESXi is based on hypervisor architecture that runs directly on top of a hardware as shown below.



Steps:

1. Download ESXi server

Get the software from the VMware ESXi download page.

Following are the various download options available. Select “ESXi 4.0 Update 1 Installable (CD ISO) Binary (.iso)” and burn a CD.

- ESXi 4.0 Update 1 Installable (CD ISO)
- Upgrade package from ESXi Server 3.5 to ESXi Server 4.0 Update 1
- Upgrade package from ESXi Server 4.0 to ESXi Server 4.0 Update 1
- VMware vSphere Client and Host Update Utility

2. VMware VMvisor Boot Menu

Once you insert the ESXi CD and reboot the server, it will display a boot menu with an option to launch “ESXi Installer”.

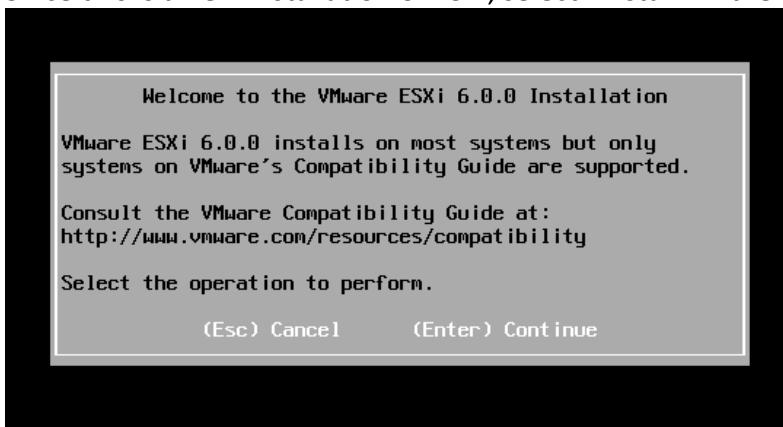
3. VMware ESXi Installer Loading

While the installer is loading all the necessary modules, it will display the server configuration information at the top as shown below.



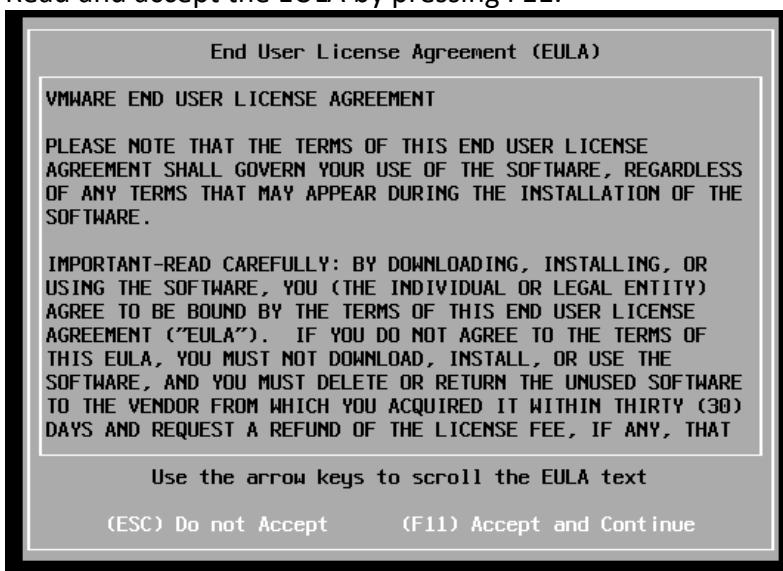
4. New ESXi Install

Since this is a new installation of ESXi, select “Install” in the following screen.



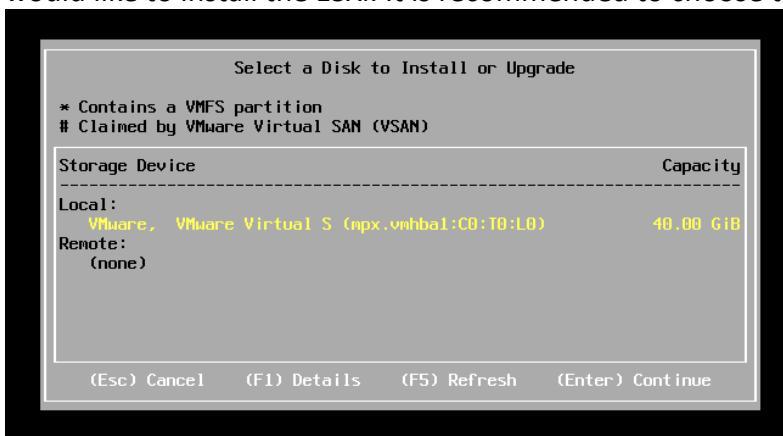
5. Accept VMware EULA

Read and accept the EULA by pressing F11.



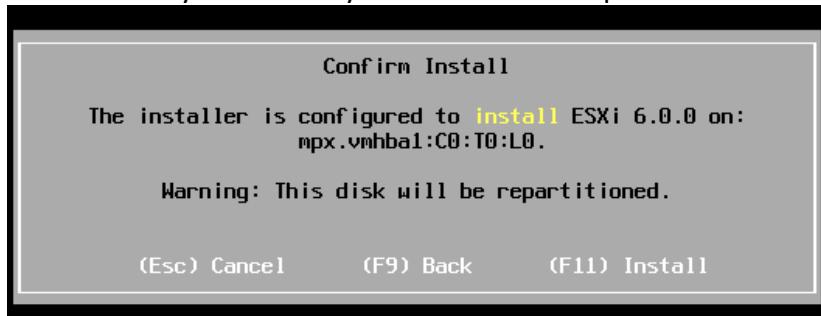
6. Select a Disk to Install VMware ESXi

VMware ESXi 4.0.0 Installer will display all available disk groups. Choose the Disk where you would like to install the ESXi. It is recommended to choose the Disk0.



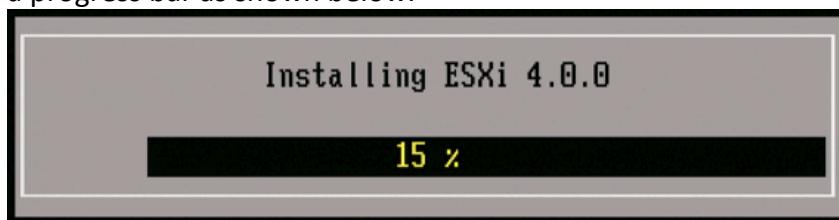
7. Confirm ESXi Installation

Confirm that you are ready to start the install process.



8. Installation in Progress

The installation process takes few minutes. While the ESXi is getting installed, it will display a progress bar as shown below.



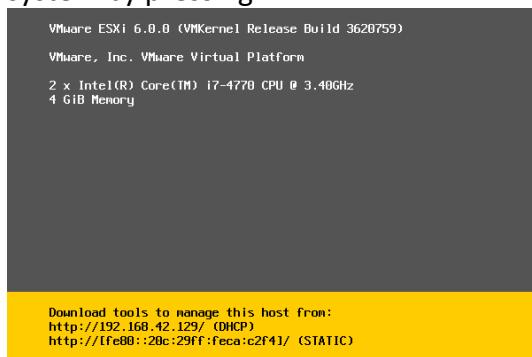
9. ESXi Installation Complete

You will get the following installation completed message that will prompt you to reboot the server.



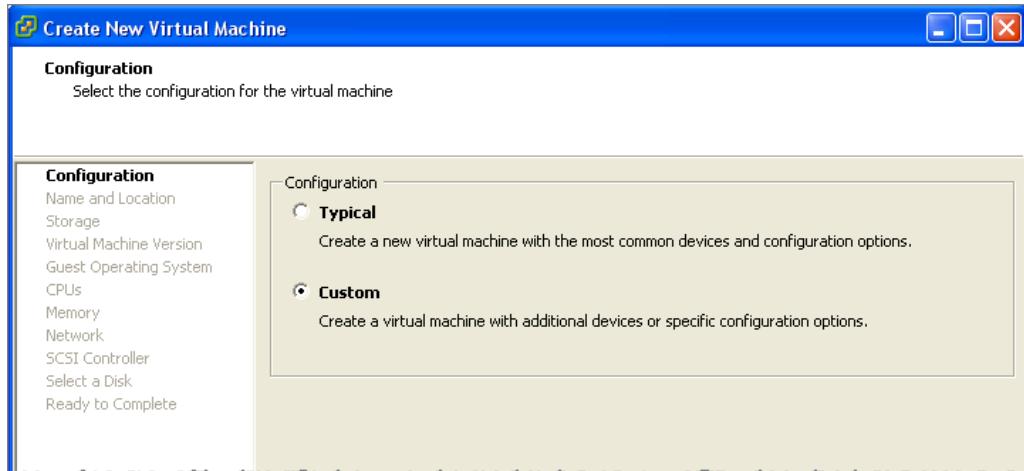
10. ESXi Initial Screen

After the ESXi is installed, you'll get the following screen where you can configure the system by pressing F2.

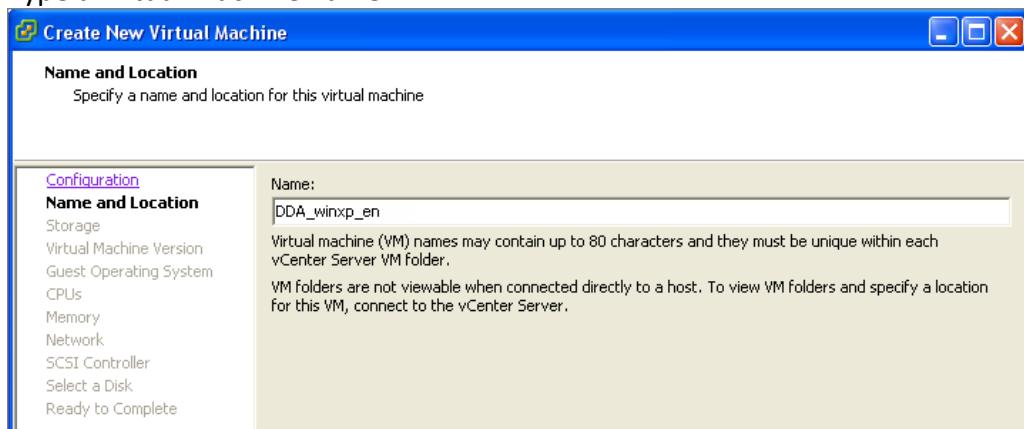


Creating a New Virtual Machine on the VMware ESXi Server

1. Log on to the VMware ESXi server using vSphere client (see Task 6: Using vSphere Client to Log On to the VMware ESXi Server).
2. Press Ctrl+N to start creating a new virtual machine.
3. Select Custom and then click Next.



4. Type a virtual machine name.



The name must:

- Be prefixed with **DDA_**.
- Not exceed 25 characters.
- Not contain special characters, such as:
\$; ' " {
- Not end with an underscore and a number
- Not contain the letters "vmx" (in this order) anywhere in the name

Examples of valid names:

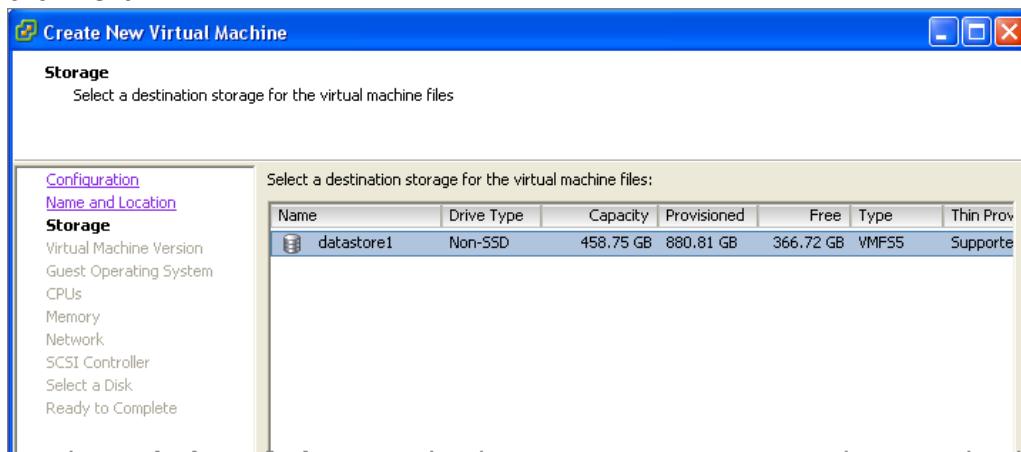
- **DDA_winxp_en**
- **DDA_win7**

Examples of invalid names:

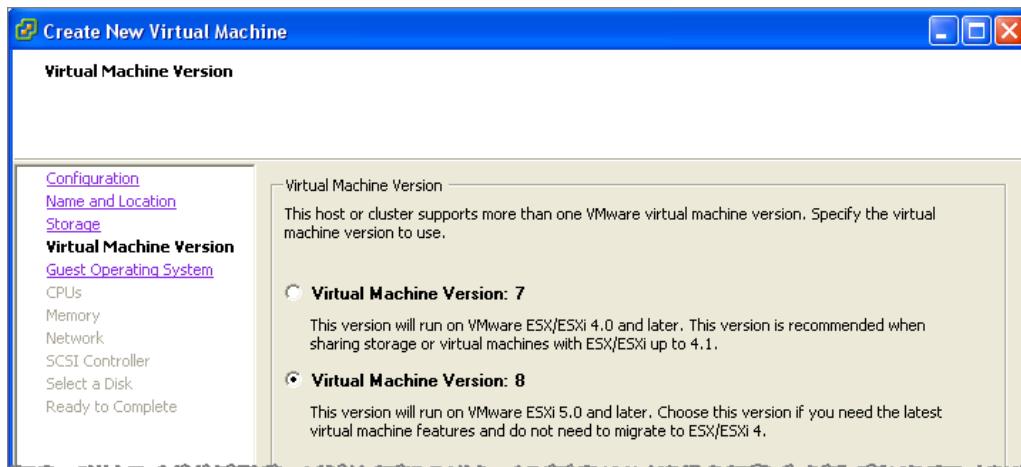
- **"DDAWin7\$"**
- **DDA_winXP_1**
- **DDA_winxpvmx**
- **DDA_vmxwinxp**

Click Next.

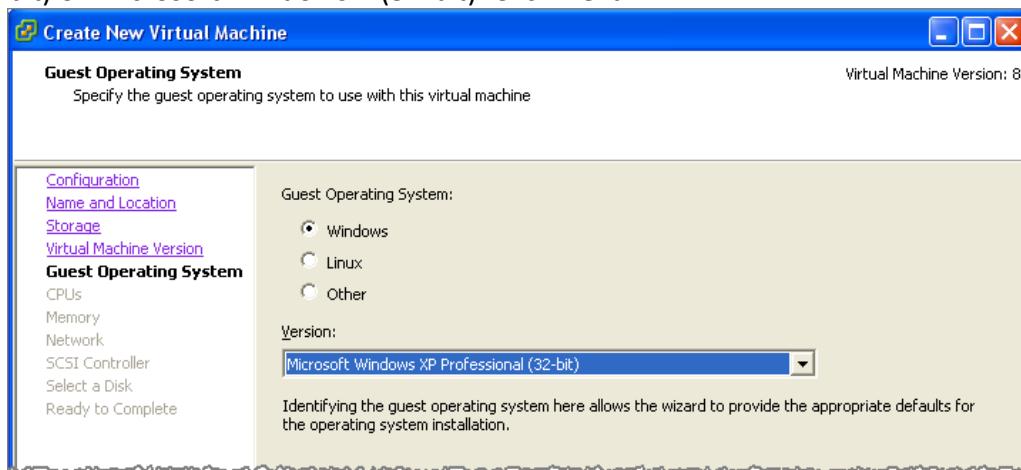
5. Select the destination storage (datastore) for the virtual machine and then click Next.



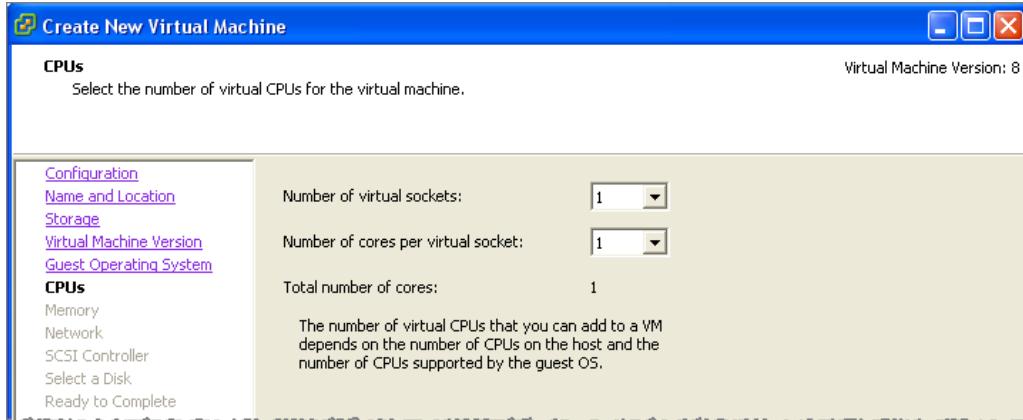
6. Select Virtual Machine Version: 8 and then click Next.



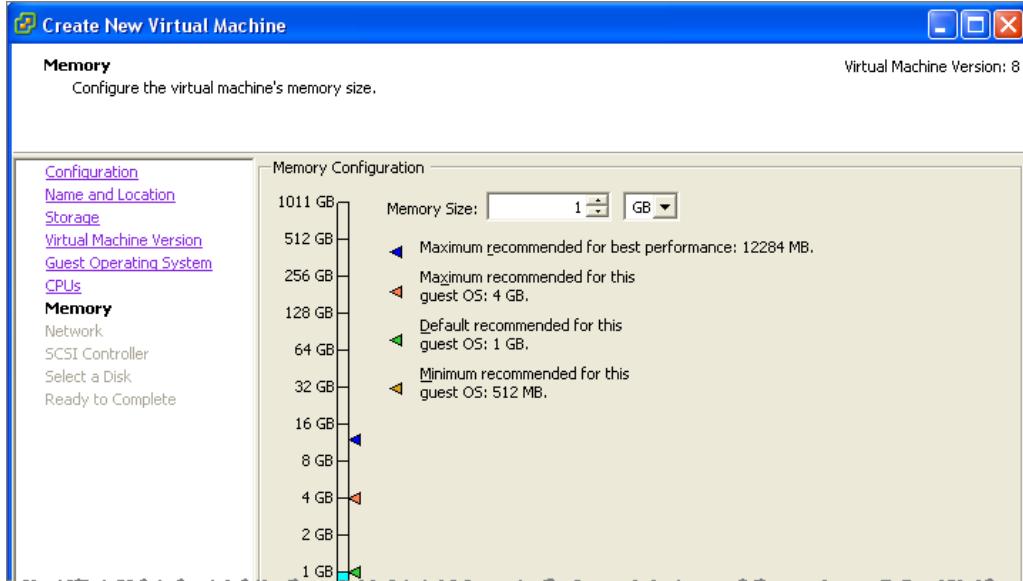
7. Select Windows and then either Microsoft Windows XP Professional (32-bit) or Microsoft Windows 7 (32-bit). Click Next.



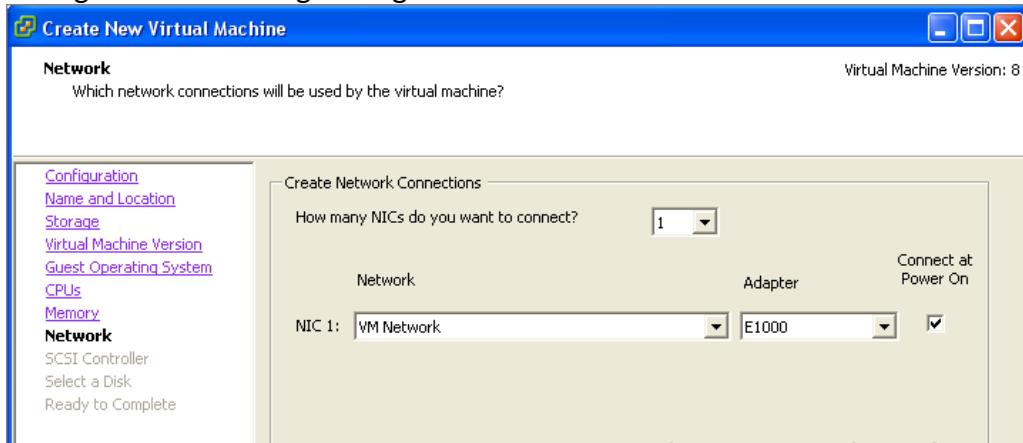
8. Accept the default values of 1 virtual socket and 1 core. Click Next.



9. Allocate 512MB of memory for Windows XP or 1GB for Windows 7. Click Next.



10. Configure the following settings:

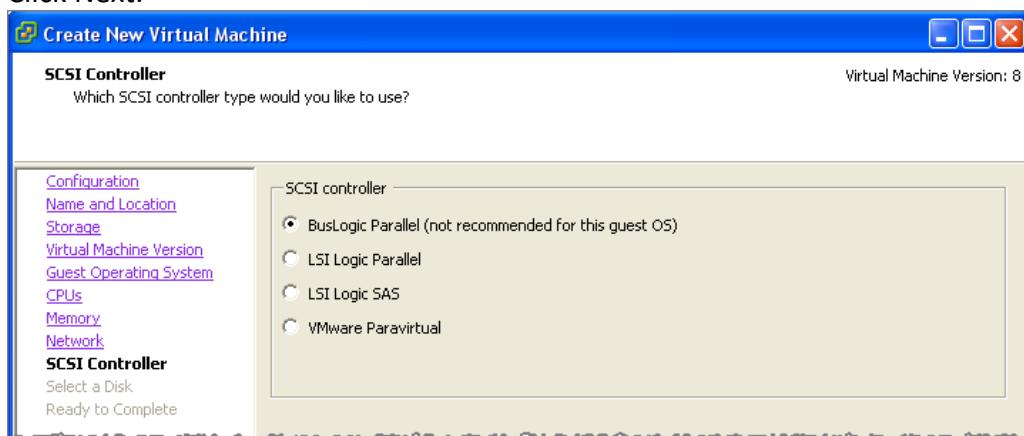
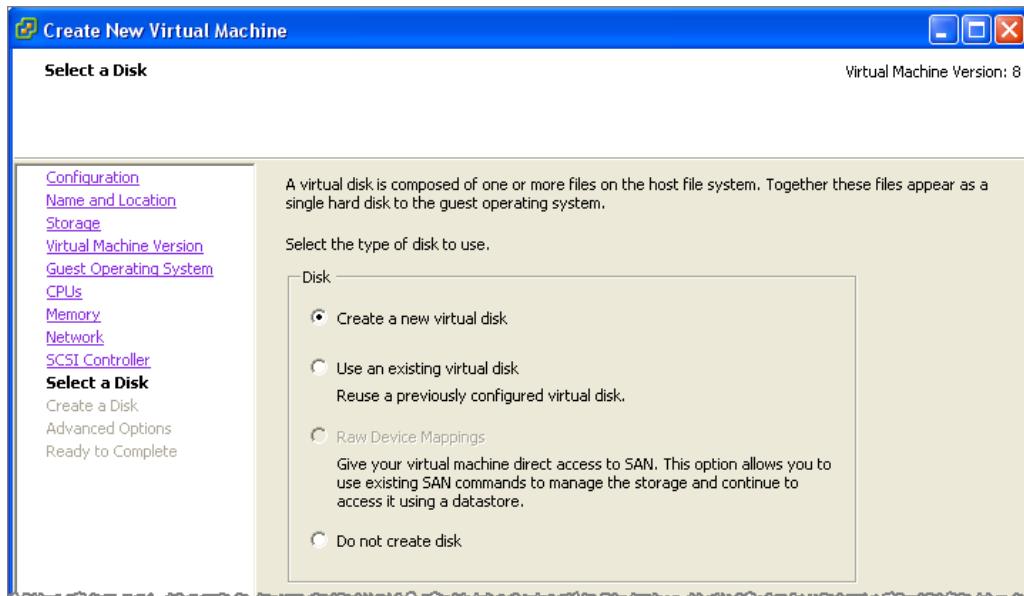
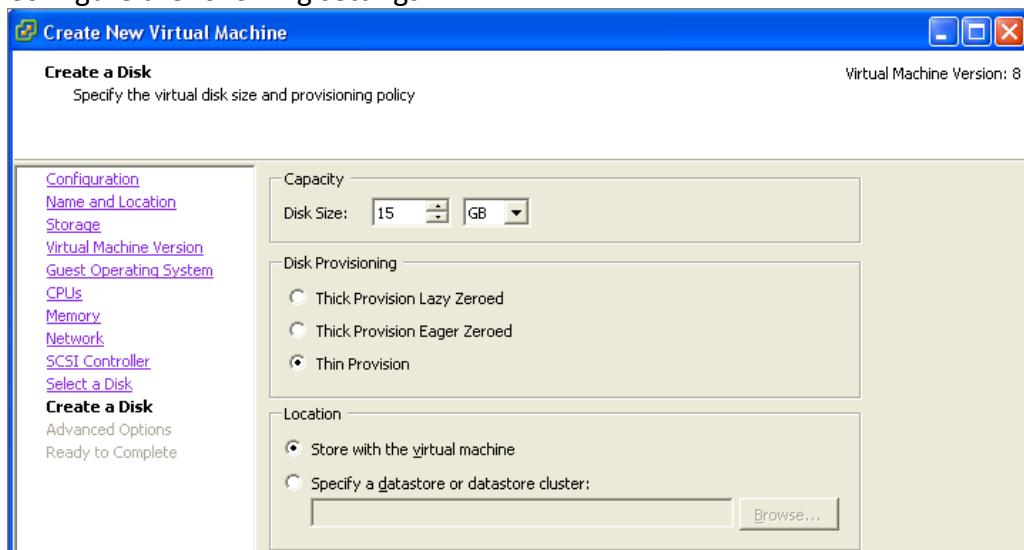


- o How many NICs do you want to connect?: 1
- o Network: VM Network
- o Adapter: E1000
- o Connect at Power On: Enabled

Click Next.

11. Select BusLogic Parallel for Windows XP or LSI Logic Parallel for Windows 7.

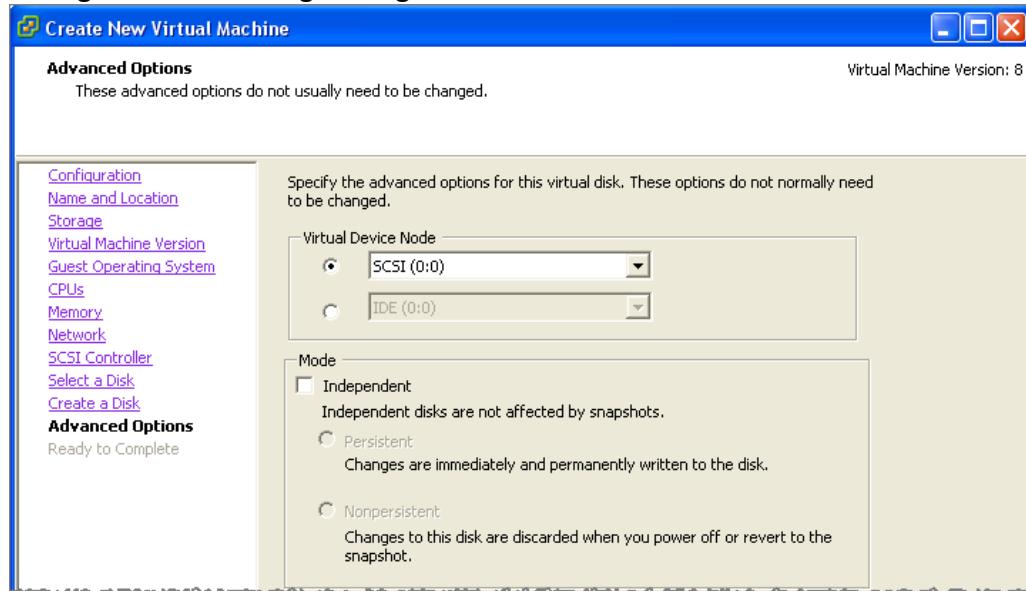
Click Next.

**12. Select Create a new virtual disk and then click Next.****13. Configure the following settings:**

- Capacity: 20GB for Windows XP, 30GB for Windows 7
- Disk Provisioning: Thin Provision
- Location: Store with the virtual machine

Click Next.

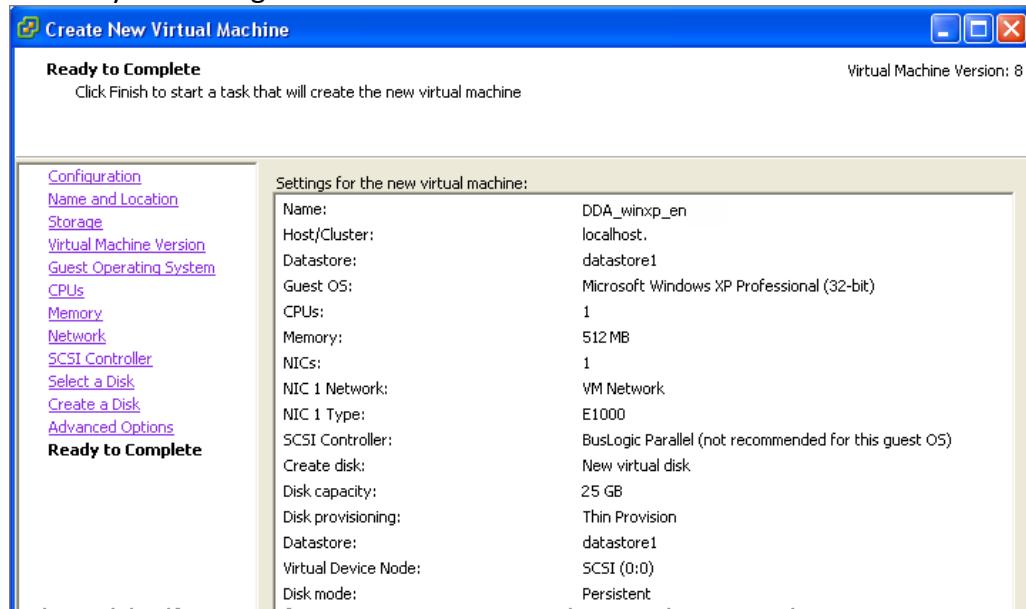
14. Configure the following settings:



- Virtual Device Node: SCSI (0:0)
- Mode: Disable Independent

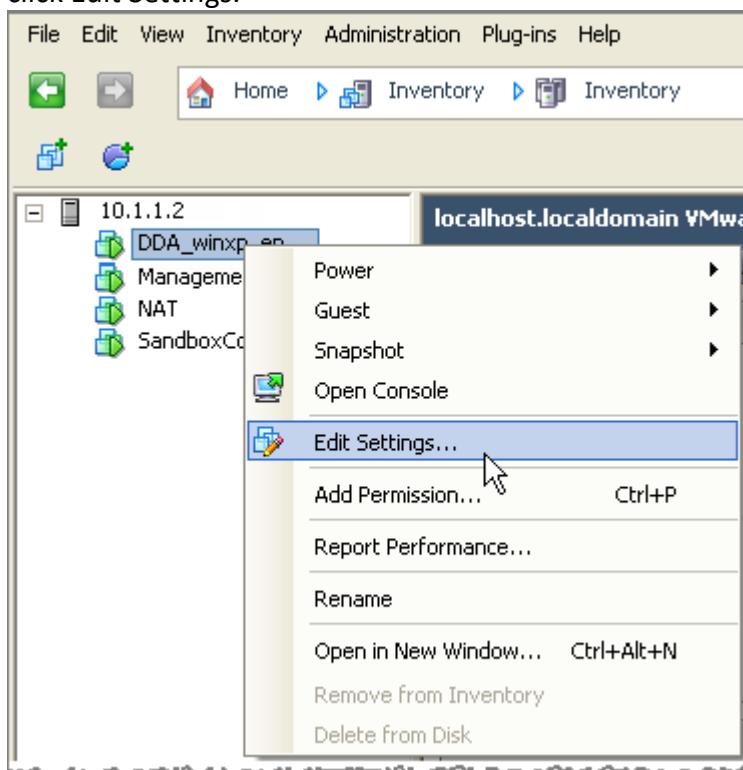
Click Next.

15. Review your settings and then click Finish.

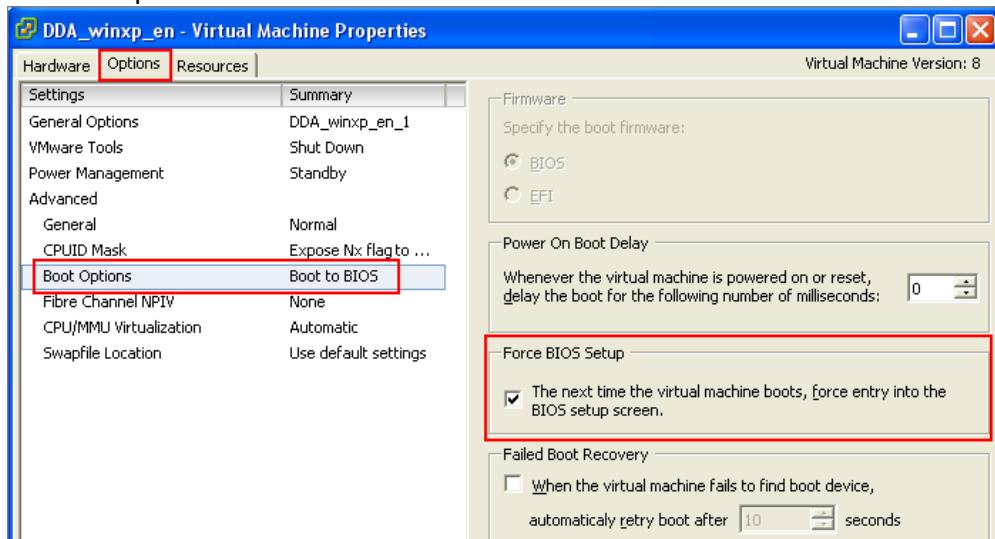


The VMware ESXi server starts to create the virtual machine.

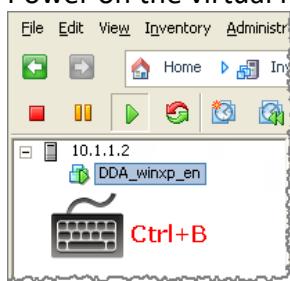
16. When the virtual machine has been created, right-click it in the inventory and click Edit Settings.



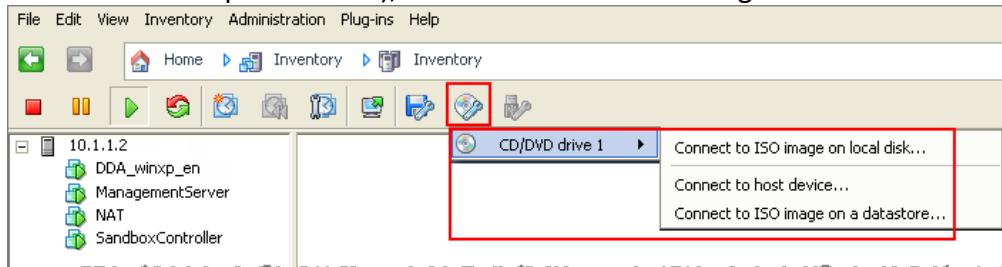
17. Click the Options tab, select Boot Options, and then select the option under Force BIOS Setup. Click OK.



18. Power on the virtual machine by selecting it in the inventory and pressing Ctrl+B.

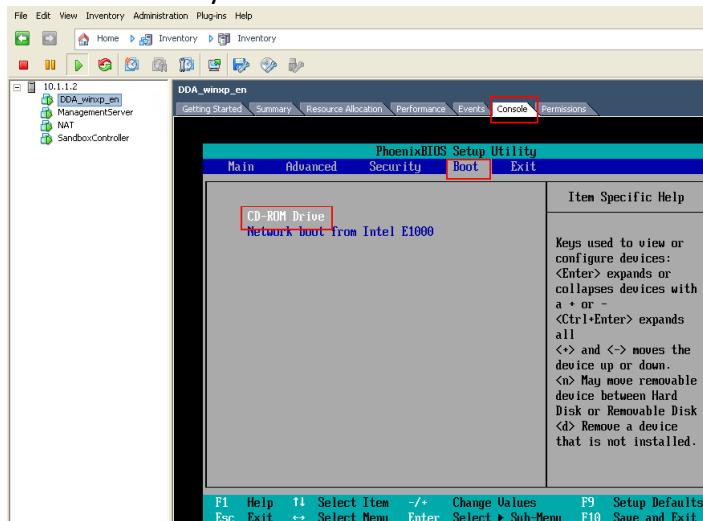


19. On the toolbar on top of the screen, click the CD icon, mouseover CD/DVD drive 1, and then select the option according to the location of the Windows operating system installer. For example, if the installer is an ISO file on the local machine (the machine that hosts the vSphere client), select Connect to ISO image on local disk.



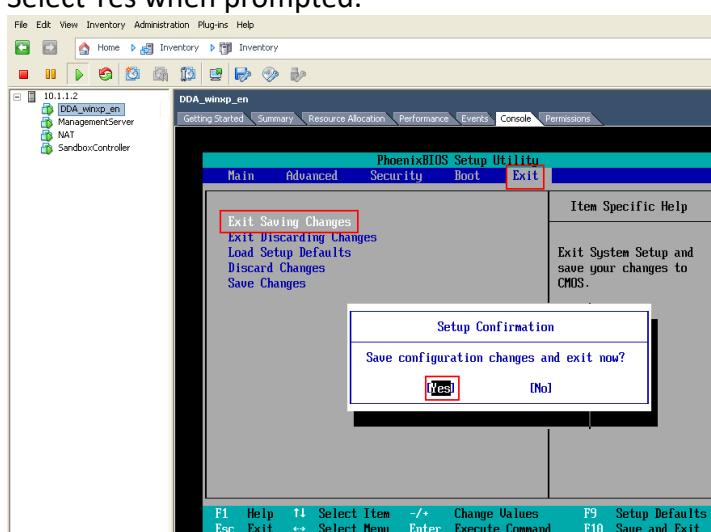
20. Click the Console tab to display the BIOS Setup screen.

- Scroll to the Boot tab.
- Scroll down to select CD-ROM Drive.
- If CD-ROM Drive is not on top of the list, move it to the top by pressing the + key one or several times.

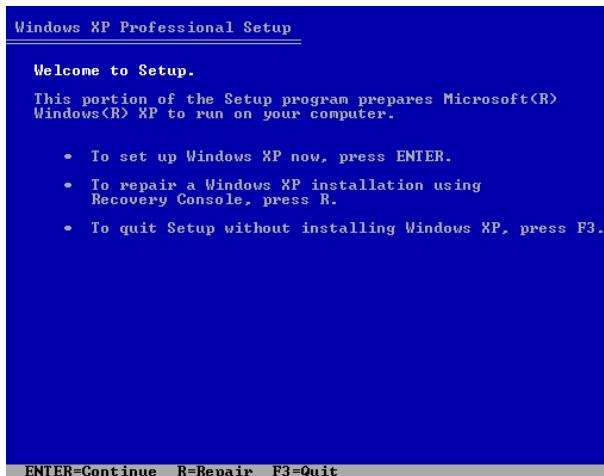


21. Scroll to the Exit tab and then scroll down to select Exit Saving Changes.

Select Yes when prompted.



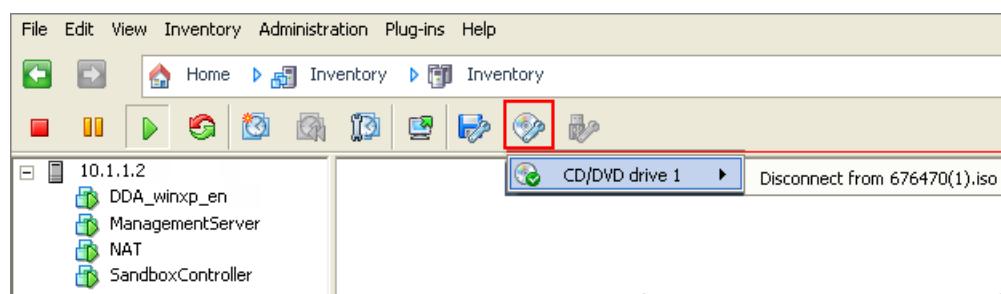
The virtual machine boots from the installer, initiating the installation of the operating system. The screen that displays depend on the operating system you want to install. The following screen is for Windows XP.



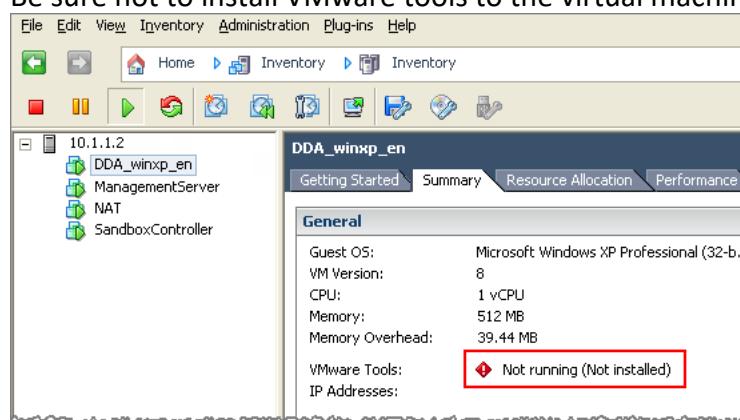
22. Follow the on-screen instructions to complete the installation.

23. When the installation is complete:

- . Disconnect the virtual machine from the CD/DVD drive.



Be sure not to install VMware tools to the virtual machine.



Conclusion:

Installed and configured latest version of VMWare ESXi. Also, demonstrated the creation of virtual machine and host OS access.

References:

<https://pubs.vmware.com/vsphere-51/index.jsp#com.vmware.vsphere.solutions.doc/GUID-5C01A62A-1048-4388-8101-443306CD5B2E.html>

Assignment 11

Title: Study, install and configure following VMWare products:

1. VMware workstation
2. vCloud Suite Standard
3. vSphere Enterprise
4. vCenter Server edition
5. vCloud Director

Theory:

1. VMware workstation

VMware Workstation Pro and VMware Workstation Player are the industry standard for running multiple operating systems as virtual machines on a single PC. Thousands of IT professionals, developers and businesses use Workstation Pro and Workstation Player to be more agile, more productive and more secure every day. With over 15 years of virtualization leadership, millions of satisfied customers, and more than 50 awards, VMware provides the most stable and secure desktop virtualization platform in the industry.

What VMware workstation does

- Powerful Desktop Virtualization - Workstation products allows users to run multiple operating systems including Linux, Windows and more, as virtual machines on a single PC. Users can replicate server, desktop and tablet environments on a virtual machine, to run applications simultaneously across operating systems without rebooting. Workstation also provides an isolated and safe environment to evaluate new operating systems like Windows 10 and test software applications, patches and reference architectures.
- Dev and Test Capabilities for any Platform - Workstation products make it easy to test almost any operating system and application, all from a local PC. Build apps for Windows 10, test compatibility with any browser, or deploy Android-x86 to check mobile behaviour without needing a mobile device.
- Connection to vSphere - IT pros use Workstation products to securely connect with vSphere, ESXi or other Workstation servers to manage virtual machines and physical hosts. A common hypervisor platform maximizes agility and productivity by enabling easy transfer of virtual machines to and from your local PC.
- Locally Controlled Corporate Desktops - Isolate corporate desktops from BYO devices by disabling copy-and-paste, drag-and-drop, shared folders and access to USB devices. Run restricted virtual machines that are encrypted and password-protected, ensuring that only authorized users can interact with corporate data.
- Secure and Isolated Environments - Run a secure second desktop with different privacy settings, tools and networking configurations, or use forensic tools to investigate operating system vulnerabilities. Workstation provides one of the most secure hypervisor in the industry, and support for security focused operating systems like Kali Linux.

2. vCloud Suite Standard

VMware vCloud Suite is an integrated offering that brings together VMware's industry-leading vSphere hypervisor and VMware vRealize Suite multi-vendor hybrid cloud management platform. VMware's new portable licensing units allow vCloud Suite to build and manage both vSphere-based private clouds as well as multi-vendor hybrid clouds.

What vCloud Suite does

- Data Center Virtualization - Leverage the world's leading virtualization platform as the foundation for multiple data center initiatives. Consolidate servers and data centers. Improve application availability and performance. Address application requirements for scale up or scale out.
- IT Automating IT - Rapidly provision infrastructure and application resources across development, test, QA and production. Effectively manage these resources from initial provisioning through retirement in order to maximize both capital and operational spending. Understand the cost of infrastructure and the consumption of resources by end users.
- Intelligent Operations - Deliver proactive management of performance, availability and capacity across all data center domains including applications, compute, network and storage. Build the foundation for successful developer agility and infrastructure modernization initiatives.
- DevOps-Ready IT - Deliver a complete application stack across vSphere and hybrid cloud resources and support developer access from both API and GUI.

3. vSphere Enterprise

vSphere helps you get the best performance, availability, and efficiency from your infrastructure and applications. It's the ideal foundation for any cloud environment. vSphere with Operations Management takes virtualization to the next level with intelligent operations management. It provides you better insights, resulting in improved performance and availability.

What vSphere Enterprise does

- Scale Enhancements – New configuration maximums to support even the largest app environments
- VMware vCenter Server Appliance – The single control center and core building block
- vCenter Server Appliance Tool – Single step migration and upgrade of existing vCenter Server to vCenter Server Appliance
- vSphere Client – HTML5-based GUI that ensures fast performance and cross-platform compatibility
- Security-at-Scale – Policy-driven security that makes securing infrastructure operationally simple
- Encryption – VM-level encryption protects unauthorized data access both at-rest and in-motion
- Audit-quality logging – Enhanced logging that provides forensic information about user actions

4. vCenter Server

Centralized visibility, proactive management and extensibility for VMware vSphere from a single console VMware vCenter Server provides a centralized platform for managing your VMware vSphere environments, so you can automate and deliver a virtual infrastructure with confidence.

What vCenter Server does

- Simple Deployment - Quickly and easily deploy vCenter Server using host profiles or a Linux-based virtual appliance.
- Proactive Optimization - Allocate and optimize resources for maximum efficiency with vCenter Server.
- Centralized Control and Visibility - Administer your entire vSphere infrastructure from a single location.
- Management - Use powerful tools to simplify management and extend your control, now including vRealize Log Insight.

5. vCloud Director

VMware vCloud Director enables service providers to build differentiated cloud services that are inherently hybrid-aware and ideal for enterprise-class organizations. It is available through the vCloud Air Network Program.

What vCloud Director does

- Allows For Enhanced Monetization And Operational Efficiency - Rapidly monetize new services, deliver radical operational efficiency to shared vCenter environments, and extend it to existing vCenters.
- Automates Data Center Services - Applies pooling, abstraction and automation to data center services such as storage, networking, and security. As a result, you can provision complete and operationally ready infrastructure without worrying about the physical configuration of hardware.
- Delivers Virtual Data Centers in Minutes - vCloud Director orchestrates the provisioning of software-defined data center (SDDC) services as complete virtual data centers that are ready for consumption in minutes.
- Delivers Multi-Tenancy Platform - You can reduce operational costs by using a single platform to host secure virtual data centers that could be consumed by multiple tenants using Role-Based Access Control.
- Supports Open APIs - Allows scripted access to consume cloud resources, such as vApp upload/download, catalog management and other operations. Using REST-based APIs and SDKs, service providers can extend the capabilities of vCloud Director.
- Enforces Security and Enables Efficient Resource Consumption - vCloud Director takes a policy-driven approach to provisioning that embeds software-defined resource consumption controls, so you can automatically enforce preconfigured IT policies.

References: <http://www.vmware.com/in.html>

Assignment 12

Title: Install & configure OSv cloud for

- a. VMWare workstation
- b. VMWare ESXi

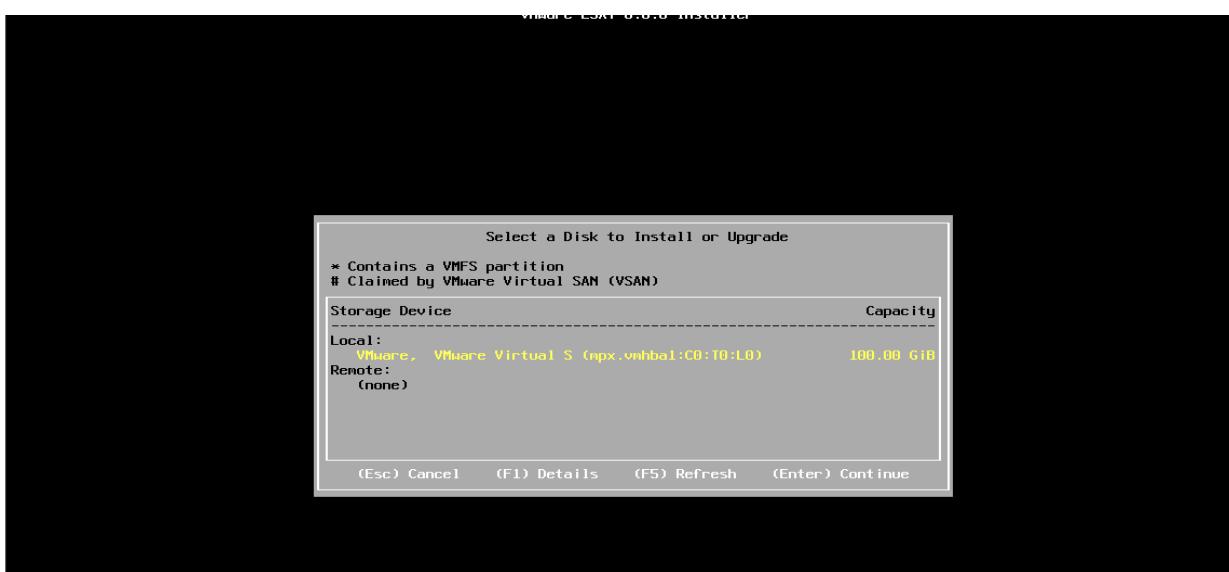
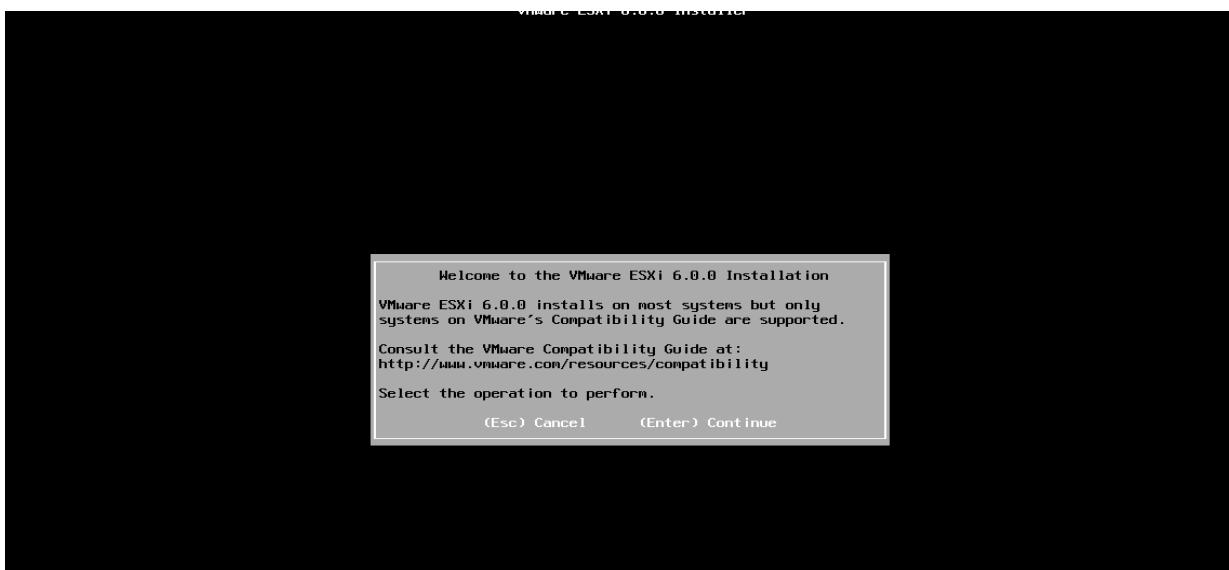
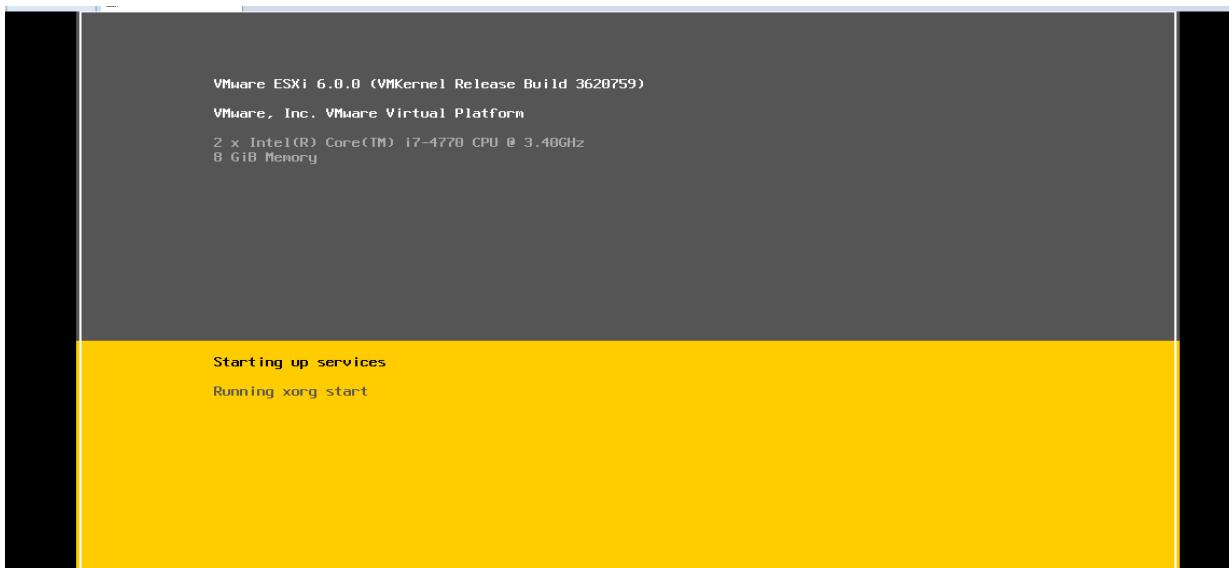
Theory:

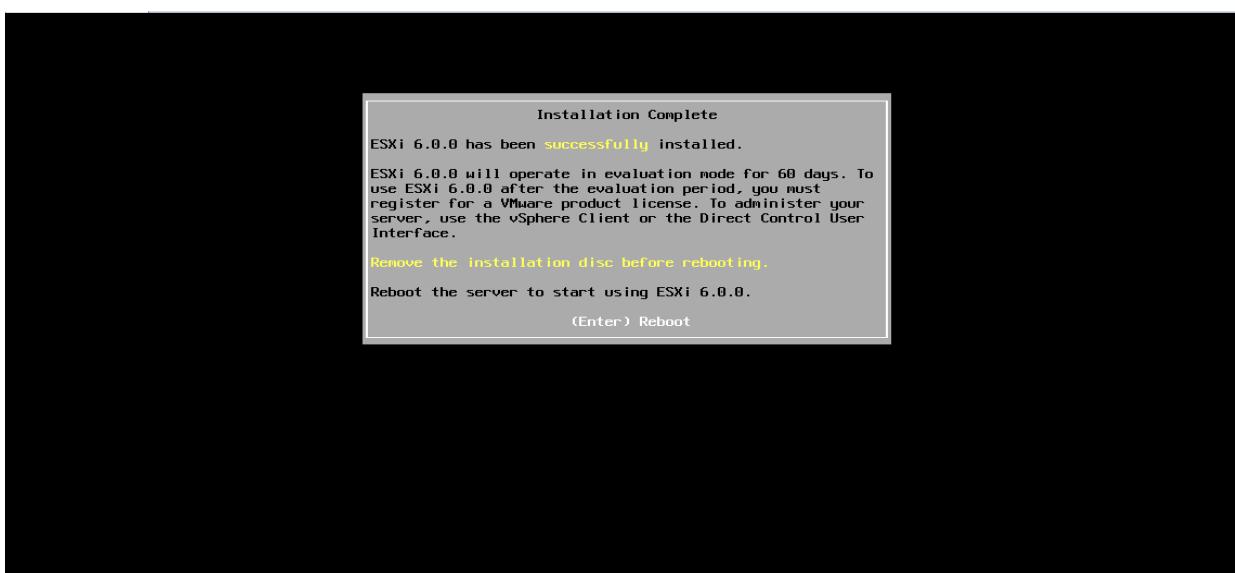
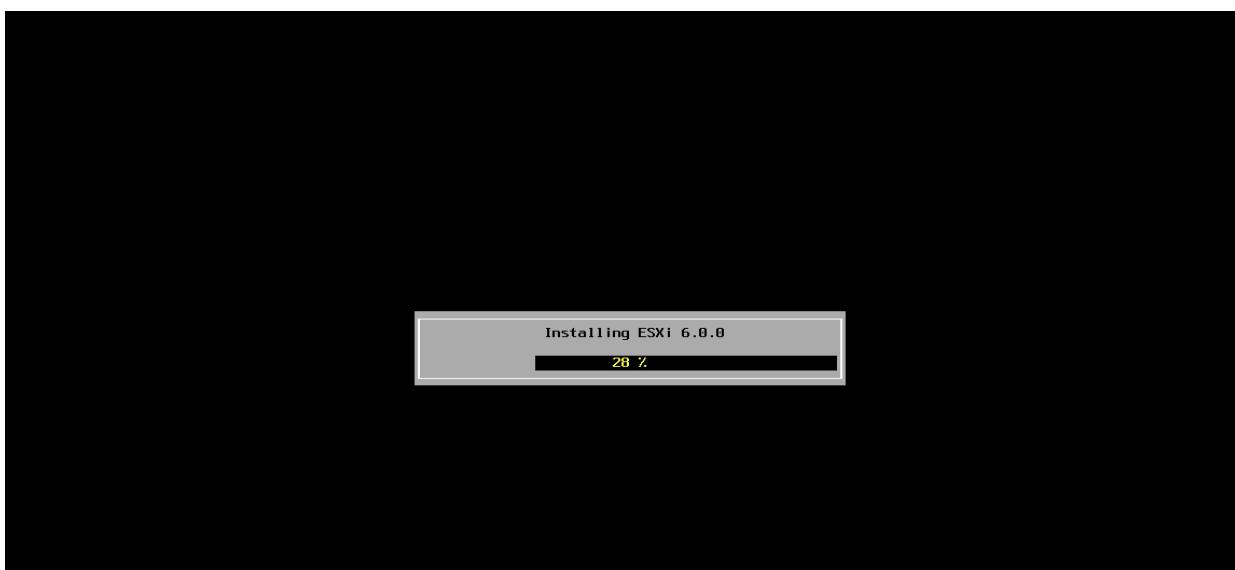
VMware ESXi (formerly ESX) is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers. As a type-1 hypervisor, ESXi is not a software application that one installs in an operating system; instead, it includes and integrates vital OS components, such as a kernel. ESXi runs on bare metal (without running an operating system) unlike other VMware products. It includes its own kernel: A Linux kernel is started first, and is then used to load a variety of specialized virtualization components, including ESX, which is otherwise known as the vmkernel component. The Linux kernel is the primary virtual machine; it is invoked by the service console. At normal run-time, the vmkernel is running on the bare computer, and the Linux-based service console runs as the first virtual machine. VMWare dropped development of ESX at version 4.1, and now uses ESXi, which does not include a Linux kernel. The vmkernel is a microkernel with three interfaces: hardware, guest systems, and the service console (Console OS).

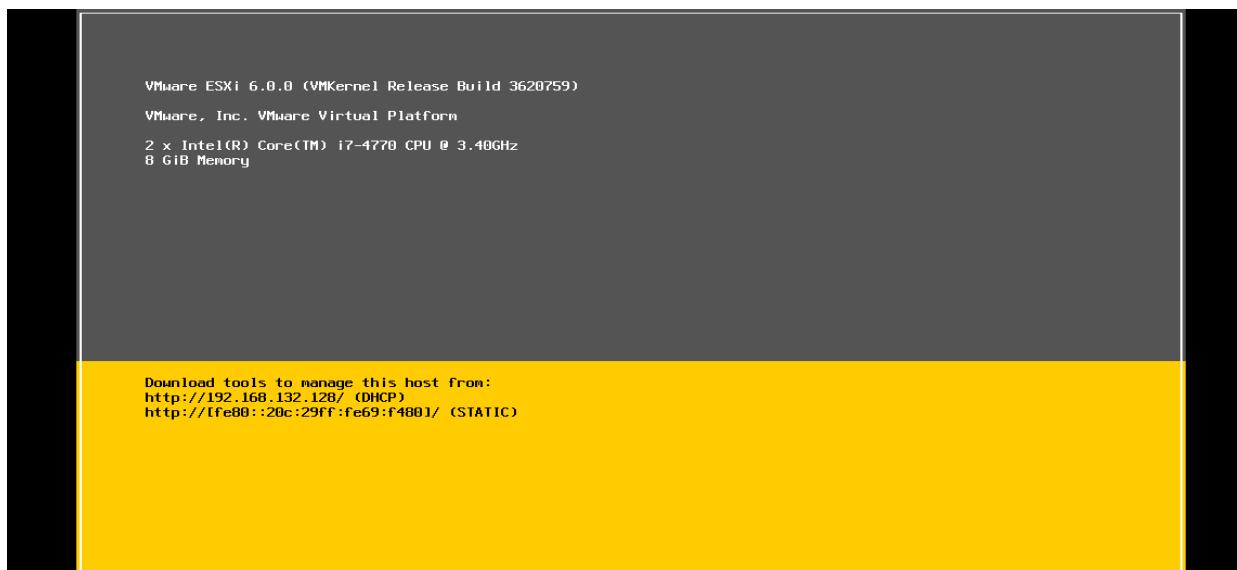
Installation:

- Download .iso of VMWare ESXi from VMWare site and boot your system to install ESXi.









Installation of OSv on ESXi

The screenshot displays the VMware ESXi host interface. At the top, the title bar shows "Welcome to VMware ESXi" and the URL "https://192.168.42.129/ui/#/host". The main window is titled "localhost.localdomain". The left sidebar, titled "Navigator", includes sections for Host, Manage, Monitor, Virtual Machines (with 0, 1, 2 icons), Storage, and Networking. The central pane shows the host's hardware configuration: CPU (2.65 GB free, 1% capacity), Memory (1.35 GB used, 34% capacity), and Storage (31.55 GB free, 972 MB used, 3% capacity). Below this, a "Recent Tasks" table shows an "Auto Start Power On" task completed successfully. A modal dialog box is open, asking for permission to participate in the Customer Experience Improvement Program (CEIP). The dialog includes a checkbox labeled "Join CEIP" and an "OK" button. To the right of the dialog, a status bar shows the date and time as "Wednesday, April 19, 2017, 17:21:54". The bottom of the screen shows a taskbar with various icons.

The image consists of three vertically stacked screenshots of the VMware ESXi web interface, all accessed via <https://192.168.42.129/ui/#/host>.

Screenshot 1: Storage Management

This screenshot shows the "Storage" section of the interface. The left sidebar has "Virtual Machines" selected. The main pane displays "localhost.localdomain - Storage" with a table of datastores. One datastore, "datastore1", is listed with details: Non-SSD, 32.5 GB capacity, 972 MB provisioned, 31.55 GB free, VMFS5 type, Supported access, and Single thin provisioning.

Name	Drive Type	Capacity	Provisioned	Free	Type	Access	Thin provis...
datastore1	Non-SSD	32.5 GB	972 MB	31.55 GB	VMFS5	Supported	Single

Screenshot 2: Virtual Machine Management

This screenshot shows the "Virtual Machines" section. The left sidebar has "Virtual Machines" selected. The main pane displays "localhost.localdomain - Virtual Machines" with a table showing "No virtual machines". A recent task table shows "Auto Start Power On" completed successfully.

Task	Target	Initiator	Queued	Started	Result	Completed
Auto Start Power On	localhost.localdomain	root	04/19/2017 17:19:07	04/19/2017 17:19:07	Completed successfully	04/19/2017 17:19:07

Screenshot 3: New Virtual Machine Creation Dialog

This screenshot shows the "New virtual machine" creation dialog. The left sidebar has "Virtual Machines" selected. The main pane shows a step-by-step wizard: "1 Select creation type" (highlighted), "2 Select OVF and VMDK files", "3 Select storage", "4 License agreements", "5 Deployment options", "6 Additional settings", and "7 Ready to complete". The "Select creation type" step is active, showing options to "Deploy a virtual machine from an OVF or OVA file" (selected) or "Register an existing virtual machine". A note states: "This option guides you through the process of creating a virtual machine from an OVF and VMDK files." Another note says: "OVA deployment is currently limited to files under 1 gigabyte in size due to web browser limitations." Buttons at the bottom include Back, Next, Finish, and Cancel.

Welcome to VMware ESXi

New virtual machine - OSv

Select OVF and VMDK files

Enter a name for the virtual machine.

OSv

Click to select files or drag/drop

Back Next Finish Cancel

Completed 04/19/2017 17:19:07

U: 0.00 kbit/s D: 26.08 kbit/s ENG IN 19-04-2017

Welcome to VMware ESXi

New virtual machine - OSv - OSv

Select OVF and VMDK files

Enter a name for the virtual machine.

OSv

x osv-v0.24.esx.ova

Back Next Finish Cancel

Completed 04/19/2017 17:19:07

U: 1.43 kbit/s D: 24.49 kbit/s ENG IN 19-04-2017

Welcome to VMware ESXi

New virtual machine - OSv - OSv

Select storage

Select the datastore in which to store the configuration and disk files.

The following datastores are accessible from the destination resource that you selected. Select the destination datastore for the virtual machine configuration files and all of the virtual disks.

Name	Capacity	Free	Type	Thin pro...	Access
datastore1	32.5 GB	31.55 GB	VMFS5	Supported	Single

Back Next Finish Cancel

Completed 04/19/2017 17:19:07

U: 21.54 kbit/s D: 150.10 kbit/s ENG IN 19-04-2017

Screenshot 1: Deployment options

Screenshot 2: Ready to complete

Screenshot 3: Virtual Machines list

Welcome to VMware ESXi

localhost.localdomain - Virtual Machines

Virtual machine	Status	Used space	Host name	Host CPU	Host mem...
OSv	Normal	32 MB	Unknown	0 MHz	0 MB

Recent Tasks

Task	Target	Initiator	Queued	Started	Result	Completed
Upload disk - 1 of 1	OSv	root	04/19/2017 17:29:40	04/19/2017 17:29:40	Completed successfully	04/19/2017 17:29:42
Reconfig VM	OSv	root	04/19/2017 17:29:42	04/19/2017 17:29:42	Completed successfully	04/19/2017 17:29:42
Power On VM	OSv	root	04/19/2017 17:29:41	04/19/2017 17:29:41	Completed successfully	04/19/2017 17:29:42
Import VApp	Resources	root	04/19/2017 17:29:32	04/19/2017 17:29:32	Completed successfully	04/19/2017 17:29:40

U: 10.76 kbit/s D: 84.09 kbit/s ENG 17:30 IN 19-04-2017

Welcome to VMware ESXi

localhost.localdomain - Virtual Machines

OSv

Console | Power on | Power off | Suspend | Reset | Edit settings | Refresh | Actions

Authentication Required

https://192.168.42.129 is requesting your username and password. The site says: "VMware HTTP server"

User Name: root
Password:

OK Cancel

General

Guest OS: Ubuntu Linux (64-bit)
Compatibility: ESXi 5.0 and later (VM version 8)
VMware Tools: No

Network

Memory: 1 vCPUs
1 GB

Storage

Hard disk 1: 10 GB
Network adapter 1: VM Network (Connected)

Notes

Recent Tasks

Task	Target	Initiator	Queued	Started	Result	Completed
Upload disk - 1 of 1	OSv	root	04/19/2017 17:29:40	04/19/2017 17:29:40	Completed successfully	04/19/2017 17:29:42
Reconfig VM	OSv	root	04/19/2017 17:29:42	04/19/2017 17:29:42	Completed successfully	04/19/2017 17:29:42
Power On VM	OSv	root	04/19/2017 17:29:41	04/19/2017 17:29:41	Completed successfully	04/19/2017 17:29:42
Import VApp	Resources	root	04/19/2017 17:29:32	04/19/2017 17:29:32	Completed successfully	04/19/2017 17:29:40

U: 0.00 kbit/s D: 29.55 kbit/s ENG 17:30 IN 19-04-2017

Welcome to VMware ESXi

localhost.localdomain - Virtual Machines

OSv

Console | Power on | Power off | Suspend | Reset | Edit settings | Refresh | Actions

VMware Tools is not installed in this virtual machine. VMware Tools allows detailed guest information to be displayed as well as allowing you to perform operations on the guest OS, e.g. graceful shutdown, reboot, etc. You should install VMware Tools. [Actions](#)

General Information

Networking: No network information
VMware Tools: Not installed

Hardware Configuration

CPU: 1 vCPUs
Memory: 1 GB

Recent Tasks

Task	Target	Initiator	Queued	Started	Result	Completed
Upload disk - 1 of 1	OSv	root	04/19/2017 17:29:40	04/19/2017 17:29:40	Completed successfully	04/19/2017 17:29:42
Reconfig VM	OSv	root	04/19/2017 17:29:42	04/19/2017 17:29:42	Completed successfully	04/19/2017 17:29:42
Power On VM	OSv	root	04/19/2017 17:29:41	04/19/2017 17:29:41	Completed successfully	04/19/2017 17:29:42
Import VApp	Resources	root	04/19/2017 17:29:32	04/19/2017 17:29:32	Completed successfully	04/19/2017 17:29:40

U: 0.00 kbit/s D: 169.86 kbit/s ENG 17:31 IN 19-04-2017

Three screenshots of the VMware ESXi web interface showing the configuration and monitoring of a virtual machine named OSv.

Screenshot 1: The OSv configuration page. It shows the following details:

- General Information:** Networking (No network information), VMware Tools (Not installed), Storage (1 disk), Notes (Edit notes).
- Hardware Configuration:** CPU (1 vCPUs), Memory (1 GB), Hard disk 1 (10 GB, VM Network (Connected)), Network adapter 1 (VM Network (Connected)), Video card (4 MB), Others (Additional Hardware).
- Resource Consumption:** Consumed host CPU (0 MHz), Consumed host memory (0 MB), Active guest memory (0 MB).
- Recent Tasks:** A table showing completed tasks:

Task	Target	Initiator	Started	Result	Completed
Upload disk - 1 of 1	OSv	root	04/19/2017 17:29:40	Completed successfully	04/19/2017 17:29:42
Reconfig VM	OSv	root	04/19/2017 17:29:42	Completed successfully	04/19/2017 17:29:42
Power On VM	OSv	root	04/19/2017 17:29:41	Completed successfully	04/19/2017 17:29:42
Import VApp	Resources	root	04/19/2017 17:29:32	Completed successfully	04/19/2017 17:29:40

Screenshot 2: The OSv monitoring page. It shows the following details:

- Performance summary last hour:** A graph showing Consumed host CPU (blue line) and Consumed host memory (red line). The memory usage spikes sharply from 17:29:40 to 17:30:00, reaching nearly 100%.
- Resource Consumption:** Consumed host CPU (0 MHz), Consumed host memory (0 MB), Active guest memory (0 MB).
- Recent Tasks:** A table showing completed tasks (same as Screenshot 1).

Screenshot 3: The OSv configuration page again, this time with a terminal window open. The terminal output shows:

```

eth0: 192.168.42.130
pthread_setcancelstate() stubbed
# date
Wed Apr 19 12:01:50 UTC 2017
# -

```

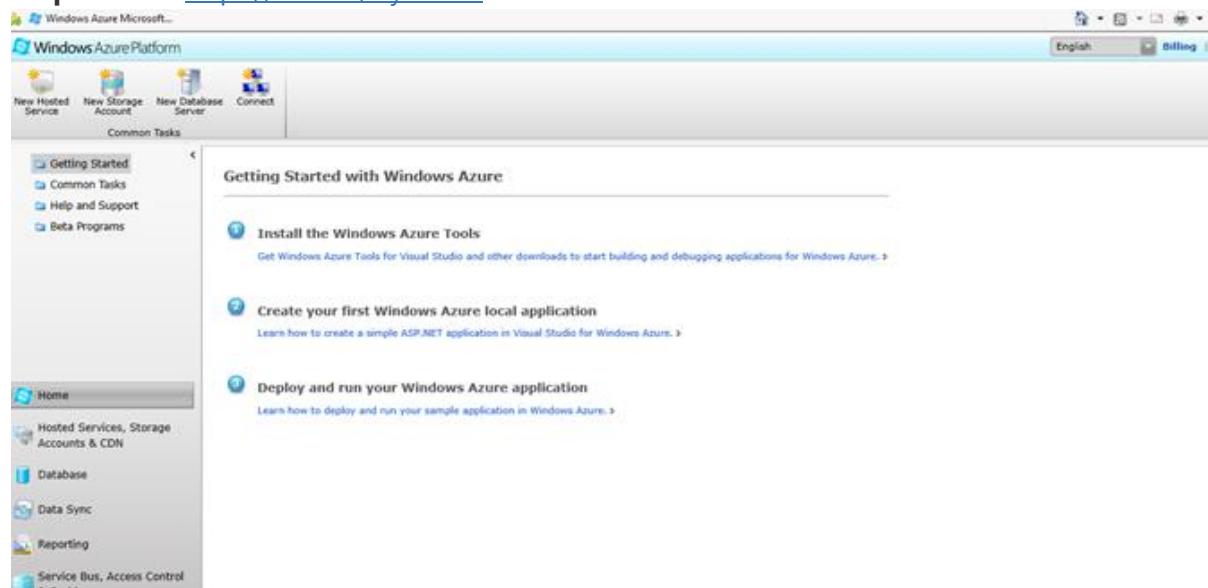
The terminal also displays a message about VMware Tools being not installed. The configuration details are identical to Screenshot 1.

Assignment 13

Title: Install and demonstrate the Microsoft Azure cloud OS.

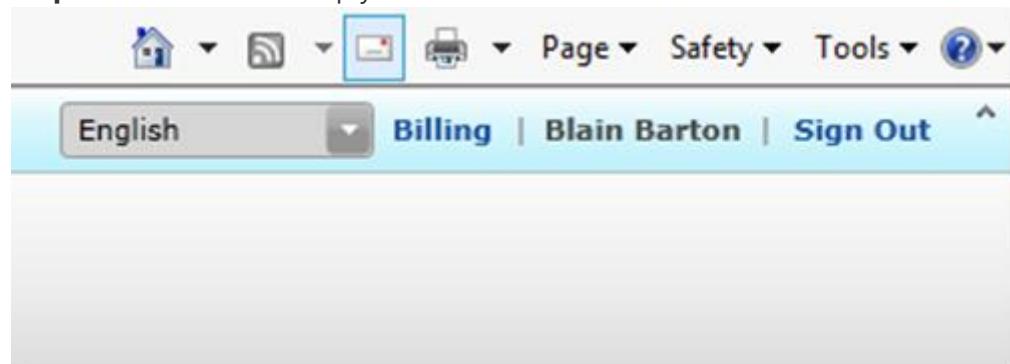
Steps:

Step 1: Go to <https://aka.ms/try-azure>



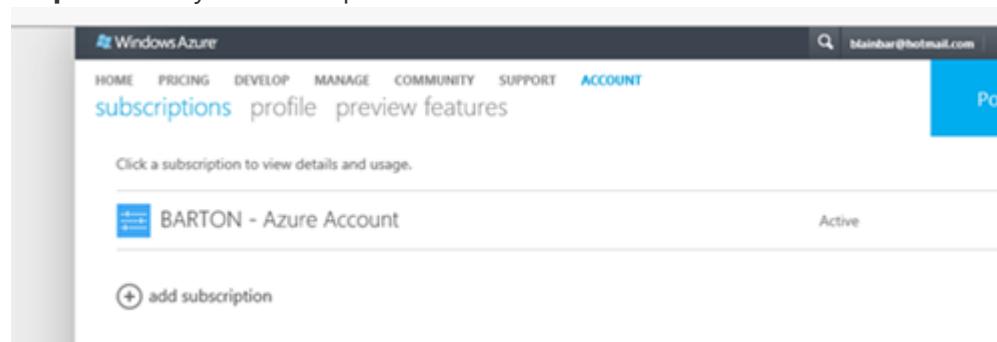
The screenshot shows the Windows Azure Platform homepage. On the left, there's a sidebar with links like 'New Hosted Service', 'New Storage Account', 'New Database Server', and 'Connect'. Below that is a 'Common Tasks' section with 'Getting Started', 'Common Tasks', 'Help and Support', and 'Beta Programs'. The main content area is titled 'Getting Started with Windows Azure' and lists three steps: 'Install the Windows Azure Tools', 'Create your first Windows Azure local application', and 'Deploy and run your Windows Azure application'. Each step has a link to learn more.

Step 2: Provision and setup your account



The screenshot shows the Windows Azure dashboard. At the top, there's a navigation bar with icons for home, RSS, mail (highlighted with a blue border), printer, page, safety, tools, and help. Below the navigation bar is a language dropdown set to 'English', a 'Billing' link, the user's name 'Blain Barton', and a 'Sign Out' link. The main content area is currently empty.

Step 3: Here's your subscription as "Active"



The screenshot shows the 'subscriptions' page in the Windows Azure portal. The top navigation bar includes 'HOME', 'PRICING', 'DEVELOP', 'MANAGE', 'COMMUNITY', 'SUPPORT', and 'ACCOUNT'. The 'ACCOUNT' tab is selected. Below the navigation is a search bar and a user profile link 'mainbar@hotmail.com'. The main content area displays a message: 'Click a subscription to view details and usage.' followed by a table with one row. The table has columns for a thumbnail icon, the subscription name 'BARTON - Azure Account', and the status 'Active'. At the bottom of the table is a link '+ add subscription'.

Step 4: Add in the "Virtual Machines & Virtual Networks – Try it Now!

The screenshot shows the Windows Azure portal interface. At the top, there's a progress bar with the text "free access for up to ten apps." Below it is a "LEARN MORE" button. The main content area has a title "Virtual Machines & Virtual Networks" with a green "try it now" button. To the left is a icon of a computer monitor with a cube on it. The text describes deploying and running virtual machines in Windows Azure, mentioning virtual private networks (VPNs) and secure connections to on-premises infrastructure. There's also a "LEARN MORE" button at the bottom.

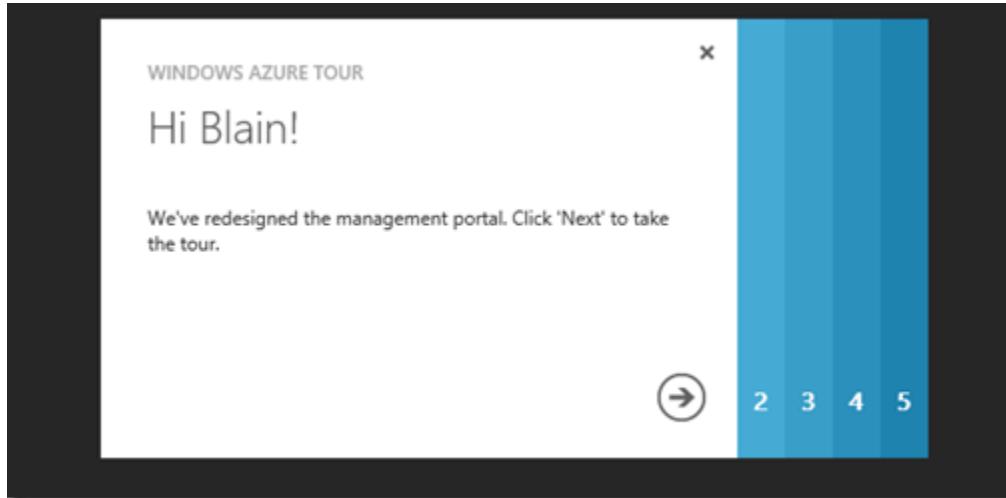
Step 5: Select the Azure Account

This screenshot shows a modal dialog box over the Windows Azure portal. The dialog is titled "Mobile Services" and says "ADD PREVIEW FEATURE". It features a large orange icon of a smartphone with a cube on its screen. The text "VIRTUAL MACHINES & VIRTUAL NETWORKS" is displayed, followed by "CHARGES MAY BE INCLUDED". A dropdown menu under "SUBSCRIPTIONS" shows "BARTON - Azure Account". Below the dialog, a message says "You will receive an e-mail when this feature is enabled for your subscription." At the bottom, there's a checkbox for accepting terms and conditions, which is checked, and a note about discounts on compute rates. A "try it now" button is visible on the right side of the dialog.

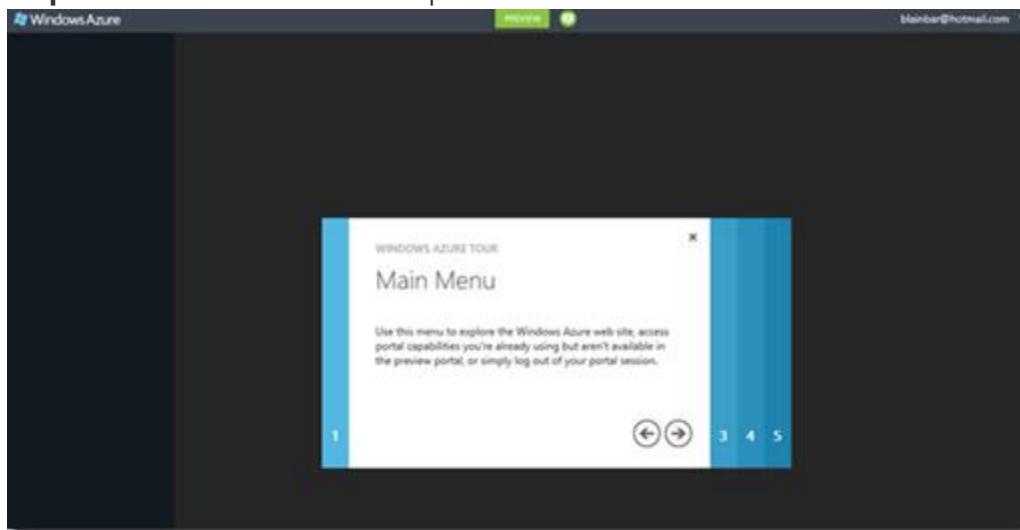
Step 6: You are now setup!

The screenshot shows a confirmation message from Microsoft. The background features a hexagonal pattern with three colored hexagons labeled "COMPUTE" (blue), "STORAGE" (orange), and "NETWORKING" (green). The main text says "Congratulations!" and informs the user that their pending request to access the Windows Azure Virtual Machines preview has been approved. It encourages them to use the new improved preview portal. Two buttons are present: "use the preview portal" with a green arrow icon and "use the old portal". At the bottom, there's a Microsoft logo and links for "2010 Microsoft Corporation | Trademarks | Privacy & Cookies | Terms of Use | Support | Account | Feedback".

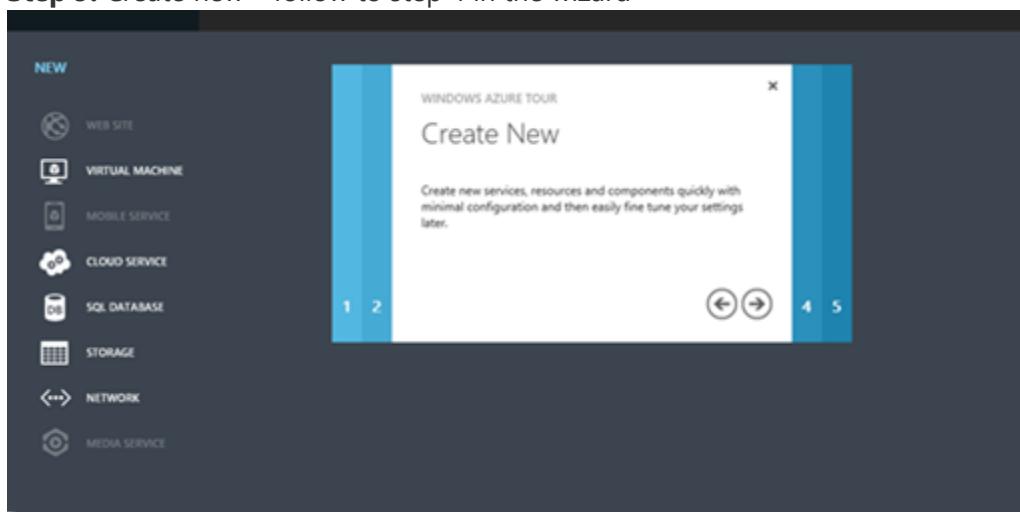
Step 7: Welcome Screen – follow to step 2 in the wizard.



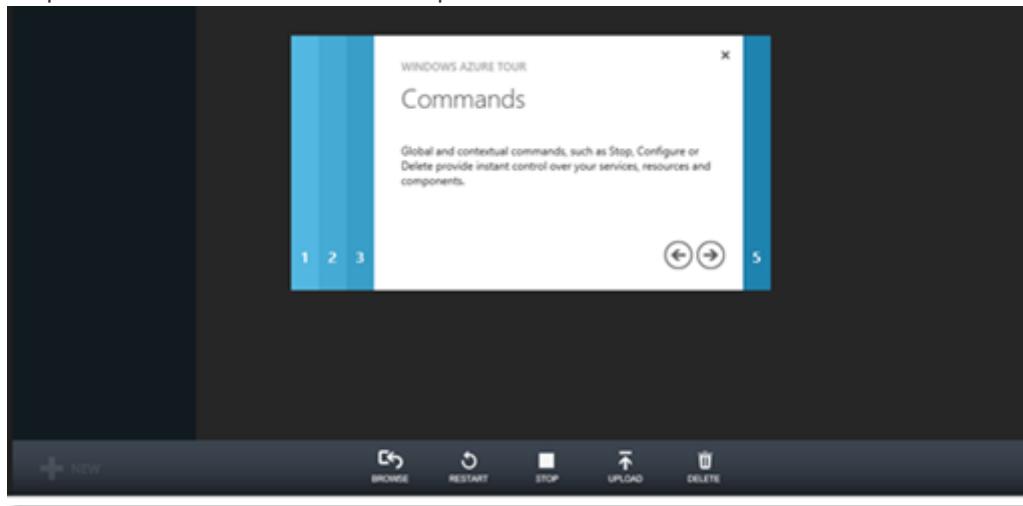
Step 8: Main Menu – follow to step 3 in the wizard.



Step 9: Create new – follow to step 4 in the wizard

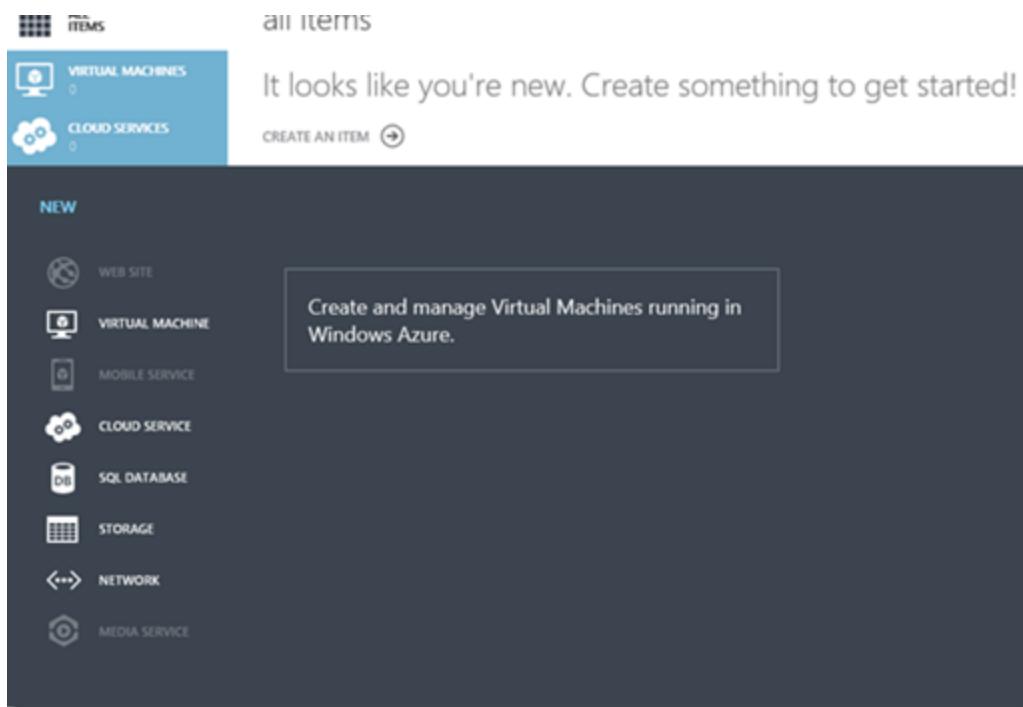


Step 10: Commands – follow to step 5 in the wizard

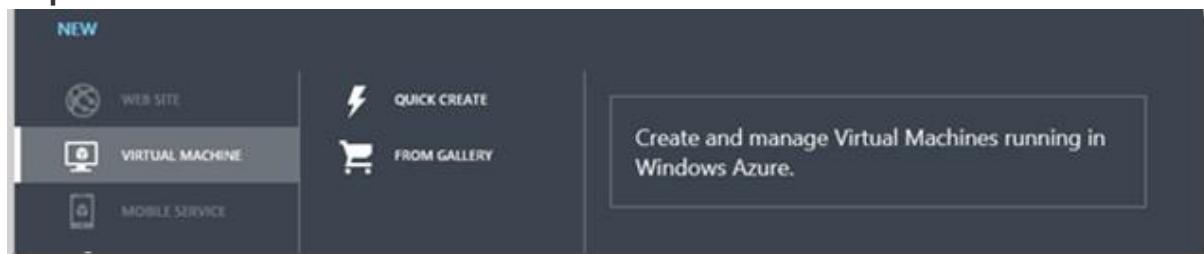


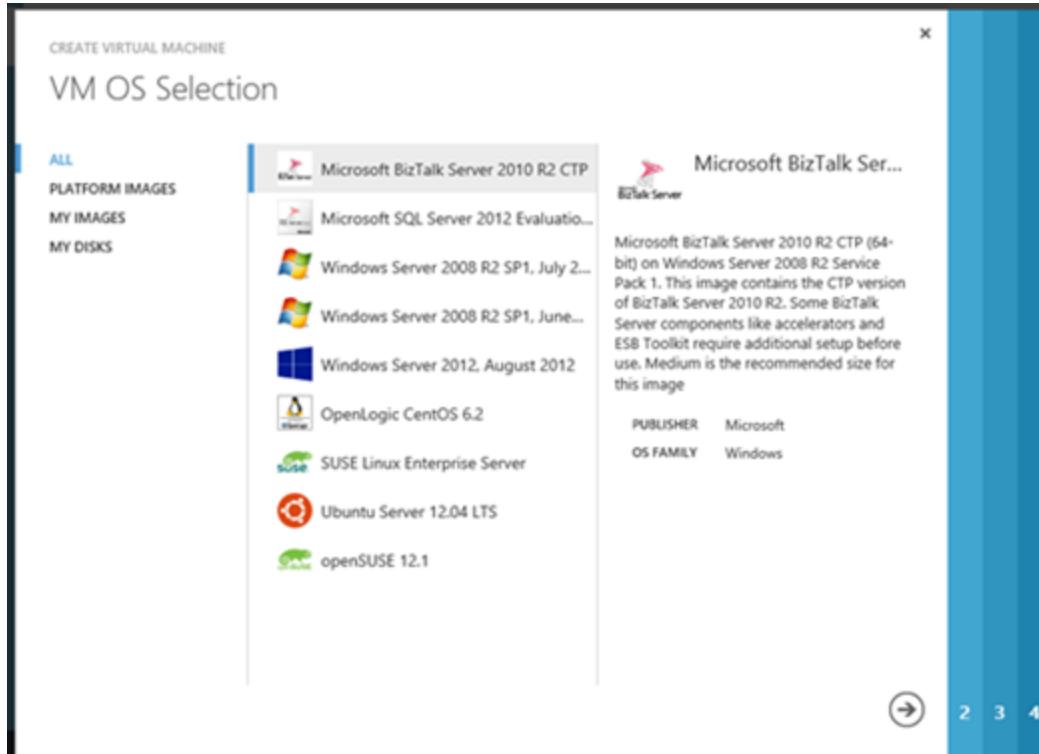
Step 11: Hey newbie, Create an Item! lol

Select Virtual Machine

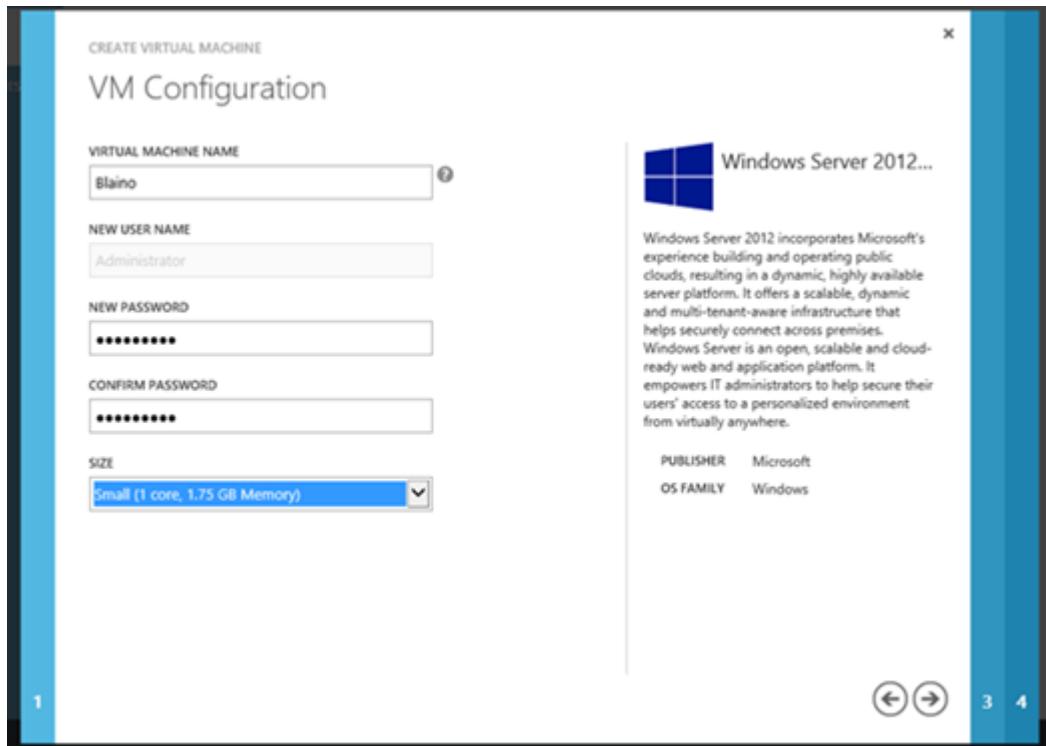


Step 12: Create a new Virtual Machine FROM GALLERY

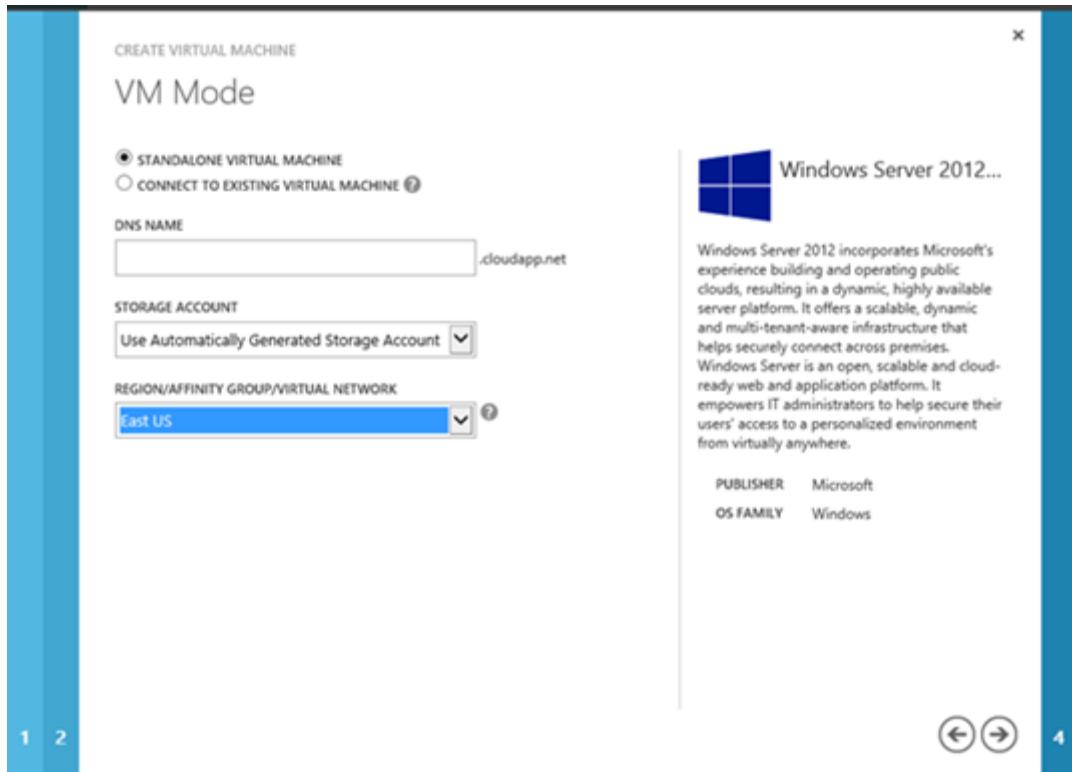


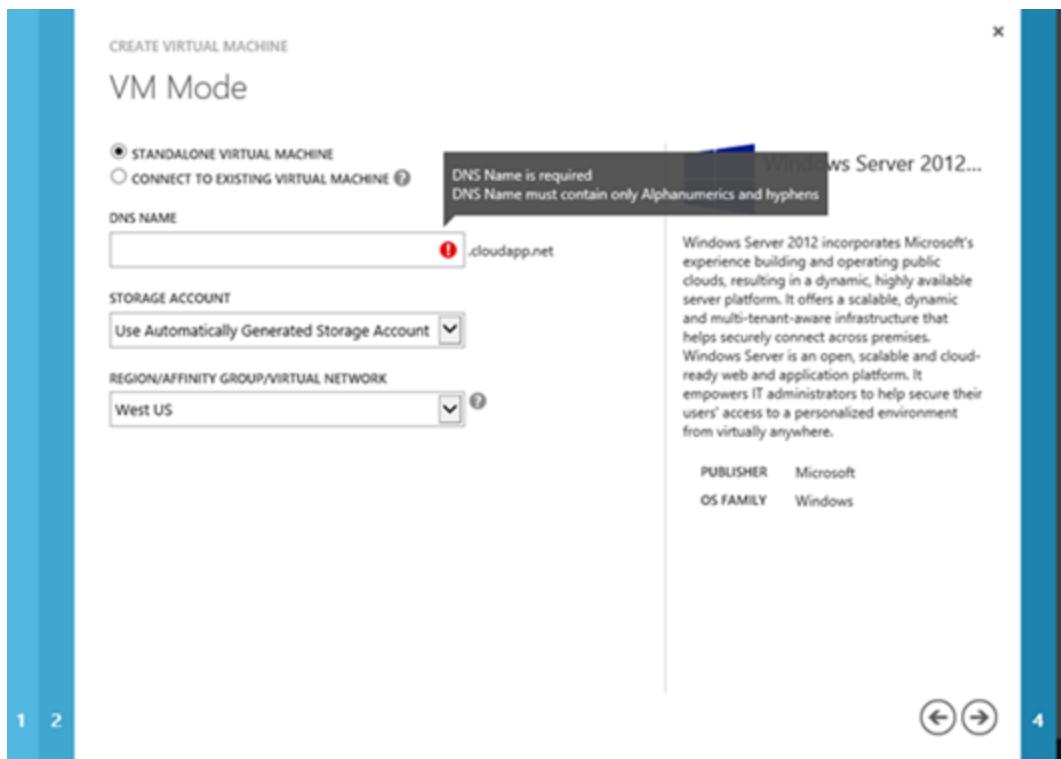
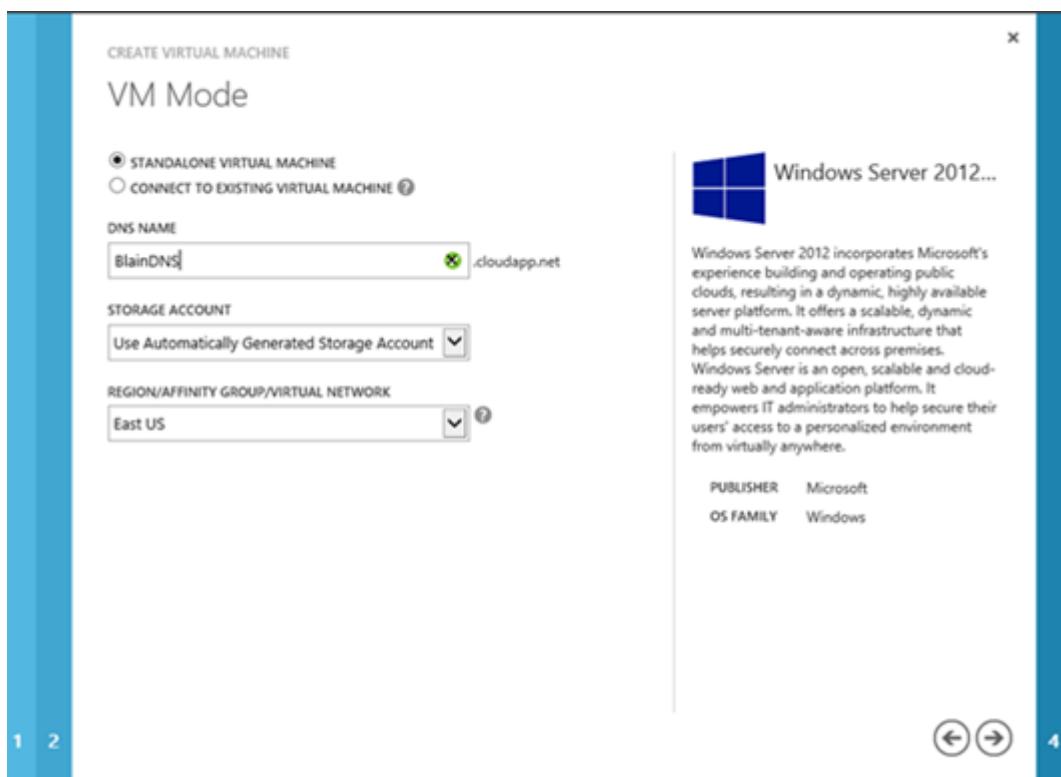
Step 13: Review the VM OS Selections**Step 14:** Select Windows Server 2012

Step 15: Select the VM Configuration, Give it a Virtual Machine Name and Password, and pick SMALL

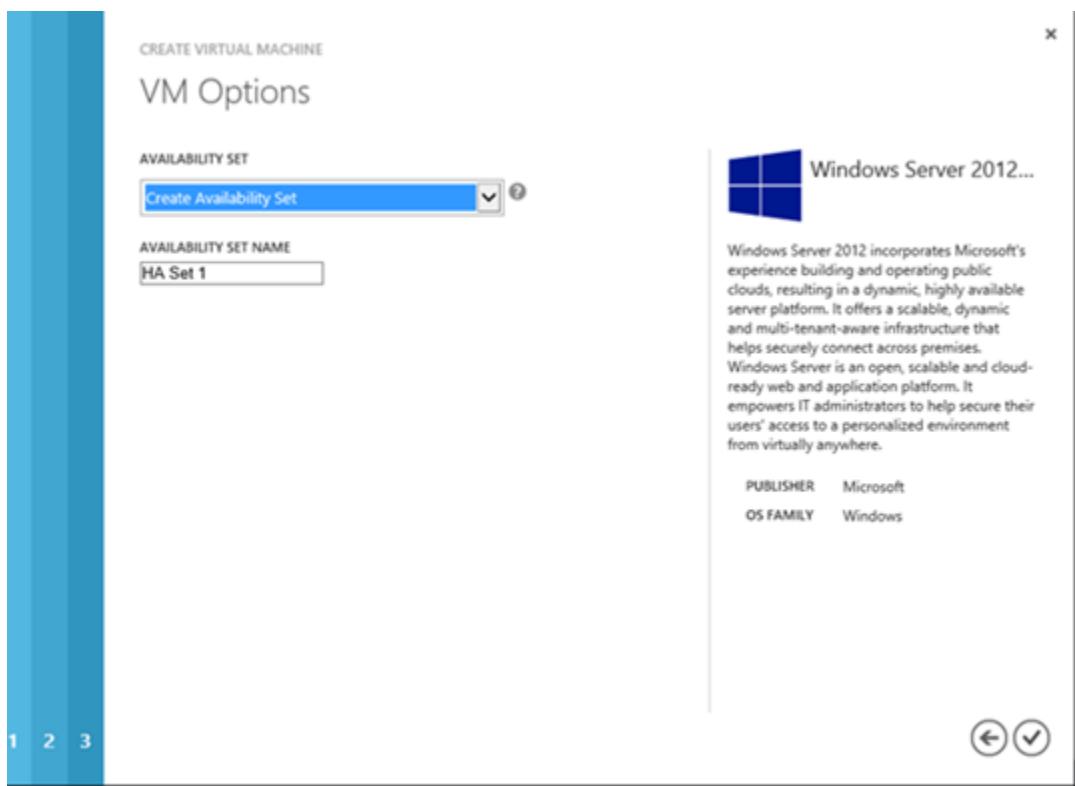


Step 16: Select the VM Mode – Standalone Virtual Machine



Step 17: Give it your own DNS name**Step 18:** Give it a friendly name for your DNS, each VM Role will have one.

Step 19: Creates the Availability State for the VM Role to run Highly Available VM Roles!



Step 20: Provisioning the Virtual Machine

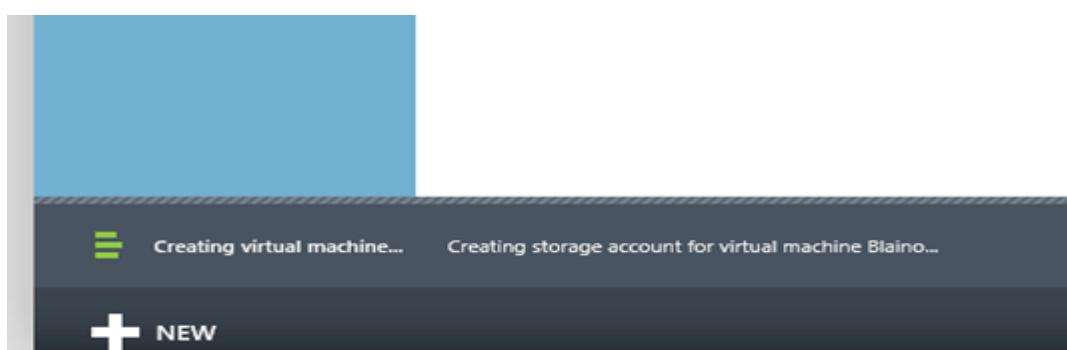
Windows Azure

PREVIEW

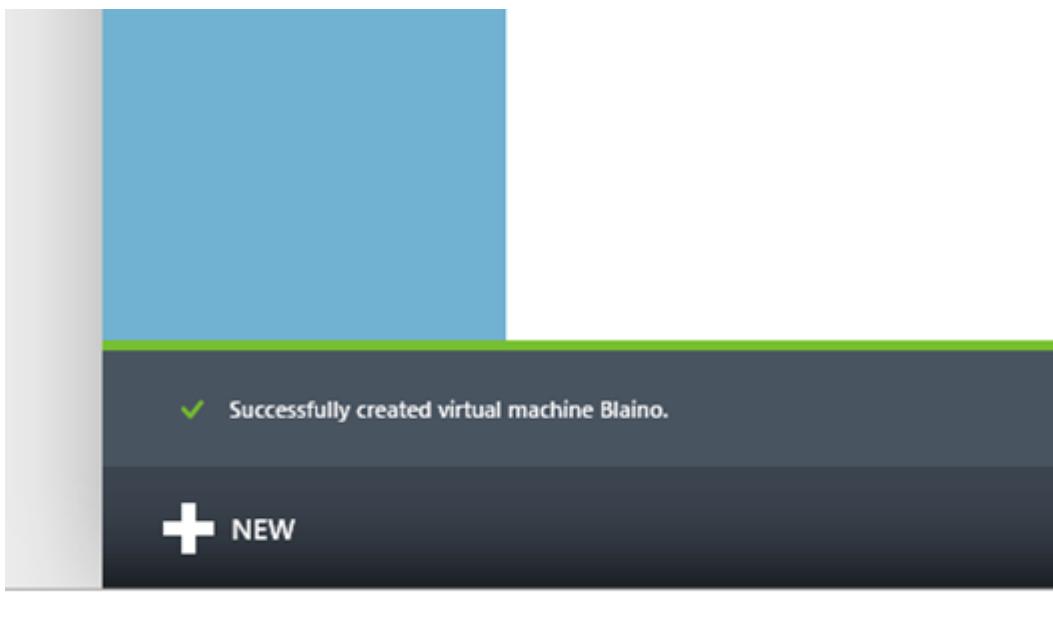
all items

NAME	TYPE	STATUS	SUBSCRIPTION
Blaino	Virtual Machine	Starting (Provisioning)	-

Step 21: Creating the Storage Account for the Virtual Machine



Step 22: Successfully created virtual machine



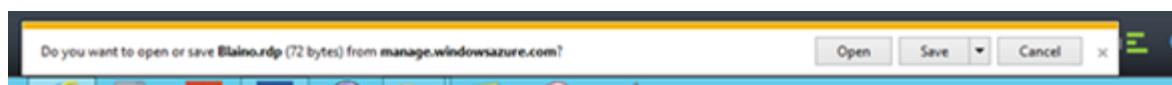
Step 23: Virtual Machine is Starting!

A screenshot of the Windows Azure portal's "virtual machines" section. The left sidebar shows "VIRTUAL MACHINES" with a count of 1. The main table lists one VM named "Blaino" with a status of "Starting". Other columns include "SUBSCRIPTION", "LOCATION", and "DNS NAME".

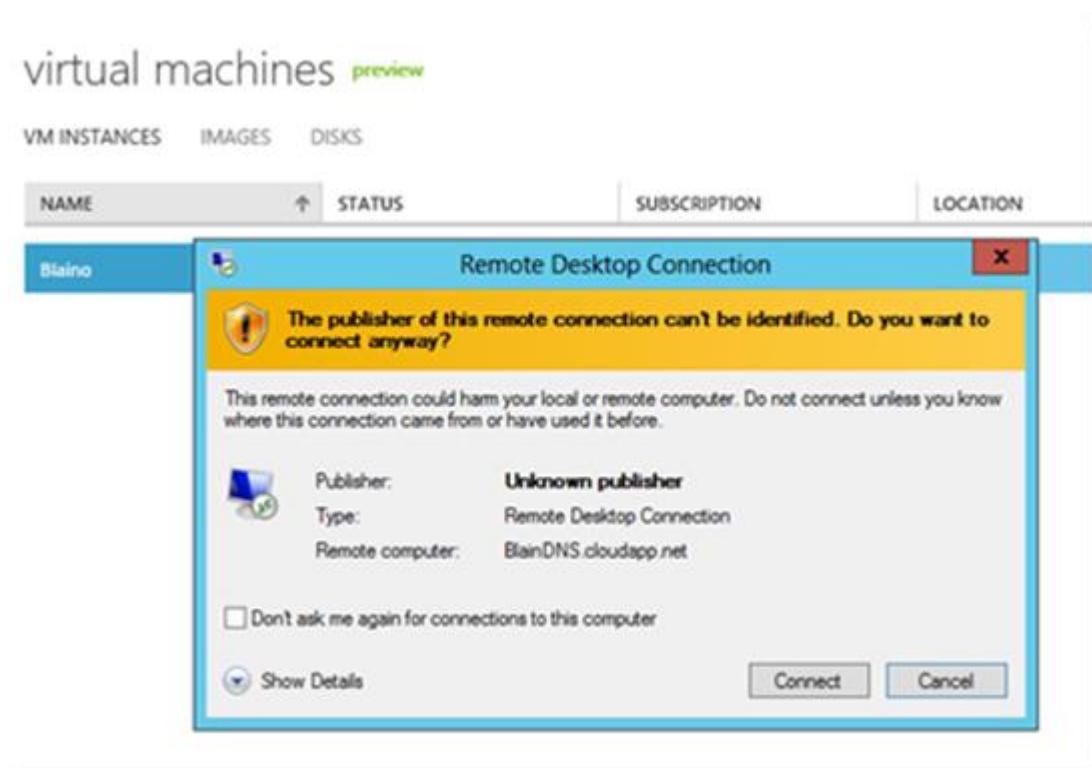
Step 24: Virtual Machine is Running!

A screenshot of the Windows Azure portal's "virtual machines" section. The left sidebar shows "VIRTUAL MACHINES" with a count of 1. The main table lists one VM named "Blaino" with a status of "Running". Other columns include "SUBSCRIPTION", "LOCATION", and "DNS NAME". At the bottom of the screen, there are three buttons: "CONNECT", "RESTART", and "SHUTOFF".

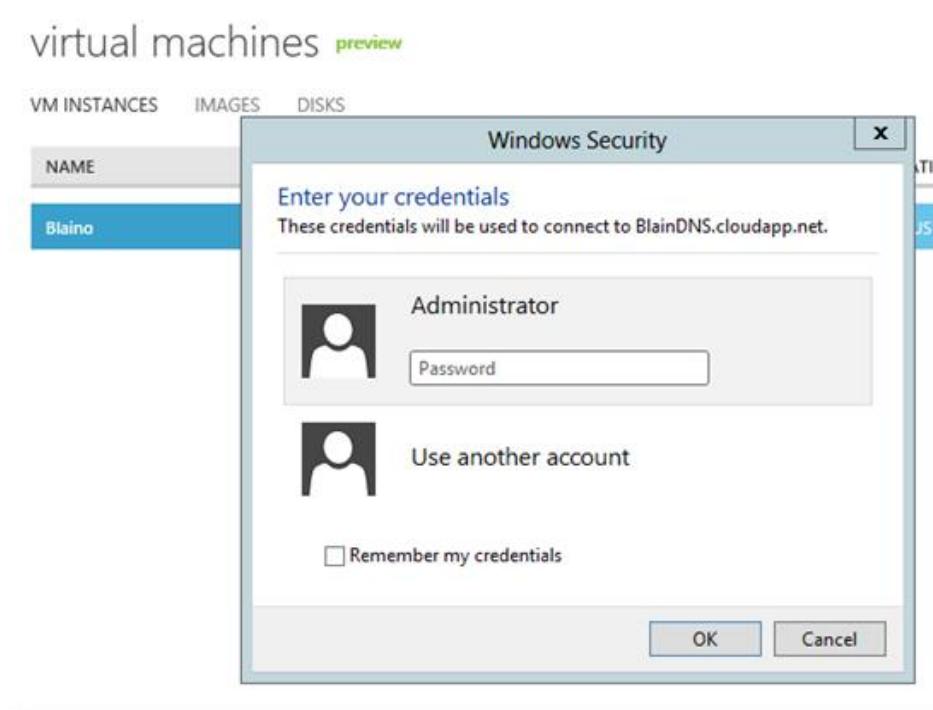
Step 25: Opening the RDP request, Click **Open**



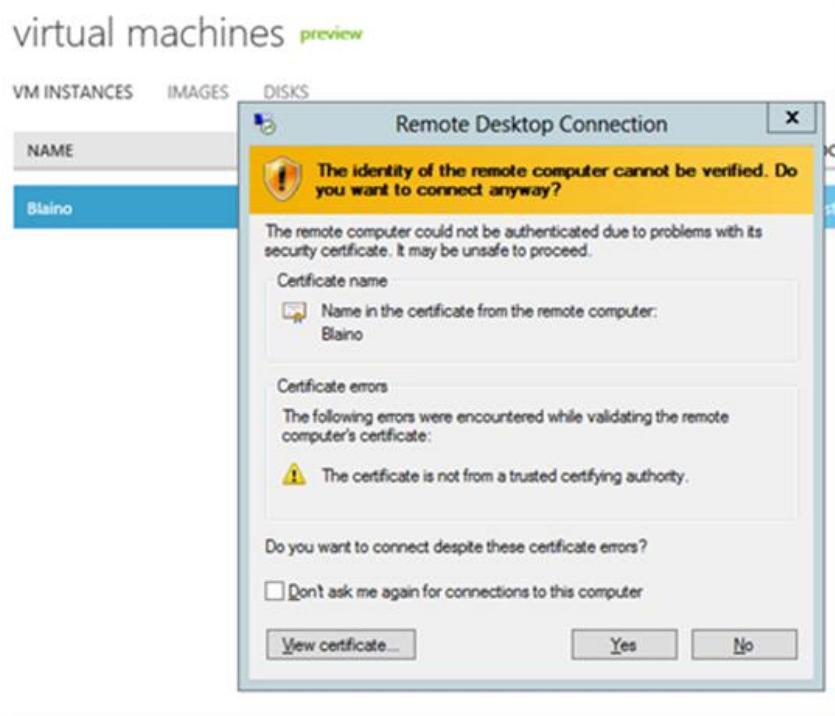
Step 26: Opening the Remote Desktop Connection to view the Virtual Machine



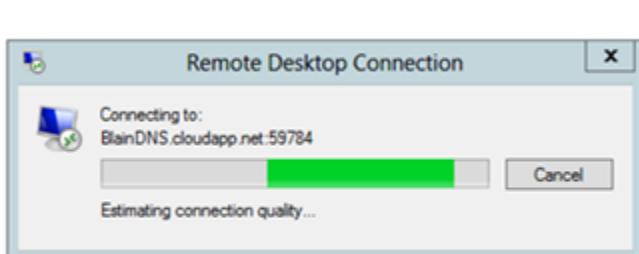
Step 27: Enter your Administrator Username and Password



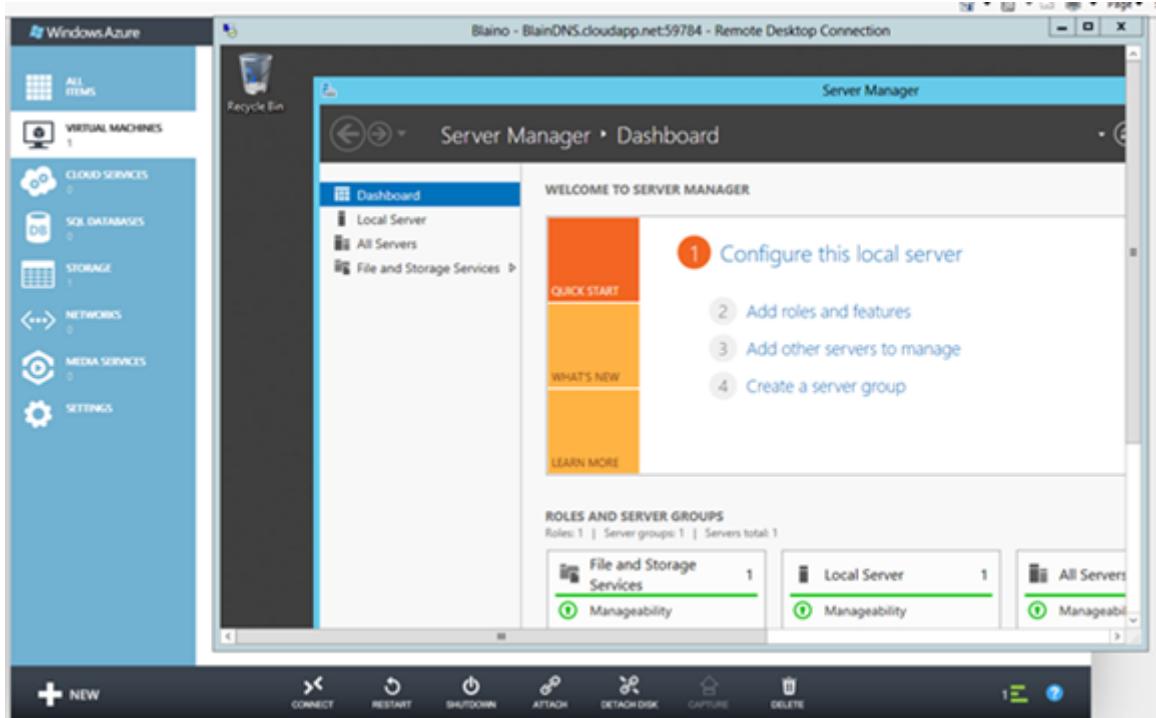
Step 28: Accept the certificate here... (Note: Later you can add your own VM's with the certificate that goes with that virtual hard disk.)



Step 29: Establish the connection with the Virtual Machine!



Step 30: Here, you see your Windows Server 2012 virtual machine via the RDP session and Server Manager. Congrats on your first steps to setting up Windows Azure. I plan on helping you through your journey as well!



Conclusion:

Successfully installed and demonstrated the Microsoft Azure cloud OS.

References:

<https://aka.ms/try-azure>