



Class:	BE IT / Semester – VII (CBGS)	Subject Code BEITC702
Subject:	Cloud Computing	
Semester :	Second Half of 2017 (AY: 2017-18)	Schedule: 02 hours per week
DOC	July 03, 2018	

..... List of Practical assignments and plan of implementation.....

No	Title	Experiment/ Assignment type	CO	PO	Plan
01	Configuration of KVM on Fedora and OS Installation	Application	CO1, CO2	1,2,3, 5	Week 1
02	Installation of Citrix Xen Server and Configuration	Application	CO1, CO3	1,2,4, 5	Week 2
03	OpenStack Configuration and Testing on Platform 9 Sandbox	Application	CO1, CO3	1,2,4, 5	Week 3 Week 4
04	Configuration of RabbitMQ / Azure Service Bus	Application	CO2, CO6	1,2,4, 5,7	Week 5
05	MongoDB NoSQL database CRUD Operations in .NET	Application	CO2, CO4	1,2,4, 5	Week 6
06	Python Program deployment on Google App Engine	Application	CO3, CO5	1,2,4, 5, 6	Week 7
07	IBM Bluemix web application development	Application	CO1, CO4	1,2,4, 5, 9	Week 8 Week 9
08	Azure Blob Storage / Web role/ Worker Role Example	Application	CO3, CO6	1,2,4, 5, 12	Week 10
09	Big Data and Deep Web	Case study	CO3, CO4	1,2,4, 5, 10, PSO 1,2	Week 11
10	Blockchain on cloud computing	Case study	CO3, CO4	1,2,4, 5, 10, PSO1, 2	Week 12

Subject Teacher

Head of Department

Dr. Vinayak Ashok Bharadi

Dr. Vinayak Ashok Bharadi



Hope Foundation's  
**Finolex Academy of Management and Technology, Ratnagiri**  
Information Technology Department

Subject name: Cloud Computing      Subject Code: BEITC702

Class      BE IT      Semester – VIII  
(CBGS)      Academic year: 2018-19

Name of Student      **QUIZ Score :**

Roll No      Assignment/Experiment No.      01

**Title: Configuration of KVM on Fedora and OS Installation**

**1. Course objectives applicable COB1.**Get familiar with cloud computing fundamentals, Virtualization, and types of virtualization

**COB2.**Understand benefits of various cloud services and Handle Open Source Cloud Implementation and Administration

**2. Course outcomes applicable:**

**CO1 :** Define cloud computing, list various avenues for cloud computing applications, Explain different cloud computing fundamental and virtualization techniques and its role in cloud computing.

**CO2 -**List benefits of different cloud services, types of services, Concept of Anything as A Service. Handle Open Source Cloud Implementation and Administration using OpenStack or Eucalyptus.

**3. Learning Objectives:**

1. To understand the concept of type 2 hypervisor
2. Configure type 2 hypervisor and create a virtual machine
3. To setup windows installer in virtual environment
4. To assess the performance of Virtual Machine

**4. Practical applications of the assignment/experiment: Hypervisors are main components of Cloud Implementation**

**5. Prerequisites:**

1. Knowledge of Virtual box
2. Windows setup knowledge

**6. Hardware Requirements:**

1. PC with 4GB RAM, 500GB HDD

**7. Software Requirements:**

1. Ubuntu / Windows Setup
2. Oracle VirtualBox Setup for appropriate Operating System,

**8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)**

1. What is a Hypervisor?
2. How many types of hypervisor are there?
3. Virtual Box is which type of hypervisor?
4. What are the system requirements for windows 10 installation?
5. What changes can you bring in the current setup for performance improvement?

**9. Experiment/Assignment Evaluation:**

Sr. No.	Parameters	Marks obtained	Out of
---------	------------	----------------	--------

<b>1</b>	Technical Understanding (Assessment may be done based on Q & A <b>or</b> any other relevant method.) Teacher should mention the other method used -		<b>6</b>
<b>2</b>	Neatness/presentation		<b>2</b>
<b>3</b>	Punctuality		<b>2</b>
<b>Date of performance (DOP)</b>		<b>Total marks obtained</b>	<b>10</b>
<b>Date of checking (DOC)</b>		<b>Signature of teacher</b>	

## **9. Theory: <Preferably given as handwritten work for students>**

### **Virtualization :**

In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.

Virtualization began in the 1960s, as a method of logically dividing the system resources provided by mainframe computers between different applications. Since then, the meaning of the term has broadened.

There are two types of hypervisors: Type 1 and Type 2. Type 2 hypervisors support guest virtual machines by coordinating calls for CPU, memory, disk, network and other resources through the physical host's operating system. This makes it easy for an end user to run a virtual machine on a personal computing device. Examples of this type of hypervisor include VMware Fusion, Oracle Virtual Box, Oracle VM for x86, Solaris Zones, Parallels and VMware Workstation. [2][3]

In contrast, a Type 1 hypervisor (also called a bare metal hypervisor) is installed directly on physical host server hardware just like an operating system. Type 1 hypervisors run on dedicated hardware. They require a management console and are used in data centers. Examples of this type of hypervisor include Oracle OVM for SPARC, ESXi, Hyper-V and KVM.

Regardless of the implementation, virtual machines (VMs) and their guest operating systems are typically unaware of which type of hypervisor is implemented, as they interact only with the hypervisor itself.

### **10. Precautions :**

1. Don't format live machine
2. Install OS on virtual machine only
3. Make sure that you have valid license of the software and operating systems that you are going to use

## **11. Installation Steps / Performance Steps -**

### **Installing KVM on Fedora 26 / Fedora 25**

[\*\*KVM\*\*](#) stands for Kernel-Based Virtual Machine, is a virtualization software which provides ability to run a multiple guest operating systems with the help of hardware virtualization extensions. It supports wide variety of guest operating system's such as [Linux](#), [Windows](#), Solaris, Haiku, REACT OS and much more. This guide shows you how to install KVM on [Fedora 26 / Fedora 25](#).

KVM can be managed using command line or available graphical tools. Virt-Manager (Virtual Machine Manager) is the most widely used application for managing KVM based virtual machines, it supports creating, editing, starting, and stopping KVM-based virtual machines, as well as live or cold migration of guest machines between hosts.

### **Prerequisites**

As said earlier, KVM will work only if the CPU has a support of hardware virtualization, either Intel VT or AMD-V. To find whether your CPU supports VT features, run the following command.

```
# egrep '(vmx|svm)' /proc/cpuinfo
[root@server ~]# egrep '(vmx|svm)' /proc/cpuinfo
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mttr pge mca cmov
pat pse36 clflush dts mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm const
ant_tsc arch_perfmon pebs bts nopl xtopology tsc_reliable nonstop_tsc aperfimperf
eagerfpu pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcn
t aes xsave avx f16c rdrand hypervisor lahf_lm arat epb xsaveopt pln pts dtherm
tpr_shadow vnmi ept vpid fsgsbase smep
```

Install KVM on Fedora 26 – Intel VT Support

If the above command returns with any output showing vmx or svm then your hardware supports VT else it does not.

### **Install KVM**

Issue the following command to install latest qemu package and also virt-manager which provides graphical interface to manage virtual machines.

```
# dnf groupinstall "Virtualization"
# dnf install qemu-img libvirt libvirt-python libvirt-client
```

- **qemu-kvm** = QEMU emulator
- **qemu-img** = QEMU disk image manager
- **virt-install** = Command line tool to create virtual machines.
- **libvirt** = Provides libvirtd daemon that manages virtual machines and controls hypervisor.

- **libvirt-client** = provides client side API's for accessing servers and also provides virsh utility which provides command line tool to manage virtual machines.
- **virt-viewer** – Graphical console

## Creating Virtual Machine

Once you have installed KVM and other tools, it is all set to start creating virtual machines. We will look be looking at creating virtual machines in both command line and graphical mode.

### Command Line Mode

virt-install is used to create virtual machines using command line, this command needs multiple inputs from us to create a virtual machine such as CPU, Memory, disk, network, installation media location, os variant and more.

```
# virt-install --name=itzgeekguest --ram=1024 --vcpus=1 --cdrom=/tmp/CentOS-6.5-x86_64-minimal.iso --os-type=linux --os-variant=rhel6 --network bridge=br0 --graphics=spice --disk path=/var/lib/libvirt/images/itzgeekguest.dsk,size=4
```

**Note:** The above command uses bridged networking “br0” for allowing virtual machines to communicate with outside network, you can find a [tutorial on creating bridged networking on Fedora.](#)

**-name** – Name of the Virtual machine

**-ram** – Memory size in MB

**-vcpus** – Virtual cpu's in numbers

**-cdrom** – Location of the ISO image

**-os-type** – OS types like Linux, Windows, and Unix

**-os-variant** – OS variant like rhel 6, solaris

**-network** – Networking

**-graphics** – Guest display settings

**-disk path** – Location of the disk with size of 4 GB

Once you've issued the above command, virt-install will create a virtual machine and starts virt viewer console for OS installation.



Install KVM on Fedora 26 – Guest Installation command Line Virt Viewer

## Graphical Mode

Type the following command in terminal in GUI mode.

```
# virt-manager
```

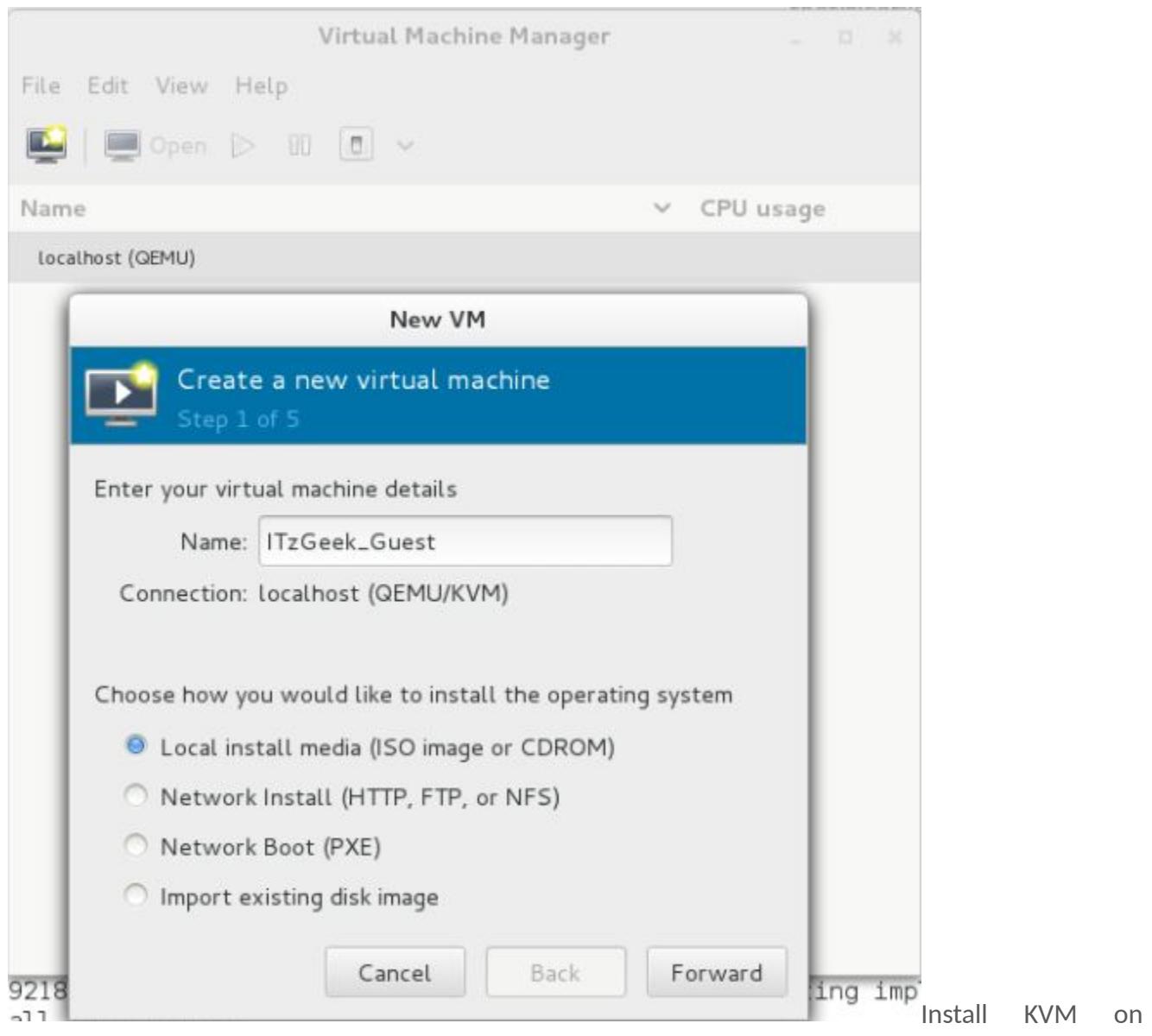
or

**Activities -> Search -> Virtual Machine Manager.**

You may be asked to enter a password if you don't have root access.

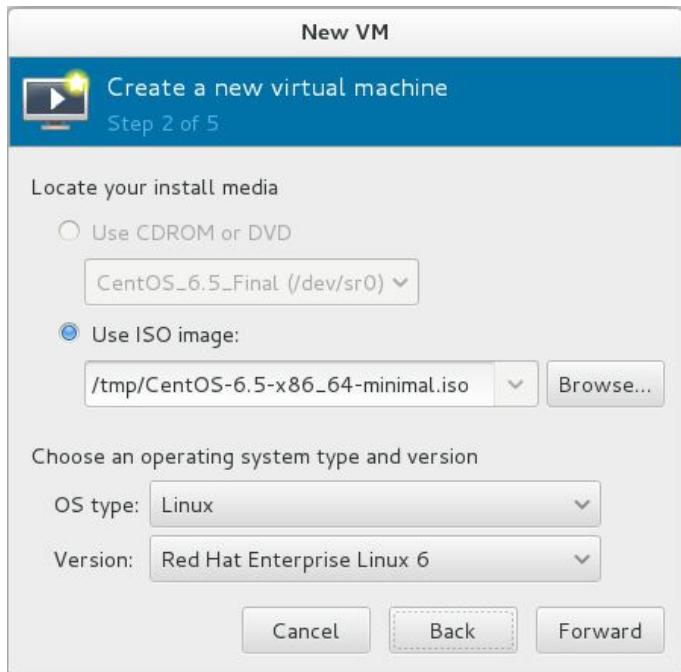
Once it is opened, right-click on localhost(QEMU) and Click on New. Virtual Machine Manager will start a new wizard for creating virtual machine, you will find no difficulty in using this graphical mode.

Step 1: You would require to name your virtual machine, in the same window, you would be asked to choose how would you like to install the operating system. Here i chose to install it from ISO image or CD-ROM.

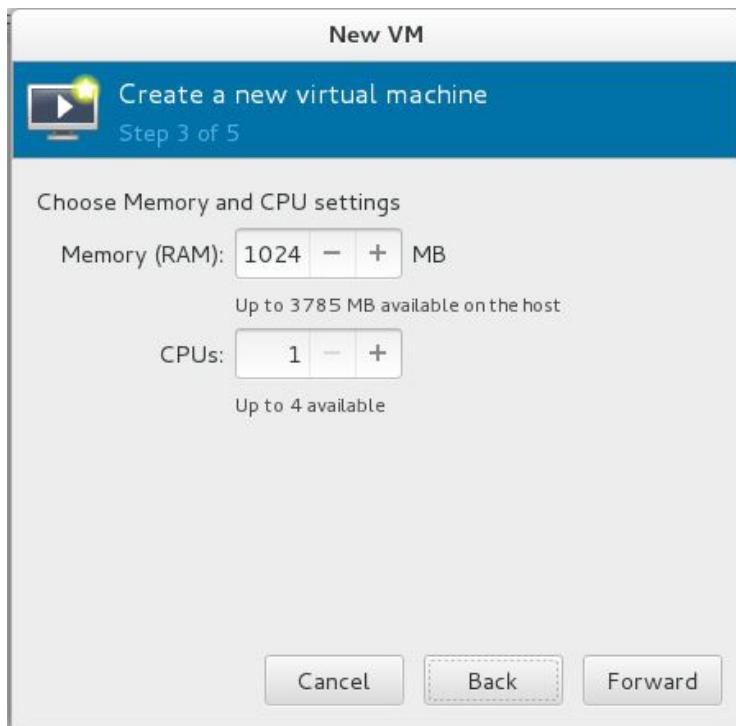


Fedora 26 – Virt Manager – Create VM

Step 2: Enter the location of ISO image or use physical CD-ROM, do not forget to choose operating system type and version.



Install KVM on Fedora 26 – Virt Manager – Installation Media  
Step 3: Configure CPU and memory for the virtual machine.

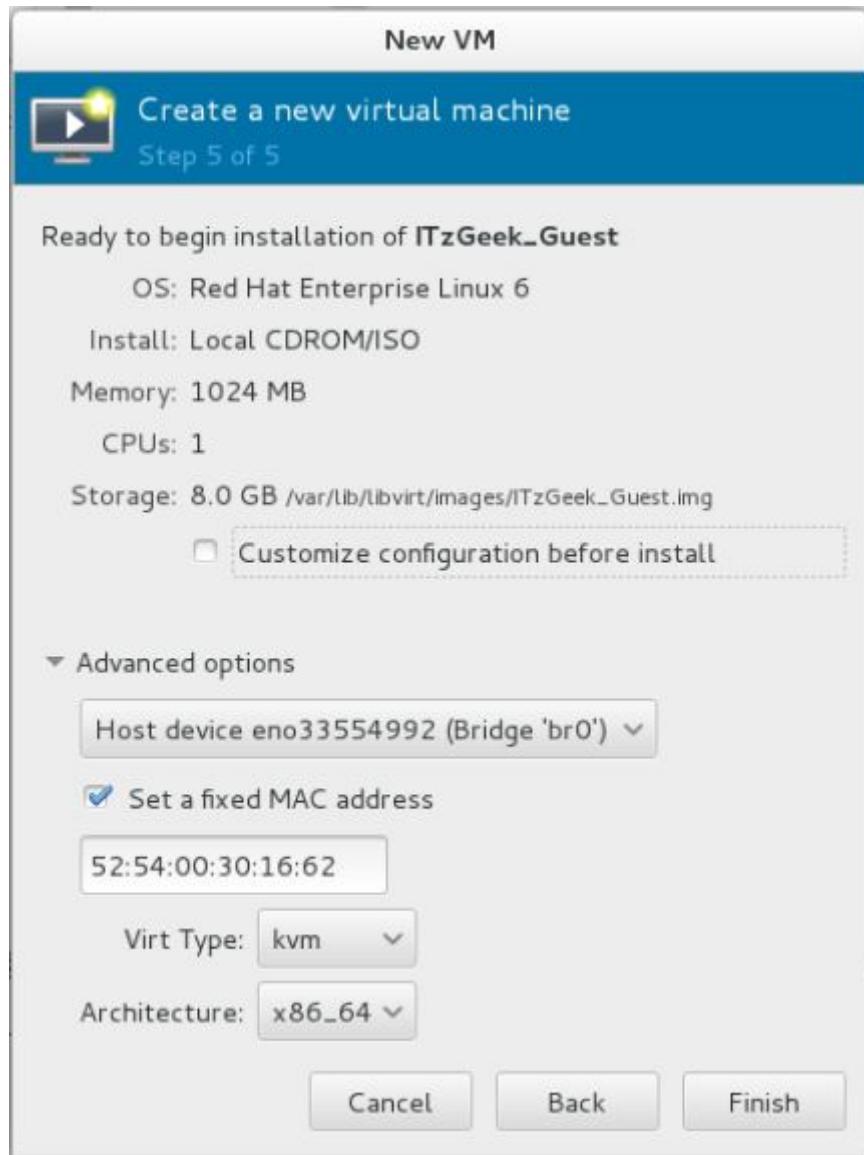


Install KVM on Fedora 26 – Virt Manager – Memory and CPU  
Step 4: You have to mention the amount of storage that you want to assign to a virtual machine, Uncheck on “Allocate entire disk now”, this will prevent from allocating full disk at the VM creation. Disk size will be allocated dynamically depending on the virtual machine usage.



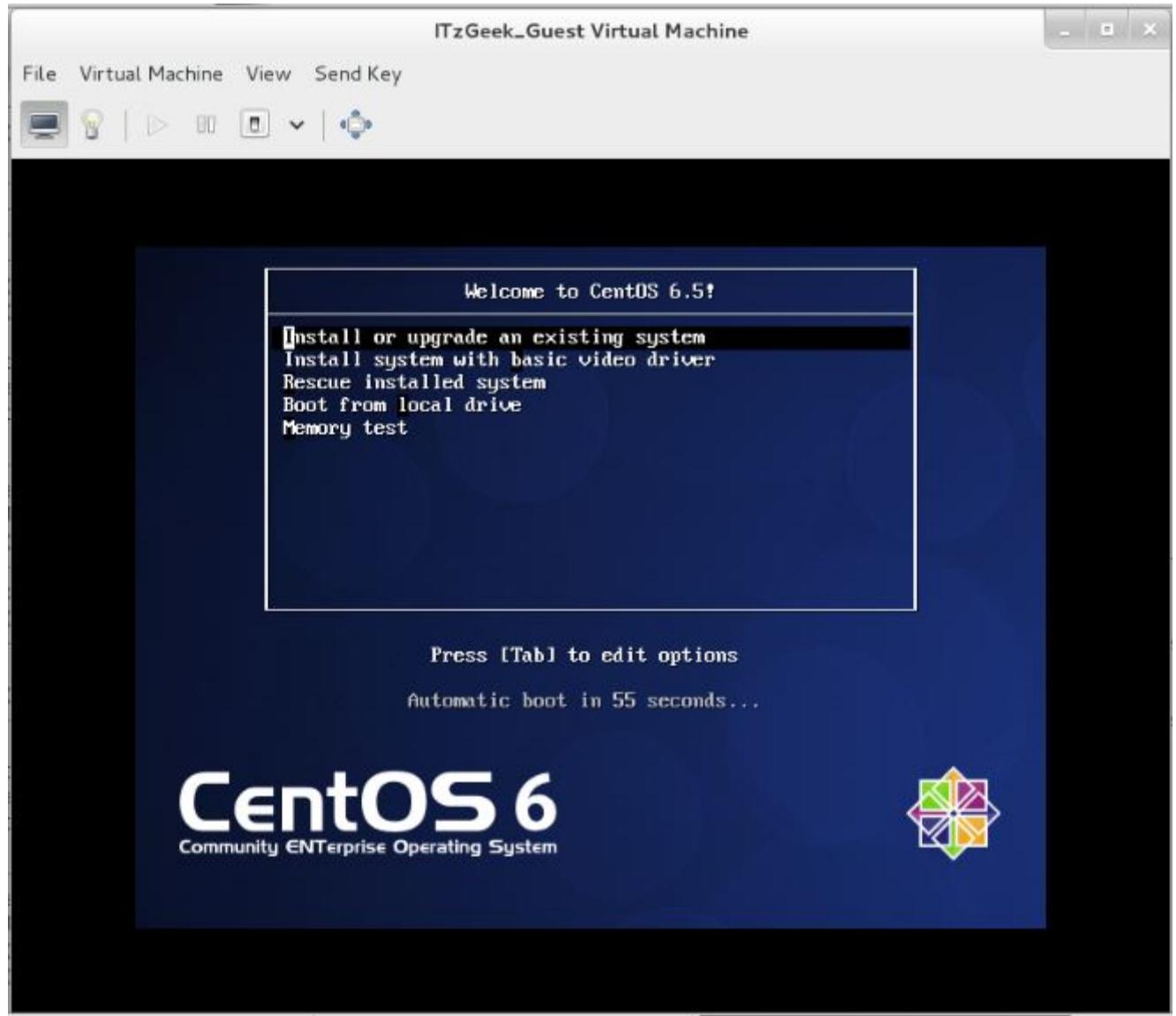
Install KVM on Fedora 26 – Virt Manager – Allocating Storage

Step 5: Summary of all the settings, click on advanced settings to select bridged networking in order to allow virtual machine to communicate outside network. Click on Finish.



#### Install KVM on Fedora 26 – Virt Manager – Summary

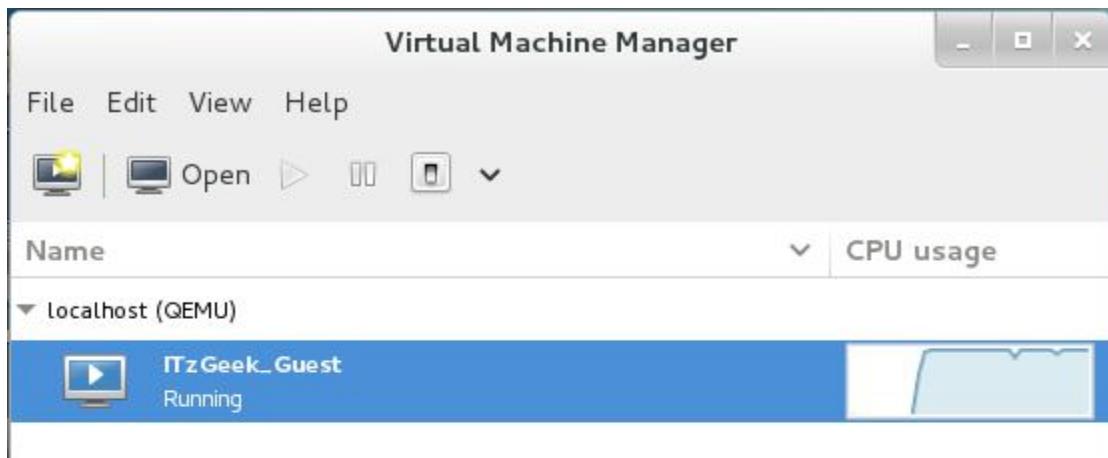
Step 6: KVM will start to create a virtual machine depends on our input, once vm is created. Virt manager will start a console for OS installation. The following screen shows the installation screen of CentOS 6.



Install KVM on Fedora 26 – Virt Manager – VM Console

## Managing Virtual Machine

With the Virtual Machine Manager, you can perform vm's life cycle actions such as start, power off, reset, clone and migration by right clicking on selected virtual machine.



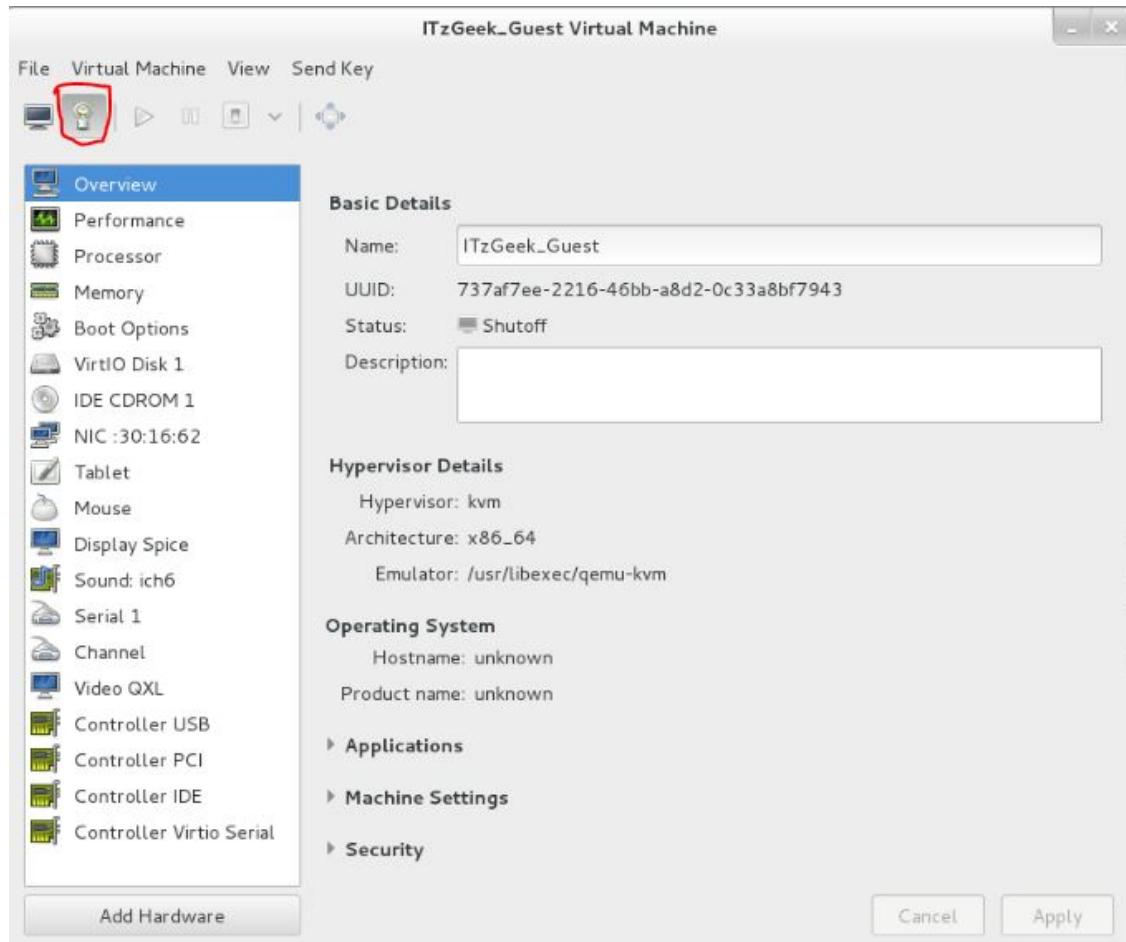
Install KVM on Fedora 26 – Virt Manager

Managing virtual machine using virt manager is very simple, you can just click on “bulp” icon in the vm console. Here you can add, modify and remove devices.

## 12.Observations

1. KVM is a Hosted (type II hypervisor)

## 13. Results:



14.

## Learning Outcomes Achieved

1. Students have used type 2 hypervisor with knowledge of its functionality
2. Installation of hypervisor, creation of Virtual machine with desired configurations
3. Windows (/Ubuntu) operating system was successfully installed
4. Virtual machine was accessed through the LAN address and remote desktop connection

## 15. Conclusion:

1. **Applications of the studied technique in industry**
  - a. Creation of VMs and installation of operating system for production environments
  - b. Creating Virtual Servers and making optimum utilization of hardware
2. **Engineering Relevance**
  - a. Virtualization is the key technology used in cloud computing
3. **Skills Developed**
  - a. Configuration of Hypervisor
  - b. Installation of operating system in the virtual environment

## **References :**

- [1] <https://en.wikipedia.org/wiki/Virtualization>
- [2] "Virtualization in education" (PDF). IBM. October 2007. Retrieved 6 July 2010. A virtual computer is a logical representation of a computer in software. By decoupling the physical hardware from the operating system, virtualization provides more operational flexibility and increases the utilization rate of the underlying physical hardware.
- [3] <http://searchservervirtualization.techtarget.com/definition/hosted-hypervisor-Type-2-hypervisor>

## **Teachers Interaction with Students**

1. Briefing about installation of operating systems
2. Discussion about hosted (Type 1) hypervisor
3. Discussion about x86 Ring structure
4. Discussion about VMM and Hypervisors and their functionality



Hope Foundation's  
**Finolex Academy of Management and Technology, Ratnagiri**  
Information Technology Department

Subject name: Cloud Computing      Subject Code: BEITC702

Class      BE IT      Semester – VIII  
(CBGS)      Academic year: 2018-19

Name of Student      **QUIZ Score :**

Roll No      Assignment/Experiment No.      02

**Title: Installation of Citrix Xen Server and Configuration**

**1. Course objectives applicable COB1.** Get familiar with cloud computing fundamentals, Virtualization, and types of virtualization

**COB3.** Understand various factors for cloud deployment, security aspects at various levels of cloud computing environments.

**2. Course outcomes applicable:**

**CO1 :** Define cloud computing, list various avenues for cloud computing applications, Explain different cloud computing fundamental and virtualization techniques and its role in cloud computing.

**CO3 -**Explain various factors for cloud deployment and key security issues at virtualization as well as network and service level

**3. Learning Objectives:**

1. To understand the concept of type 1 hypervisor
2. Configure type 1 hypervisor and create a virtual machine
3. To Install a Bare Metal Hypervisor (Citrix Xen) and Configure it
4. To assess the performance of Virtual Machine

**4. Practical applications of the assignment/experiment: Hypervisors are main components of Cloud Implementation**

**5. Prerequisites:**

1. Knowledge of Hypervisor and its types
2. Booting from USB Drive
1. At least a 100mbps network card; multiple gigabit suggested

**6. Hardware Requirements:**

1. PC with 4GB RAM, 500GB HDD / Typically a server machine if available
2. Server capable of virtualization
3. Hardware Compatibility List is here: <http://hcl.xenserver.org/>
4. Minimum 2GB ram; 4GB or more recommended to run virtual machines
5. Minimum 1 64bit 1.5GHz cpu; 2GHz or more and multiple CPUs are suggested
6. Hard drive space of at least 16GB; more required if virtual machines will be saved locally
7. At least a 100mbps network card; multiple gigabit suggested

**7. Software Requirements:**

1. Ubuntu / Windows Setup
2. Citrix Xen Setup, XenServer 6.5 ISO : <http://xenserver.org/open-source-virtualization-download.html>

**8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)**

1. What is a Hypervisor?

- |  |
|--|
| 2. What is bare metal hypervisor?  |
| 3. What are security challenges in hypervisor?                               |
| 4. How will you setup a multitenant environment using bare metal hypervisor? |
| 5. What is datacenter virtualization?  |

#### **9. Experiment/Assignment Evaluation:**

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A <b>or</b> any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
<b>Date of performance (DOP)</b>		<b>Total marks obtained</b>	<b>10</b>
<b>Date of checking (DOC)</b>		<b>Signature of teacher</b>	

#### **9. Theory: <Preferably given as handwritten work for students>**

**Bare-metal hypervisor-** A bare-metal hypervisor, also known as a Type 1 hypervisor, is virtualization software that has been installed directly onto the computing hardware.

This type of hypervisor controls not only the hardware, but one or more guest operating systems (OSes). In comparison, a client hypervisor, or Type 2 hypervisor, runs within the host OS, so the underlying hardware is managed by the host OS.

Bare-metal hypervisors feature high availability and resource management; they also provide better performance, scalability and stability because of their direct access to the hardware. On the other hand, the built-in device drivers can limit hardware support.

**Citrix XenServer** is a server virtualization platform based on the Xen hypervisor that allows IT administrators to host, deploy and manage virtual machines.

The main components of Citrix XenServer are the hypervisor, XenCenter integrated management and XenMotion live migration. The platform also delivers free tools for physical-to-virtual and virtual-to-virtual server conversion. It supports up to 500 virtual machines (VMs) and 4,000 virtual CPUs per host. XenServer is designed to work specifically with Citrix Systems' XenApp and XenDesktop tools for desktop and application virtualization.

Prior to XenServer 6.2, IT administrators could download only the base hypervisor for free and deploy as many VMs as they wanted, but the hypervisor lacked many key features such as storage integration and strong load balancing. Version 6.5 extended some of the capabilities from the Enterprise and Platinum editions to the basic edition, but that version is no longer completely free. IT shops have to pay licensing and support fees, which includes 24/7 service, Subscription Advantage upgrade rights and XenCenter automatic updates. Support is not unlimited, however; it is on a per-incident basis.

The latest edition of XenServer features 64-bit architecture that allows it to deliver faster, more efficient VMs than past versions. In addition, it emphasizes scalability and density for VM farms.

#### 10. Precautions :

1. Take backup if you are formatting a working PC
2. If you are installing on a server machine, make sure that an expert is with you as the installation steps may vary
3. Make sure that you have valid license of the software and operating systems that you are going to use

#### 11. Installation Steps / Performance Steps -

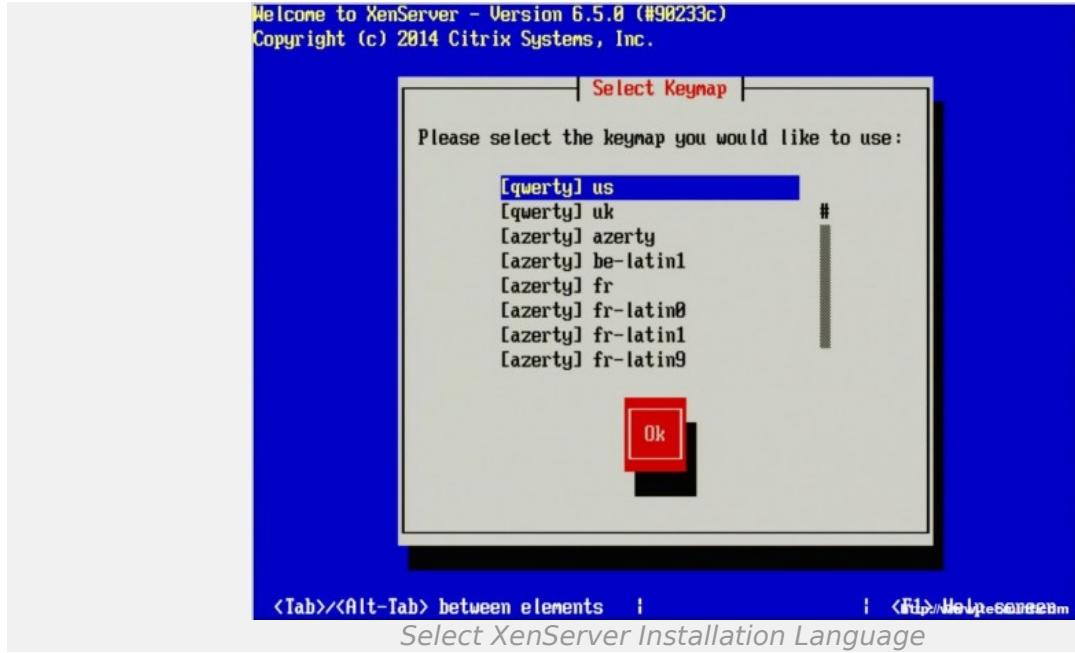
##### Installation of Citrix Xen V 6.5

## Installation of Citrix Xenserver 6.5

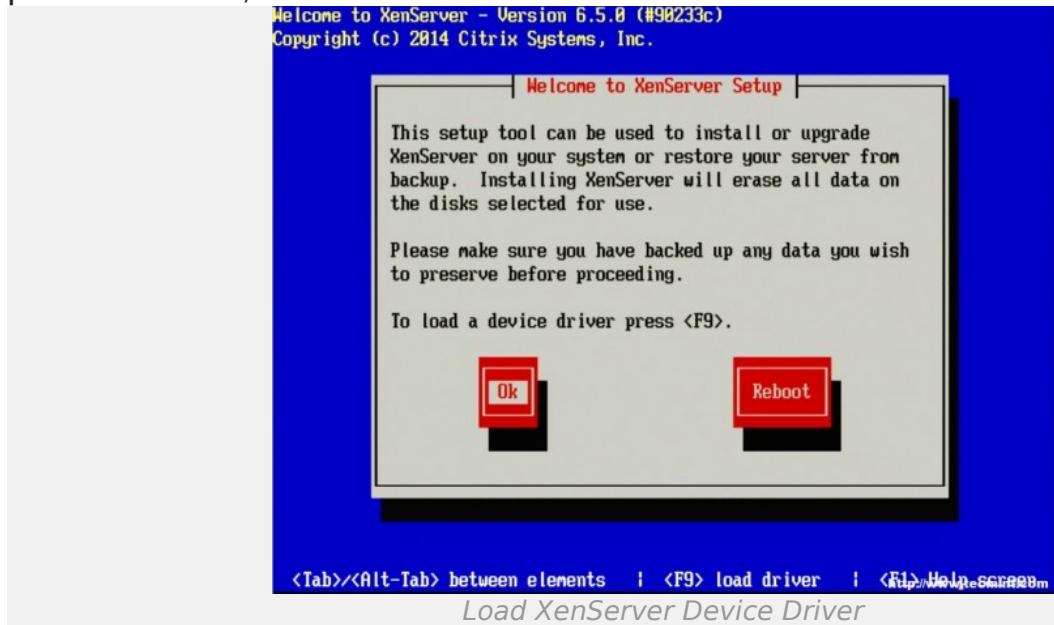
1. The first step in the installation is to download the XenServer ISO file. Make a bootable USB Media, insert the media into the system that XenServer will be installed and boot to that media. Upon successful boot the user should be greeted by the wonderful Citrix XenServer boot splash.



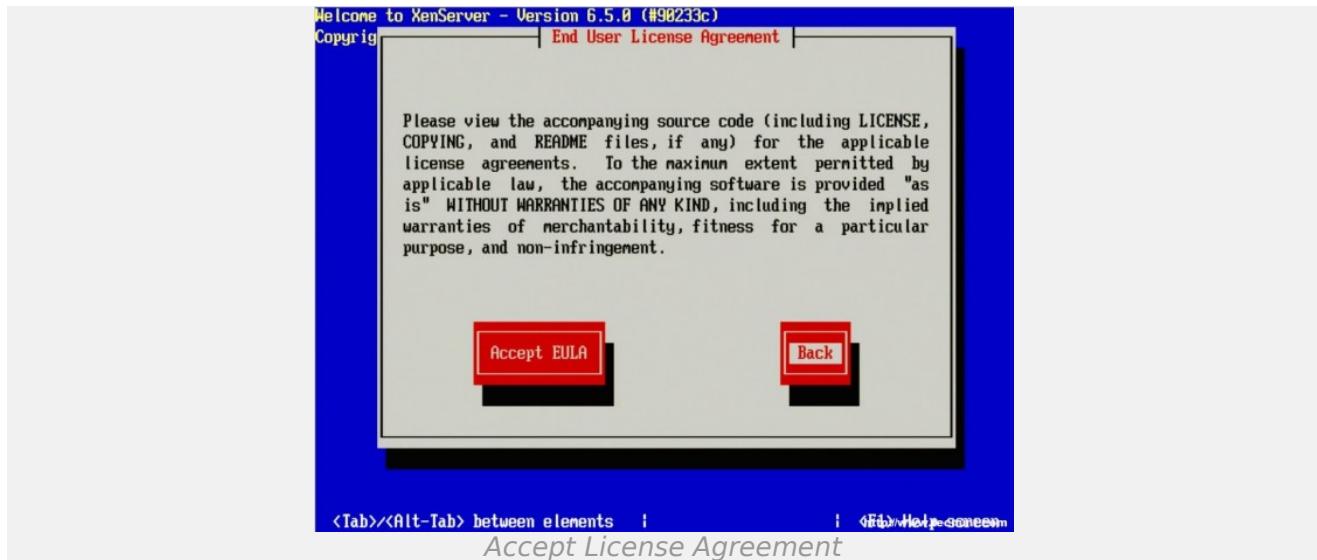
3. At this point simply press enter to begin the booting process. This will boot the user into the XenServer installer. The first screen will ask the user to provide a language selection.



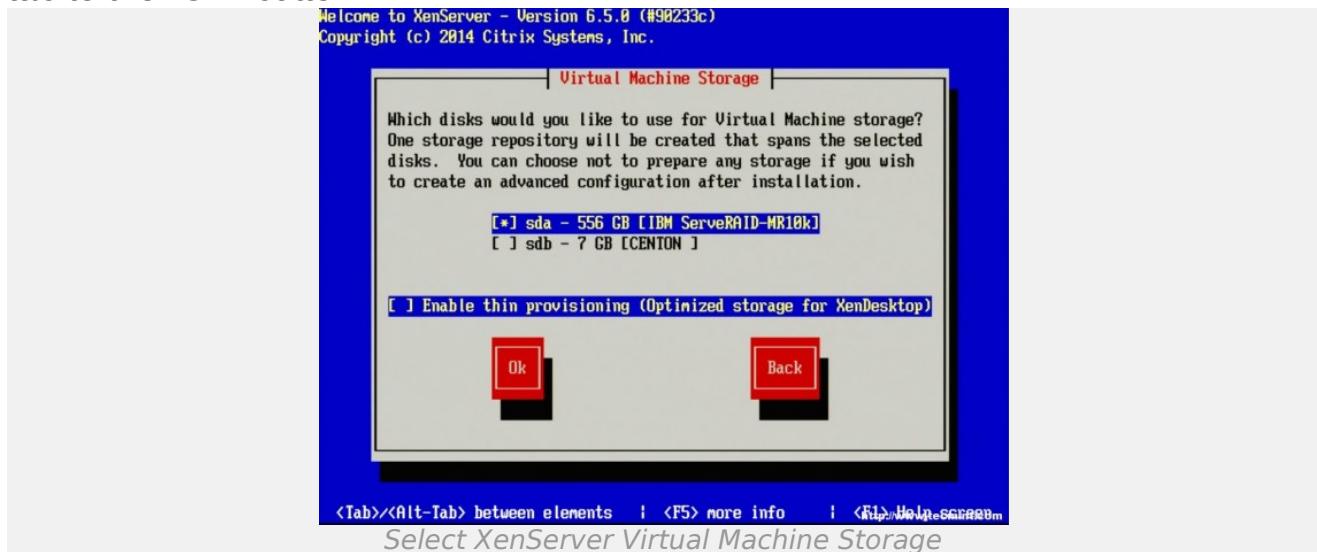
4. The next screen asks the user to confirm the reason for booting to this media as well as provide the option to load extra hardware drivers if needed. In this particular case, it is to install XenServer to the machine so it is safe to click “OK”.



5. The next prompt is the obligatory EULA (End User License Agreement). Feel free to read the whole thing, as you’re supposed to anyways right, otherwise using the keyboard arrows move the cursor over to the “Accept EULA” button and hit enter.



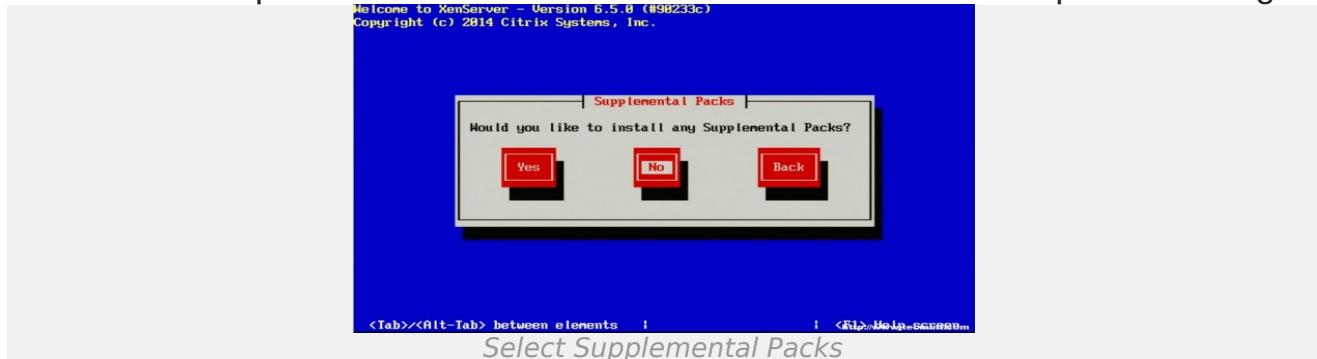
6. The next screen requests the installation device. In this example the RAID setup on the server is where XenServer will be installed. The RAID system is reflected as “sda – 556 GB [IBM ServeRAID-MR10k]” For this guide, thin provisioning is not necessary. Make sure the asterisk ( \* ) character is next to the hard drive selection to install XenServer and tab to the “OK” button.



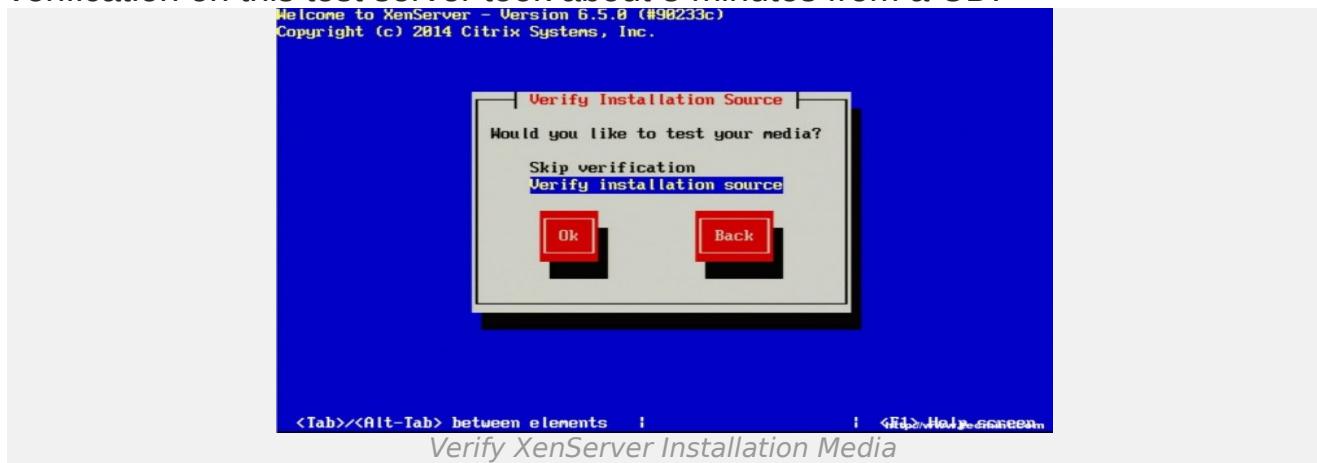
7. The next screen will prompt the user for the location of the installation files. Since the installer was boot locally with a CD/DVD/USB, make sure to select the “Local Media” option.



8. The next step allows for the installation of Supplemental Packs (SP) at the time of install. For this guide, none of the supplemental packs available will be installed at this point but will be covered later once XenServer is up and running.



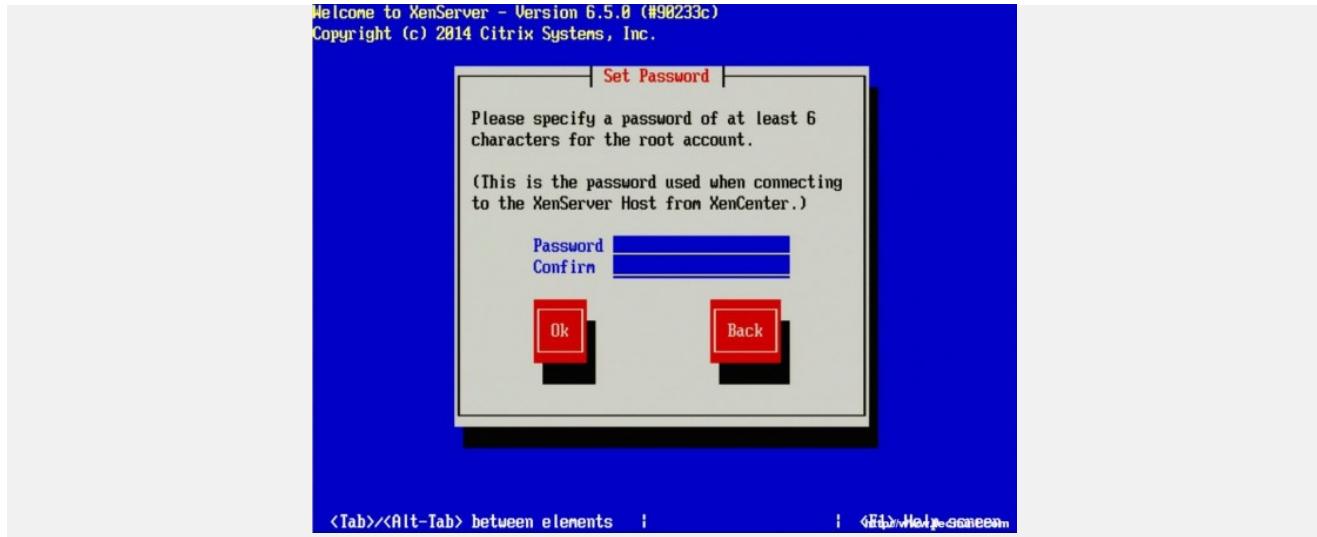
9. The next screen will ask if the user wishes to verify that the installer media is not corrupt. Generally this is a good idea but is a personal choice. All in all the verification on this test server took about 3 minutes from a CD.





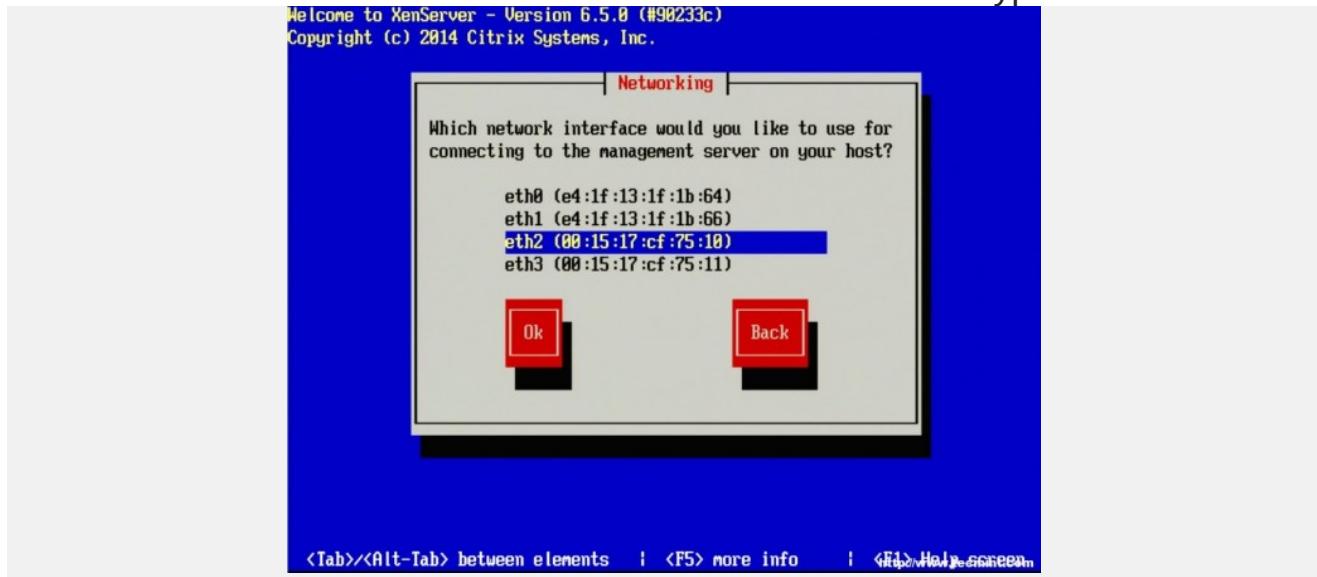
10. Once the verification is completed, if selected during install, the XenServer installer will ask the user to setup some system information. The first prompt will be to set the root user's password. Now, since XenServer will be the underlying system to potentially several important virtualized servers, it is imperative that the password be secured as well as sufficiently complex!

**Important:** Do not forget this password either as there will not be any other users on the system once the installer finishes!



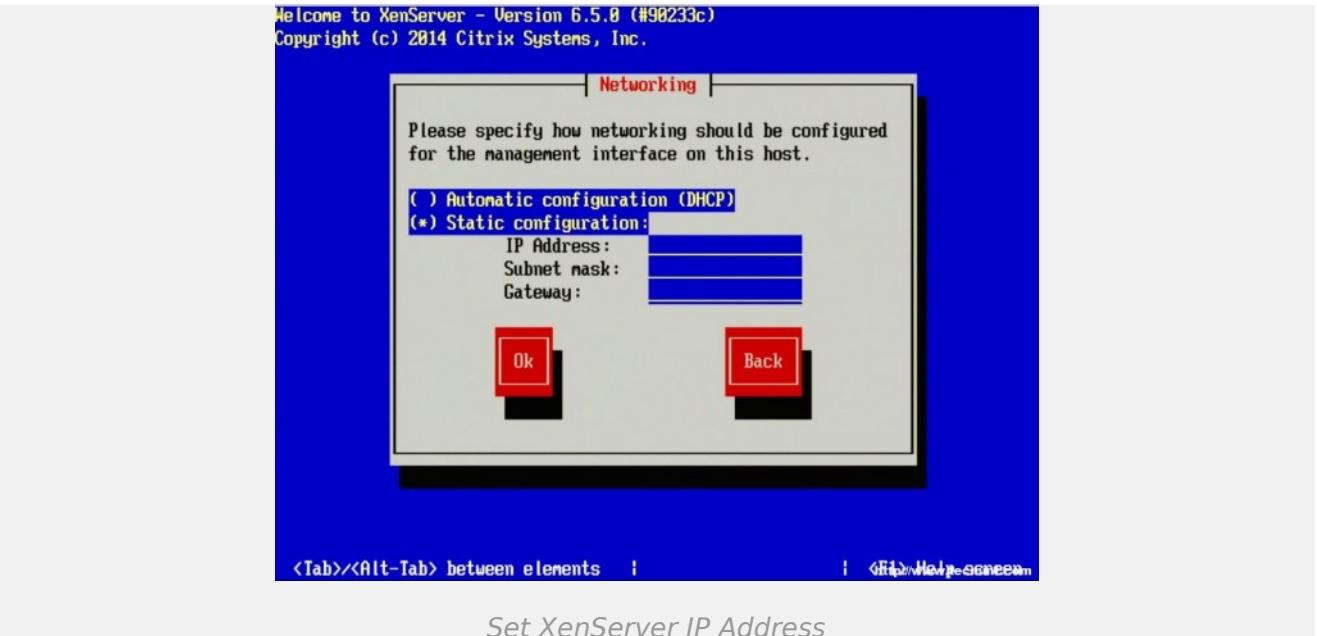
#### *Set XenServer Root Password*

11. The next screen will ask the user to chose a management interface. The number of interfaces that shows up will vary from system to system and the management interface doesn't have to be a separate interface. A separate interface can be used to secure administrative access to the hypervisor itself.



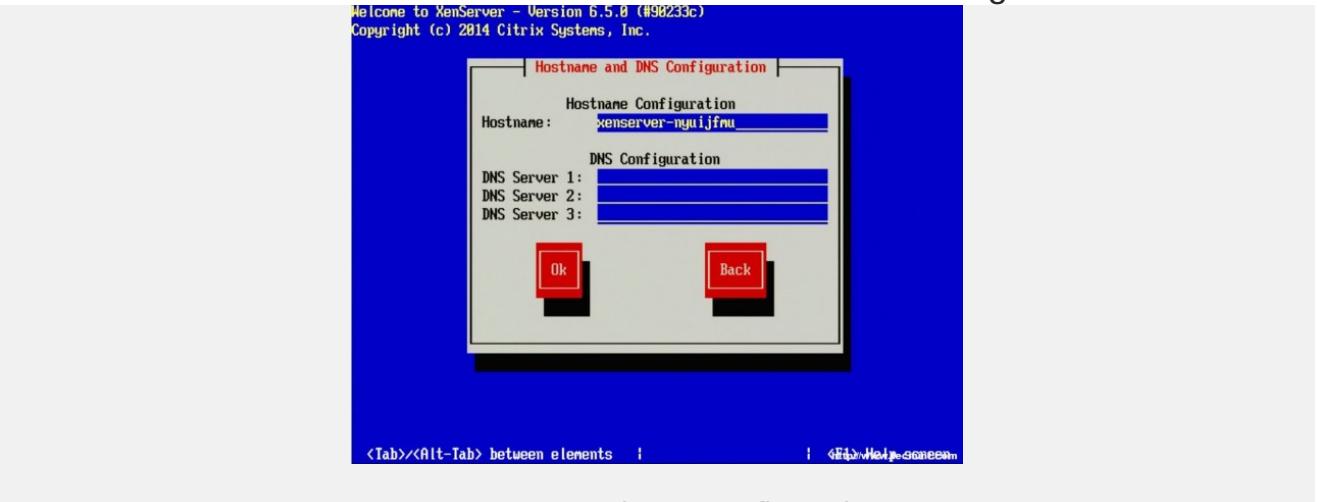
#### *Configure XenServer Network*

12. Once the management interface is selected, the system will prompt for how to obtain an IP address for the management interface. This step will obviously vary from site to site as well.



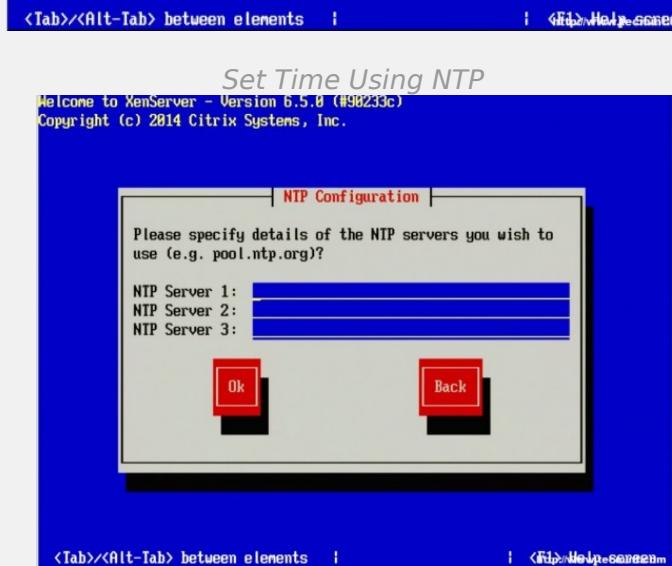
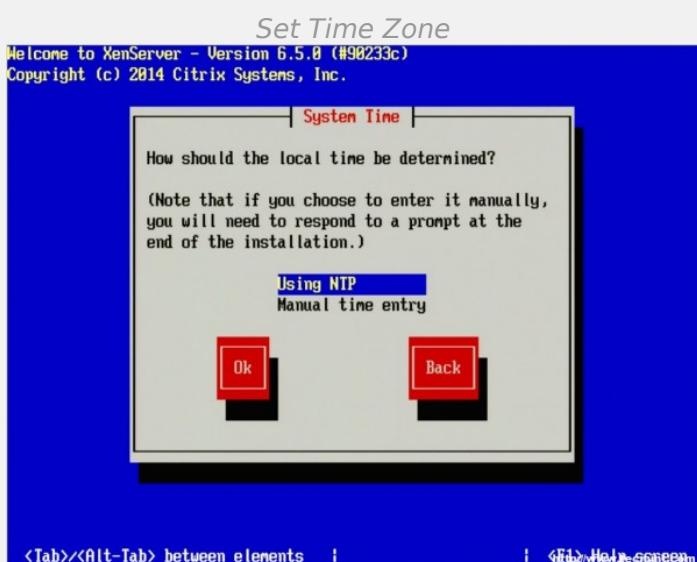
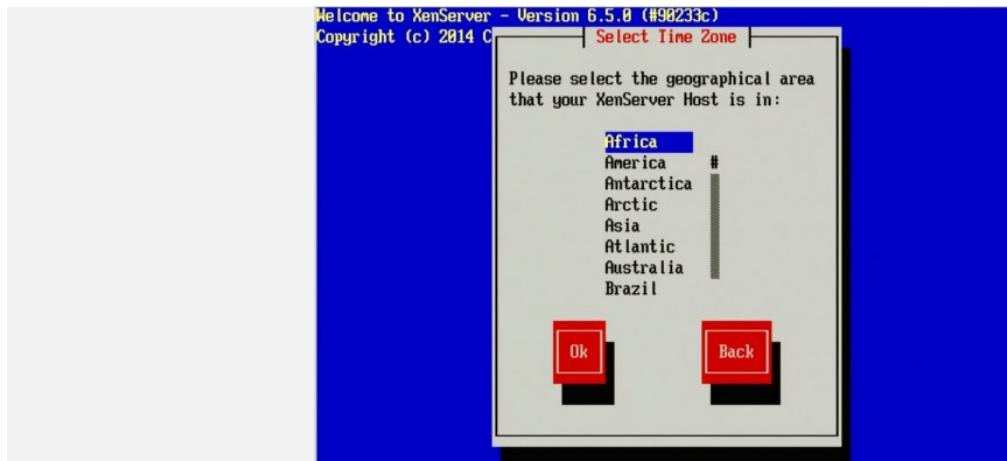
*Set XenServer IP Address*

13. After determining the IP address information for the XenServer, the system will ask for a hostname to be set as well as DNS server configuration.



*Hostname and DNS Configuration*

14. The next three screens will go through the steps of setting up time zones as well as the method for keeping track of time. Generally, Network Time Protocol is suggested for keeping system time synchronized on systems but again this will vary from location to location.  
If a network time server isn't available or the XenServer doesn't have access to the Internet time servers, the installer will allow for manual time setting.



#### Add NTP Servers

15. At this point all of the initial configuration parameters will be complete and the XenServer installer will be ready to begin the installation of the hypervisor.

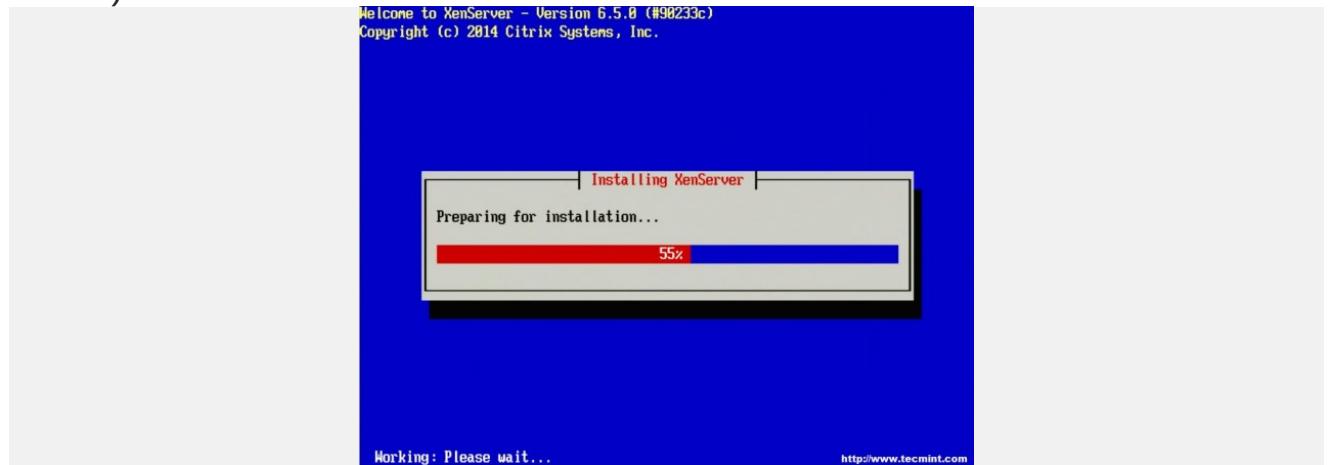
**WARNING** – Continuing at this point WILL ERASE ALL DATA on the target disks!



<Tab>/<Alt-Tab> between elements | | <F1> Help <F11> Screen Saver

*Confirm XenServer Installation Drive*

16. The installation will take some time but a progress bar will be displayed. Once the installer is done, it will prompt the user to restart the system to boot to the newly installed hypervisor (be sure to remove the XenServer installation disk upon reboot).



Working: Please wait...

<http://www.tecmint.com>

*Preparing XenServer Installation*

Welcome to XenServer – Version 6.5.0 (#90233c)  
Copyright (c) 2014 Citrix Systems, Inc.



<Tab>/<Alt-Tab> between elements | | <F1> Help <F11> Screen Saver

*XenServer Installation Complete*

17. Upon removing the installation media and rebooting the system, the user should be presented with the Citrix XenServer splash page. Allow the system to continue booting.



*Citrix XenServer Booting*

18. Once the booting has completed, the system will present the XenServer control page.

*XenServer Configuration Page*

## 12.Observations

1. Xen server was successfully installed , but as the RAM is low and processor support is limited, VM could not be started
2. Server configuration was successful

## 13. Results:



XenServer was successfully installed and is now ready to have any hotfixes/supplemental packs applied, storage repositories designated, and ultimately virtual machines created.

## 14. Learning Outcomes Achieved

1. Students have used type 1 hypervisor with knowledge of its functionality
2. Installation of bare metal hypervisor, creation of Virtual machine with desired configurations
3. Configuration of Bare metal hypervisor is completed and the Hypervisor is accessed on networked environment using Xencenter
4. Virtual machine was accessed through the LAN address and remote Xencenter

## 15. Conclusion:

1. **Applications of the studied technique in industry**
  - a. As an integral part of IaaS setup
  - b. Creation of large scale virtualization environments
2. **Engineering Relevance**
  - a. Bare metal hypervisors are mainly used in datacenter virtualization for optimized use of compute resources
3. **Skills Developed**
  - a. Configuration of Hypervisor
  - b. Installation of bare metal hypervisor

## 16. Experiment/Assignment Evaluation

SR	Parameters	Weight	Excellent	Good	Average	Poor	Not as per requirement
		Scale Factor ->	5	4	3	2	0
1	Technical Understanding	25					
2	Performance / Execution	25					
3	Question Answers	20					
4	Punctuality	20					
5	Presentation	10					
	Total out of 100 -->		$\Sigma (\text{Weight} * \text{Scale Factor}) / 5 = _____$				

## References :

- [1] [https://www.tecmint.com/citrix-xenserver-installation-and-network-configuration-in-linux/2/](https://www.tecmint.com/citrix-xenserver-installation-and-network-configuration-in-linux/)  
<https://en.wikipedia.org/wiki/Virtualization>
- [2] " <https://www.sotechdesign.com.au/xenserver-6-2-install-step-by-step/>

## Teachers Interaction with Students

1. Briefing about installation of Citrix xen systems
2. Demonstration of Citrix Xen hypervisor, configuration and remote access.



Hope Foundation's  
**Finolex Academy of Management and Technology, Ratnagiri**  
Information Technology Department

Subject name: Cloud Computing      Subject Code: BEITC702

Class      BE IT      Semester – VIII  
(CBGS)      Academic year: 2018-19

Name of Student      **QUIZ Score :**

Roll No      Assignment/Experiment No.      03

**Title: OpenStack Configuration and Testing on Platform 9 Sandbox**

**1. Course objectives applicable COB1.** Get familiar with cloud computing fundamentals, Virtualization, and types of virtualization

**COB3.** Understand various factors for cloud deployment, security aspects at various levels of cloud computing environments.

**2. Course outcomes applicable:**

**CO1 :** Define cloud computing, list various avenues for cloud computing applications, Explain different cloud computing fundamental and virtualization techniques and its role in cloud computing.

**CO3 -**Explain various factors for cloud deployment and key security issues at virtualization as well as network and service level

**3. Learning Objectives:**

1. To understand the concept of cloud orchestration software
2. Familiarize with Openstack Dashboard
3. Deploy VM Instances on Openstack
4. Explore the SDN Concept
5. Logon to the remote VMs
6. Monitor and configure remote resources

**4. Practical applications of the assignment/experiment: Hypervisors are main components of Cloud Implementation**

**5. Prerequisites:**

1. Knowledge of Openstack
2. Internet Access
3. Knowledge of CIDR, IP Addressing

**6. Hardware Requirements:**

1. Internet Access with Browser
2. Access to Platform 9 Cloud Sandbox

**7. Software Requirements:**

Browser like Chrome, Internet Explorer Edge

**8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)**

1. What is SDN?
2. List components of Openstack?
3. Which version of Openstack is under development now?

--

#### **9. Experiment/Assignment Evaluation:**

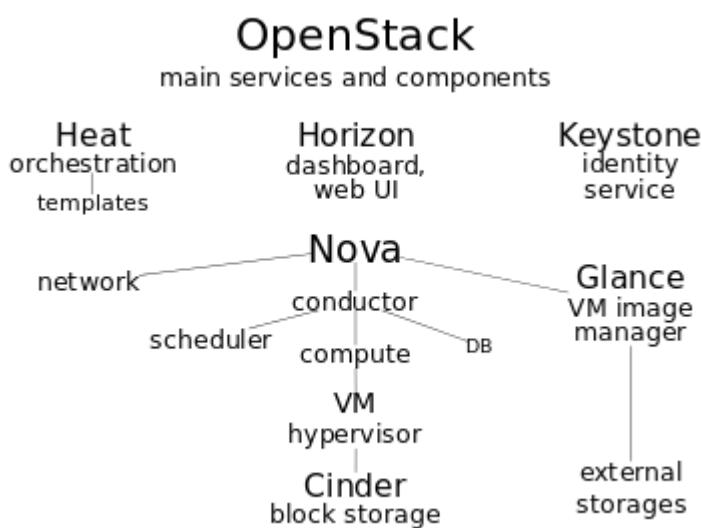
Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A <u>or</u> any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
<b>Date of performance (DOP)</b>		<b>Total marks obtained</b>	<b>10</b>
<b>Date of checking (DOC)</b>		<b>Signature of teacher</b>	

#### **10. Theory: <Preferably given as handwritten work for students>**

OpenStack is a free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service (IaaS), whereby virtual servers and other resources are made available to customers.[1] The software platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout a data center. Users either manage it through a web-based dashboard, through command-line tools, or through RESTful web services.

OpenStack began in 2010 as a joint project of Rackspace Hosting and NASA. As of 2016, it is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012[4] to promote OpenStack software and its community.[5] More than 500 companies have joined the project.

Openstack Components are as follows:



As the OpenStack project has matured, vendors have pioneered multiple ways for customers to deploy OpenStack:

- OpenStack-based Public Cloud: A vendor provides a public cloud computing system based on the OpenStack project.
- On-premises distribution: In this model, a customer downloads and installs an OpenStack distribution within their internal network. See Distributions.
- Hosted OpenStack Private Cloud: A vendor hosts an OpenStack-based private cloud: including the underlying hardware and the OpenStack software.
- OpenStack-as-a-Service: A vendor hosts OpenStack management software (without any hardware) as a service. Customers sign up for the service and pair it with their internal servers, storage and networks to get a fully operational private cloud.
- Appliance based OpenStack: Nebula was a vendor that sold appliances that could be plugged into a network which spawned an OpenStack deployment.

OpenStack software controls large pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API. OpenStack works with popular enterprise and open source technologies making it ideal for heterogeneous infrastructure.

## **10. Precautions :**

1. Select the address range carefully
2. Use Private address space e.g. 192.168.0.1 –254

## **11. Installation Steps / Performance Steps –**

1. Logon to Platform 9 Cloud
2. Open the Dashboard
3. Create Virtual Machine from Existing Images
4. Explore the Size of Compute give to you
5. Explore the network
6. Comment on the Orchestration provided by OpenStack

## **12.Observations**

1. Total 1 VMs with 4GB RAM and CentOS were deployed successfully
2. SDN with one Router with public IP and networking of 3VMs was implemented successfully using Neutron component of Openstack

## 13. Results:

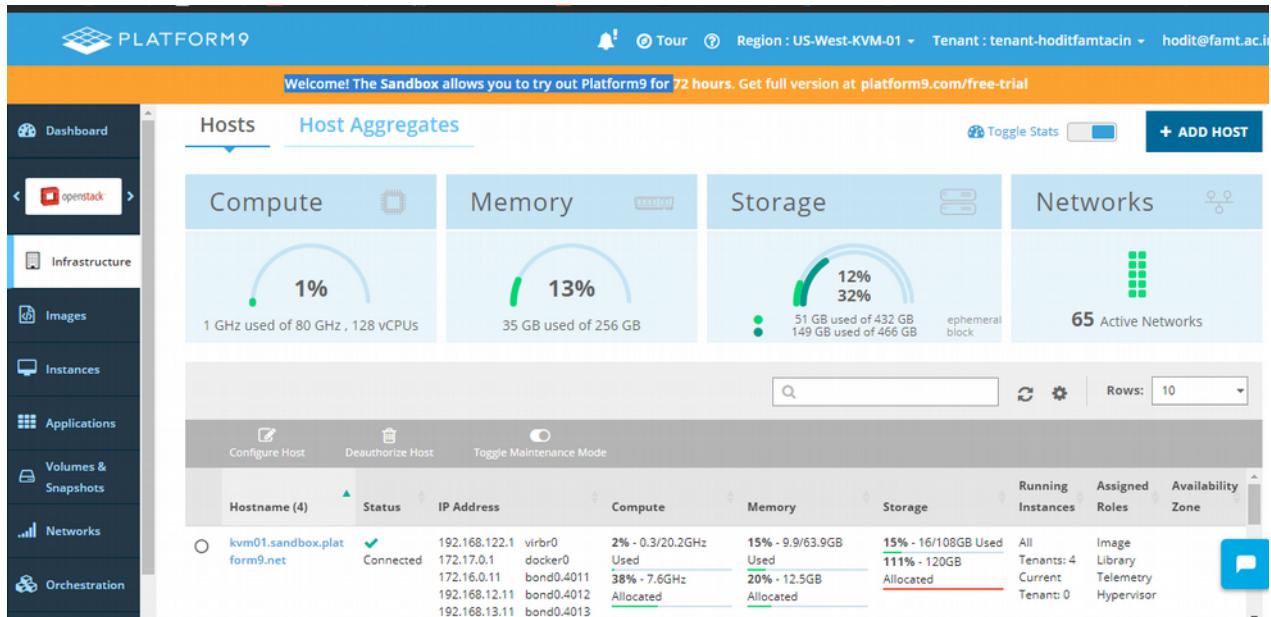


Fig 1: Snapshot of Dashboard Showing the Compute Resources

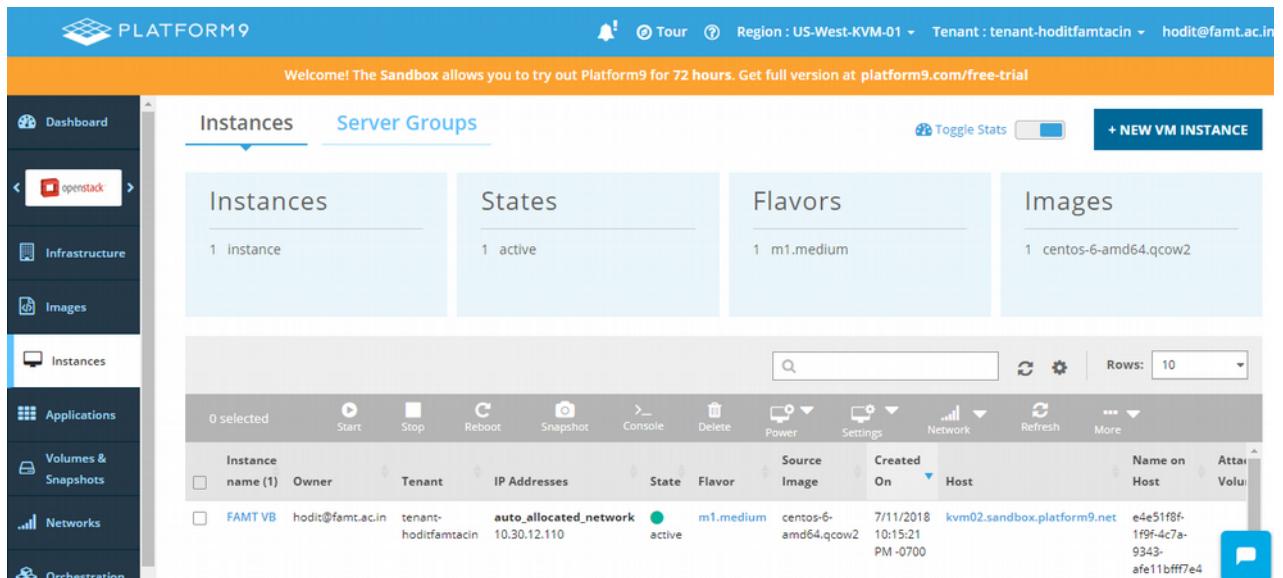


Fig 2: Snapshot Virtual Machine Status

### Overview

OpenStack ID : e4e51f8f-1f9f-4c7a-9343-afe11bfff7e4

Status : ACTIVE

Created : Jul 11, 2018 10:15:21 PM

VM State : active

Task State : n/a

Key Pair : n/a

Security Groups :default

#### Specs

Flavor : m1.medium

RAM : 4096 MB

CPU : 1000 MHz

VCPUs :2 VCPUs

Disk :40 GB

## Networks

Name	IP Address	MAC Address	Type
auto_allocated_network	10.30.12.110	fa:16:3e:65:a2:d2	fixed

## Host : kvm02.sandbox.platform9.net

The screenshot shows the OpenStack dashboard interface. On the left, there is a sidebar with navigation links: Dashboard, Infrastructure, Images, Instances, Applications, Volumes & Snapshots, Networks (which is currently selected), and Orchestration. The main content area is titled "Subnet". It contains the following configuration details:

Setting	Value
Shared	false
Port Security	true
Name	IT1
Network Address (CIDR)	192.168.0.0/24
IP Version	4
Gateway IP	192.168.0.1
Enable DHCP	true
Allocation Pools	192.168.0.7 to 192.168.0.200
DNS Name Servers	192.168.0.3
Host Routes	(empty)

At the bottom of the main content area, there are two buttons: "BACK" and "CREATE".

Name	Subnets Associated	Tenant	Shared	Security	External Network	Admin State	Status
auto_allocated_network	auto_allocated_subnet_v4	tenant-hoditfamtacin	No	Enabled	No	Up	ACTIVE
Public Internet	FIP-Agents-01, Public-Subnet01, Public-Subnet02, Public-Subnet03, Public-Subnet04	service	Yes	Enabled	Yes	Up	ACTIVE

**Fig 3: Snapshot Network Status**

## 14. Learning Outcomes Achieved

1. Openstack was used as cloud orchestration software
2. Students created VMs using Openstack Dashboard
3. VMs were successfully deployed and IP addresses were assigned
4. SDN with one router with public IP and 3 VMs network was created
5. Remote VM access was made using console
6. Overall instance monitoring and configuration was done using dashboard

## 15. Conclusion:

1. **Applications of the studied technique in industry**
  - a. As an integral part of IaaS at Enterprise level
  - b. Main technology for cloud orchestration
2. **Engineering Relevance**
  - a. Openstack is one of the widely used software for setup of Private clouds and cloud management
  - b. Openstack is Open source software with large community support
3. **Skills Developed**
  - a. Using Openstack Dashboard for performance monitoring and VM configuration
  - b. SDN Creation and access

## **References :**

- [1] Ocata Release Notes — OpenStack". Wiki.openstack.org. Retrieved 22 February 2017.  
Jump up ^ "ReleaseAnnouncement/Ocata — OpenStack". www.openstack.org. Retrieved 22 February 2017.
- [2] "OpenStack Open Source Cloud Computing Software". Retrieved 29 November 2013.
- [3] "OpenStack Launches as Independent Foundation, Begins Work Protecting, Empowering and Promoting OpenStack". BusinessWire. 19 September 2012. Retrieved 7 January 2013.

## **Teachers Interaction with Students**

1. Concept of Software defined network with demonstration
2. Openstack Dashboard for VM Monitoring
3. Remote access of VM



Hope Foundation's  
**Finolex Academy of Management and Technology, Ratnagiri**  
Information Technology Department

Subject name: Cloud Computing Subject Code: BEITC702

Class BE IT Semester – VIII (CBGS) Academic year: 2018-19

Name of Student **QUIZ Score :**

Roll No Assignment/Experiment No. 04

**Title: Azure Service Bus Configuration / RabbitMQ Configuration**

**1. Course objectives applicable COB2.**Understand benefits of various cloud services and Handle Open Source Cloud Implementation and Administration  
**COB6.**Understand AAA administration and mobile cloud computing

**2. Course outcomes applicable:**

**CO2** -List benefits of different cloud services, types of services, Concept of Anything as A Service. Handle Open Source Cloud Implementation and Administration using OpenStack or Eucalyptus.

**CO6**-Explain cloud administration process, model for secure access for cloud and mobile cloud computing and its future implications

**3. Learning Objectives:**

1. To understand concept of message queue
2. Configure Message Queuing Server
3. Send Message and consume message on the queue
4. Understand the role of message queue in SaaS model of cloud

**4. Practical applications of the assignment/experiment: Message Queues are used by cloud based application to communicate with each other**

**5. Prerequisites:**

1. Knowledge of Cloud Application Architecture
2. Internet Access
3. Knowledge of Azure cloud, Service Bus

**6. Hardware Requirements:**

1. Internet Access with Browser
2. Access to Microsoft Azure Cloud Subscription

**7. Software Requirements:**

Browser like Chrome, Internet Explorer Edge , RabbitMQ Server, Visual Studio 2017

**8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)**

1. What is Service Bus?
2. Which version of Visual Studio is used?
3. What is an Access Key?

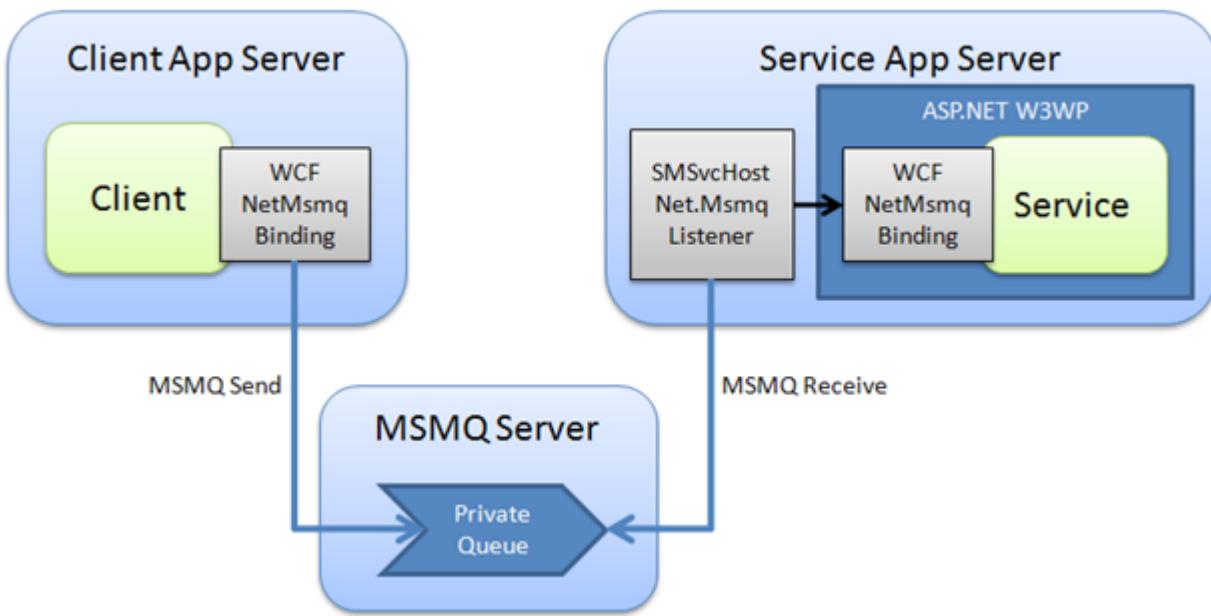
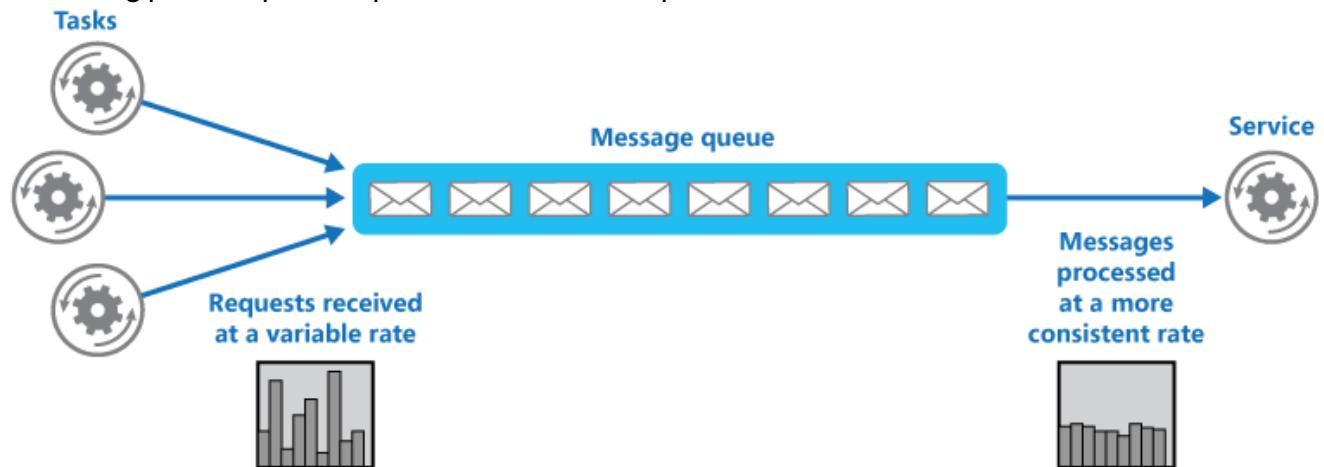
---

**9. Experiment/Assignment Evaluation:**

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A <b>or</b> any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
<b>Date of performance (DOP)</b>		<b>Total marks obtained</b>	<b>10</b>
<b>Date of checking (DOC)</b>		<b>Signature of teacher</b>	

## 1. Theory: <Preferably given as handwritten work for students>

A message queueing service is a message-oriented middleware or MOM deployed in a compute cloud using software as a service model. Service subscribers access queues and/or topics to exchange data using point-to-point or publish and subscribe patterns.



A message queueing service aims to eliminate the traditional overhead associated with operating in-house messaging infrastructures. These operating overheads include:

- Unused capacity installed to meet peak demands
- Human resources that are necessary to maintain messaging infrastructure
- Projects idle time waiting for resource provisioning
- Need to isolate messaging resources

Besides reducing cost, a message queueing service seeks to simplify access to messaging resources and therefore facilitate integration efforts within organizations and between them.

A message queueing service also creates new value by providing reduced costs, enhanced performance and reliability. In order to provide those benefits, a message queueing service leverages cloud computing resources such as storage, network, memory and processing capacity. By using virtually unlimited cloud computing resources, a message queueing service provides an internet scale messaging platform.

## **10. Precautions :**

1. Configure the RabbitMQ server first, Create queue and then send the messages

## **11. Installation Steps / Performance Steps -**

### **Introduction**

The Azure Service Bus Queue is different than the Azure Storage Queues. In my last [article](#), we have seen how to get started with Azure Service Bus by creating a namespace. This article tells you how to work with Azure Service Bus Queues with a sample console application.

### **Content**

- Service Bus Queues
- Create Service Bus Queues in Azure portal
- Send and receive a message using service bus Queues

### **Pre-request**

- Visual Studio 2017 update 3 or later
- Azure Subscription

### **Service Bus Queues**

- Service Bus Queues provide a queueing mechanism
- Message will appear only once
- Message is processed using FIFO(First In First out) pattern
- Support transactions

The message lock can be renewed, It consists of few major parts,

1. Body – The body can be serialized object or a stream
2. Label – Simple text label

3. Time to Live – How long the message is stored in queue
4. Properties – Dictionary of properties that can be used by your specific consumer

## Create a service bus

### Step 1

Log in to the Azure portal (<http://portal.azure.com>), using your Azure account.

### Step 2

Go to the service bus namespace (refer my previous article to create a namespace) which is already created.

NAME	STATUS	MAX SIZE	ENABLE PARTITIONING
azuresb	Active	16 GB	true

Figure 1: Service Bus Namespace

### Step 3

Click on Queues in Entity and create a queue as shown in below figure.

Home > FAMTCC2018 - Queues > azuresb

**azuresb**  
Service Bus Queue

Search (Ctrl+ /)

**Overview**

**Diagnose and solve problems**

**SETTINGS**

Shared access policies

Metrics (preview)

Properties

Locks

Automation script

**SUPPORT + TROUBLESHOOTING**

New support request

Delete

**Essentials**

Namespace: FAMTCC2018

Queue URL: <https://FAMTCC20>

ACTIVE MESSAGE COUNT: 0 MESSAGES

MAX SIZE: 16 GB

SCHEDULED MESSAGE COUNT: 0 MESSAGES

CURRENT: 0.0 KB

DEAD-LETTER MESSAGE COUNT: 0 MESSAGES

TRANSFER MESSAGE COUNT: 0 MESSAGES

TRANSFER DEAD-LETTER MESSAGE COUNT: 0 MESSAGES

100% FREE SPACE



**Figure 2**

## Create queue

Service Bus

\* Name i

msgqueue



Max queue size

1 GB



Message time to live i

Days

Hours

Minutes

Seconds

14

0

0

0

Lock duration i

Days

Hours

Minutes

Seconds

0

0

0

30

Enable duplicate detection i

Enable dead lettering on message expiration i

Enable sessions i

Enable partitioning i

**Create**

**Figure 3**

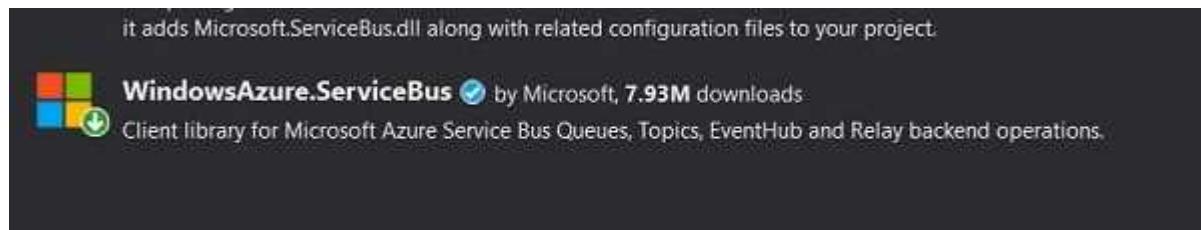
Give a name for the queue, in my case I named it as azuresb.

Message time to live – by default it will be 14 days, it means how long the message will be stored in queue. Keep the rest of the option to default setting and click on create

**Send and receive a message using service bus Queues**

Create a new console application project in Visual Studio 2017.

Download the service bus queues messaging package from NuGet.



**Figure 4: Nuget**

Write the below function in Program.cs file.

```
1. static async Task MainAsync()
2. {
3.
4.     const string connectionString = "<Namespace connection String>"; // get it
   from azure portal from service bus namespace shared
   //access policy
5.     const string queueName = "msgqueue";
6.     var _client = QueueClient.CreateFromConnectionString(connectionString, que
ueName);
7.     string Message = "I'm in Azure Service Bus Queue";
8.     BrokeredMessage message = new BrokeredMessage(Message);
9.     await _client.SendAsync(message);
10.
11. }
```

Queue client is an abstract class, where the function CreateFromConnectionString (string, string) is used to create a new copy of Queue Client from the connection string with specified queue path.

BrokeredMessage is a sealed class; here, it is used to initialize the new instance and to serialize the message which is sent to the Queue with the help of queue client. From the above code, it is obvious that the message with string type is serialized and send to queue using Queue Client.

### Check the stats in Azure – Shown in Results

### Receive a Message

```
1. static void GetMessage()
```

```

2.    {
3.        const string connectionString = "<Namespcae connection string>";
4.        const string queueName = "msgqueue";
5.        var queueClient = QueueClient.CreateFromConnectionString(connectionString, queueName);
6.        BrokeredMessage message = queueClient.Receive();
7.        string body = message.GetBody<string>();
8.        message.Complete();
9.        message.Abandon();
10.       Console.WriteLine(body);
11.       Console.ReadLine();
12.
13.    }

```

The above function is used to receive the message from the queue using the QueueClient and get the body of the message using the brokeredMessage.

### Program.cs

```

1. using System;
2. using System.Threading.Tasks;
3. using Microsoft.ServiceBus.Messaging;
4.
5. namespace ServiceFabricDemo
6. {
7.     class Program
8.     {
9.         static void Main(string[] args)
10.        {
11.            MainAsync().GetAwaiter().GetResult();
12.            GetMessage();
13.        }

```

```

14.
15.     static async Task MainAsync()
16.     {
17.
18.         const string connectionString = "Endpoint=sb://msgdemobus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=rJ0AZa4UFgpbUN8fGL7eUJYSLfiwtlvP4mnPAXPSu68=";
19.         const string queueName = "msgqueue";
20.         var _client = QueueClient.CreateFromConnectionString(connectionString, queueName);
21.         string Message = "I'm in Azure Service Bus Queue";
22.         BrokeredMessage message = new BrokeredMessage(Message);
23.         await _client.SendAsync(message);
24.
25.     }
26.     static void GetMessage()
27.     {
28.         const string connectionString = "Endpoint=sb://msgdemobus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=rJ0AZa4UFgpbUN8fGL7eUJYSLfiwtlvP4mnPAXPSu68=";
29.         const string queueName = "msgqueue";
30.         var queueClient = QueueClient.CreateFromConnectionString(connectionString, queueName);
31.         BrokeredMessage message = queueClient.Receive();
32.         string body = message.GetBody<string>();
33.         message.Complete();
34.         message.Abandon();
35.         Console.WriteLine(body);
36.         Console.ReadLine();
37.
38.     }
39.

```

```
40. }
```

```
41. }
```

## Output

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation menu includes 'Create a resource', 'All services' (selected), 'Dashboard', 'Resource groups', 'App Services', 'Function Apps', 'SQL databases', 'Azure Cosmos DB', and 'Virtual machines'. The main content area is titled 'azuresb' under 'Service Bus Queue'. The 'Overview' tab is selected. It displays the namespace 'FAMTCC2018' and the 'Queue URL' <https://FAMTCC2018.servicebus.windows.net/>. A large circular chart indicates '100% FREE SPACE'. Below the chart, various metrics are listed:

Metric	Value
ACTIVE MESSAGE COUNT	0 MESSAGES
SCHEDULED MESSAGE COUNT	0 MESSAGES
DEAD-LETTER MESSAGE COUNT	0 MESSAGES
TRANSFER MESSAGE COUNT	0 MESSAGES
TRANSFER DEAD-LETTER MESSAGE COUNT	0 MESSAGES
MAX SIZE	16 GB
CURRENT	0.0 KB

Fig1. Screenshot Showing 0 Messages

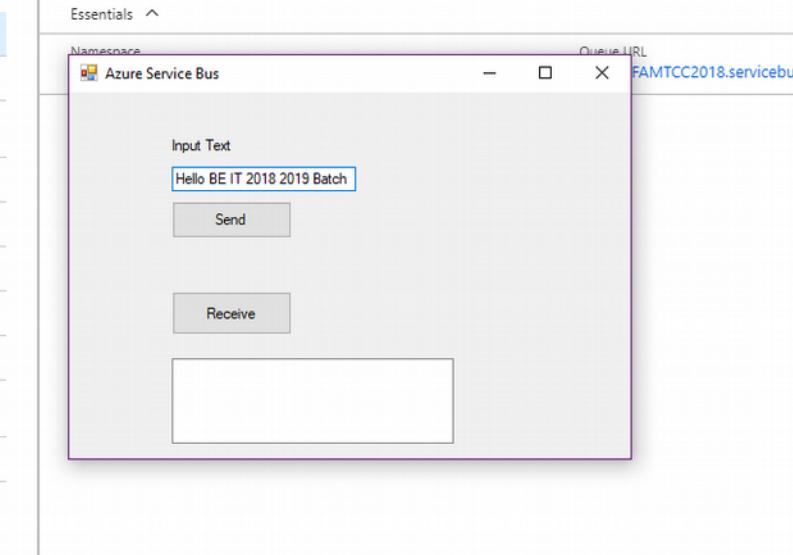
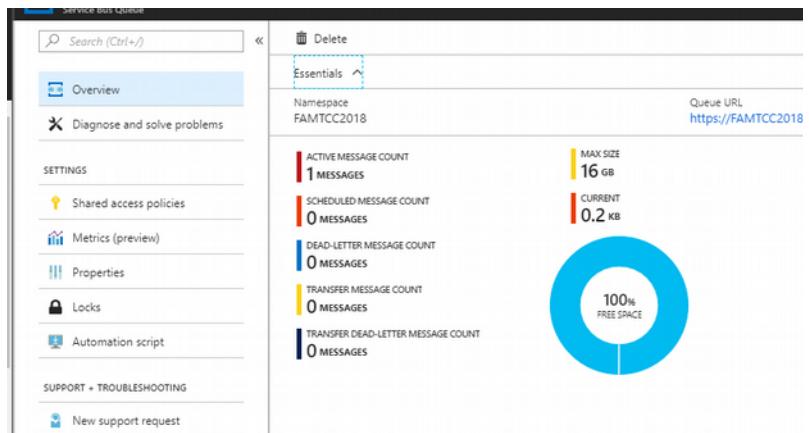
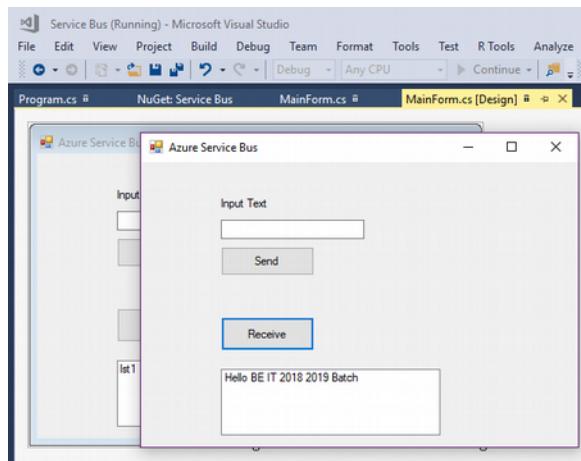


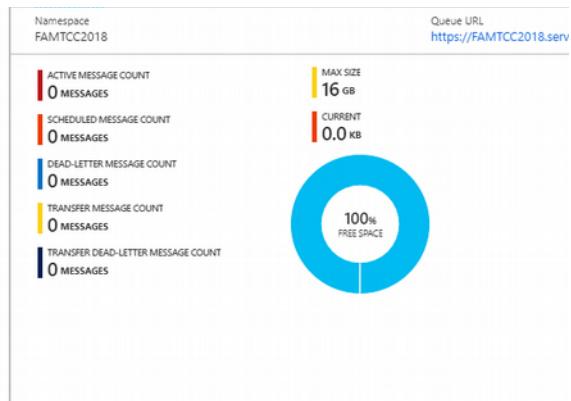
Fig 2. Sending Message on Azure Queue



**Fig3. Screenshot Showing 1 Messages**



**Fig 4. Message received from Queue as shown above**



**Figure 5 : Message received from Queue – Showing 0 remaining messages**

## **14. Learning Outcomes Achieved**

1. Students Configured Azure Cloud Account and Queuing Server
2. Students Configured Azure Message Queue
3. Message was sent using producer program code and consumed by the receiver
4. Discussion about how SaaS Web services transfer messages using queues

## **15. Conclusion:**

- 1. Applications of the studied technique in industry**
  - a. Message Queues are used by Web services for communication purpose
  - b. For invoking web services as well as passing results message queues are used
- 2. Engineering Relevance**
  - a. SaaS Model is the main consumer of Message Queues
  - b. Cloud based applications are using message queues for sending and receiving messages
- 3. Skills Developed**
  - a. Configuration of message queues and server
  - b. Generating and Consuming messages on message queues

## **References :**

[1] <https://www.rabbitmq.com/>

[2] [https://en.wikipedia.org/wiki/Message\\_queuing\\_service](https://en.wikipedia.org/wiki/Message_queuing_service)

## **Viva Questions**

4. What is SaaS Model?
5. List components of Message Queuing Server?
6. Which version of visual studio is used for this experiment?

## **Teachers Interaction with Students**

1. Concept of Message Queuing
2. Rabbit MQ/ Azure Service Bus Configuration
3. Coding for Generating and Consuming Messages



Hope Foundation's  
**Finolex Academy of Management and Technology, Ratnagiri**  
Information Technology Department

Subject name: Cloud Computing      Subject Code: BEITC702

Class      BE IT      Semester – VIII  
(CBGS)      Academic year: 2018-19

Name of Student      **QUIZ Score :**

Roll No      Assignment/Experiment No.      05

**Title:** MongoDB NoSQL database CRUD Operations

**1. Course objectives applicable**

**COB2.Understand** benefits of various cloud services and Handle Open Source Cloud Implementation and Administration

**COB6.Understand** AAA administration and mobile cloud computing

**2. Course outcomes applicable:**

**CO2** -List benefits of different cloud services, types of services, Concept of Anything as A Service. Handle Open Source Cloud Implementation and Administration using OpenStack or Eucalyptus.

**CO6**-Explain cloud administration process, model for secure access for cloud and mobile cloud computing and its future implications

**3. Learning Objectives:**

1. To understand concept of NoSQL Database and Document Database
2. Create Connection String for MongoDB
3. Perform CRUD Operations on MongoDB
4. Understand CAP Theorem and BASE Principals of NoSQL

**4. Practical applications of the assignment/experiment:**

MongoDB is a leading Document Database used in Unstructured Data Processing

**5. Prerequisites:**

1. Knowledge of NoSQL Database
2. Internet Access
3. MongoDB Server Setup,
4. In case of online deployment, MongoDB access on [www.mlab.com](http://www.mlab.com)

**6. Hardware Requirements:**

1. Internet Access with Browser
2. PC to install MongoDB

**7. Software Requirements:**

Browser like Chrome, Internet Explorer Edge, MongoDB Server Setup

**8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)**

1. What is Service Bus?
2. Which version of Visual Studio is used?
3. What is an Access Key?

**9. Experiment/Assignment Evaluation:**

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A <u>or</u> any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
<b>Date of performance (DOP)</b>		<b>Total marks obtained</b>	<b>10</b>
<b>Date of checking (DOC)</b>		<b>Signature of teacher</b>	

## 1. Theory: <Preferably given as handwritten work for students>

A NoSQL (originally referring to "non SQL" or "non relational")<sup>[1]</sup> database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Such databases have existed since the late 1960s, but did not obtain the "NoSQL" moniker until a surge of popularity in the early twenty-first century,<sup>[2]</sup> triggered by the needs of Web 2.0 companies such as Facebook, Google, and Amazon.com.<sup>[3][4][5]</sup> NoSQL databases are increasingly used in big data and real-time web applications.<sup>[6]</sup> NoSQL systems are also sometimes called "Not only SQL" to emphasize that they may support SQL-like query languages.<sup>[7][8]</sup>

MongoDB (from humongous) is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is developed by MongoDB Inc., and is published under a combination of the GNU Afferro General Public License and the Apache License.

There have been various approaches to classify NoSQL databases, each with different categories and subcategories, some of which overlap. What follows is a basic classification by data model, with examples:

- **Column:** [Accumulo](#), [Cassandra](#), [Druid](#), [HBase](#), [Vertica](#), [SAP HANA](#)
- **Document:** [Apache CouchDB](#), [ArangoDB](#), [Clusterpoint](#), [Couchbase](#), [Cosmos DB](#), [HyperDex](#), [IBM Domino](#), [MarkLogic](#), [MongoDB](#), [OrientDB](#), [Qizx](#), [RethinkDB](#)
- **Key-value:** [Aerospike](#), [ArangoDB](#), [Couchbase](#), [Dynamo](#), [FairCom c-treeACE](#), [FoundationDB](#), [HyperDex](#), [InfinityDB](#), [MemcacheDB](#), [MUMPS](#), [Oracle NoSQL Database](#), [OrientDB](#), [Redis](#), [Riak](#), [Berkeley DB](#), SDBM/Flat File [dbm](#)
- **Graph:** [AllegroGraph](#), [ArangoDB](#), [InfiniteGraph](#), [Apache Giraph](#), [MarkLogic](#), [Neo4J](#), [OrientDB](#), [Virtuoso](#)
- **Multi-model:** [ArangoDB](#), [Couchbase](#), [FoundationDB](#), [InfinityDB](#), [MarkLogic](#), [OrientDB](#)

### CAP Theorem

Eric Brewer, a professor at the University of California, Berkeley, and cofounder and chief scientist at Inktomi, made the conjecture that Web services cannot ensure all three of the following properties at once (signified by the acronym CAP):<sup>2</sup>

**Consistency.** The client perceives that a set of operations has occurred all at once.

**Availability.** Every operation must terminate in an intended response.

**Partition tolerance.** Operations will complete, even if individual components are unavailable.

Specifically, a Web application can support, at most, only two of these properties with any database design. Obviously, any horizontal scaling strategy is based on data partitioning; therefore, designers are forced to decide between consistency and availability.

The **BASE** acronym is used to describe the properties of certain databases, usually NoSQL databases. It's often referred to as the opposite of ACID. In the NoSQL world, ACID transactions are less fashionable as some databases have loosened the requirements for immediate consistency, data freshness and accuracy in order to gain other benefits, like scale and resilience.

This leaves me with some questions about **the definition:**

Basically Available, Soft state, Eventual consistency

Basic Availability -The database appears to work most of the time.

Soft-state- Stores don't have to be write-consistent, nor do different replicas have to be mutually consistent all the time.

Eventual consistency - Stores exhibit consistency at some later point (e.g., lazily at read time).

BASE properties are much looser than ACID guarantees, but there isn't a direct one-for-one mapping between the two consistency models.

A BASE datastore values availability (since that's important for scale), but it doesn't offer guaranteed consistency of replicated data at write time. Overall, the BASE consistency model provides a less strict assurance than ACID: data will be consistent in the future, either at read time (e.g., Riak) or it will always be consistent, but only for certain processed past snapshots (e.g., Datomic).

The BASE consistency model is primarily used by aggregate stores, including column family, key-value and document stores.

## 10. Precautions :

1. Configure the MongoDB server first before starting CRUD Operations.

## 11. Installation Steps / Performance Steps –

### Executing NoSql Commands on MongoDB shell

```
tar -zxvf mongodb-linux-x86_64-3.4.7.tgz  
mkdir -p mongodb  
  
cp -R -n mongodb-linux-x86_64-3.4.7/ mongodb  
export PATH=/home/students/bin:$PATH mkdir  
-p /data/db  
  
chmod 777 /data/db  
  
nevigate to bin directory cd ../  
.mongod
```

This will start a server and will wait for client request

```

Activities Terminal Mon 12:10*
root@localhost:~/mongodb/mongodb-linux-x86_64-3.4.7/bin

File Edit View Search Terminal Help
check.py      mongodb-linux-x86_64-3.4.7    sendi.py.save
data.txt      mongodb-linux-x86_64-3.4.7.tgz  send.py
ddk.pem       Music                           server.py
Desktop      pi.py                            success.txt
Documents    Pictures                         t
Downloads   Public                           t2.txt
epel-release-6-8.noarch.rpm  RabbitMQ.odt    task.sh
fedora26.pem  receive111.py                 Templates
function1.py   receive1.py                  t.txt
google-cloud-sdk  receive.py                Videos
live.sh      remi-release-6.rpm             'VirtualBox VMs'
[root@localhost ~]# cd mongodb
[root@localhost mongodb]# ls
mongodb-linux-x86_64-3.4.7
[root@localhost mongodb]# cd mongodb-linux-x86_64-3.4.7
[root@localhost mongodb-linux-x86_64-3.4.7]# ls
bin  GNU-AGPL-3.0  MPL-2  README  THIRD-PARTY-NOTICES
[root@localhost mongodb-linux-x86_64-3.4.7]# cd bin/
[root@localhost bin]# ls
bsondump  mongodump  mongoimport  mongoreplay  mongostat
mongo    mongoreport  mongooplog  mongorestore  mongotop
mongod  mongofiles  mongoperf  mongos
[root@localhost bin]# ./mongod
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten] MongoDB starting : pid=17606 port=27017 dbpath=/data/db 64-bit host=localhost.localdomain
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten] db version v3.4.7
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten] git version: cf38c1b8a0a8dca4a11737581beafe4fe120bcd
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten] allocator: tcmalloc
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten] modules: none
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten] build environment:
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten]   distarch: x86_64
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten]   target arch: x86_64
2017-09-04T12:07:04.110+0530 I CONTROL [initandlisten] options: {}
2017-09-04T12:07:04.160+0530 I -          [initandlisten] Detected data files in /data/db created by the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
2017-09-04T12:07:04.160+0530 I STORAGE [initandlisten]
2017-09-04T12:07:04.160+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2017-09-04T12:07:04.160+0530 I STORAGE [initandlisten] See http://dochub.mongodb.org/core/prodnotes-filesystem

```

OPEN ONE MORE TERMINAL , NAVIGATE TO EXTRACTED MONGODB FOLDER cd mongo ..

```

cd bin
./mongo

```

It would start mongo shell where you can run nosql commands.. as follows

```

Activities Terminal Mon 12:11*
root@localhost:~/mongodb/mongodb-linux-x86_64-3.4.7/bin

File Edit View Search Terminal Help
[root@localhost bin]# pwd
/root/mongodb/mongodb-linux-x86_64-3.4.7/bin
[root@localhost bin]# ./mongo
MongoDB shell version v3.4.7
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.7
Server has startup warnings:
2017-09-04T12:07:04.160+0530 I STORAGE [initandlisten]
2017-09-04T12:07:04.160+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2017-09-04T12:07:04.160+0530 I STORAGE [initandlisten] See http://dochub.mongodb.org/core/prodnotes-filesystem
2017-09-04T12:07:05.303+0530 I CONTROL [initandlisten]
2017-09-04T12:07:05.303+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2017-09-04T12:07:05.303+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2017-09-04T12:07:05.303+0530 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2017-09-04T12:07:05.303+0530 I CONTROL [initandlisten]
>

```

**Now run CRUD operations on shell**

To see the list of databases in the system:  
>show dbs

To select a database:

```
> use movies switched to
```

```
db movies
```

```
> db.comedy.insert({name:"Sairat", year:2016})
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.comedy.insert({name:'The School of Rock', year:2003})
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.comedy.find()
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016
}{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }
```

```
> db.comedy.find().limit(1)
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016
}
```

```
> db.comedy.findOne()
```

```
{
  "_id" : ObjectId("59acf61f5726b577df63e869"),
  "name" : "Sairat",
  "year" : 2016
}
```

```
> db.comedy.find({year:{$gt:1994}})
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016
}
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }
```

```
> db.comedy.find({year:{'$gt':1994}}, {name:true})
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat" }
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock" }
```

```
> db.comedy.insert({name:"Bill & Ted's Excellent Adventure", year:1989})
```

```
WriteResult({ "nInserted" : 1 })
```

**Now Update a document**

```
> db.comedy.update({name:"Bill & Ted's Excellent Adventure"}, {'$set':
{director:'Stephen Herek', cast:['Keanu Reeves', 'Alex Winter']}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.comedy.find()
```

```
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016
}
```

```
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }
```

```
{ "_id" : ObjectId("59acf6ba5726b577df63e86b"), "name" : "Bill & Ted's Excellent
Adventure", "year" : 1989, "director" : "Stephen Herek", "cast" : [ "Keanu
Reeves", "Alex Winter" ] }
```

**Now remove a document**

```

db.comedy.remove({name:'Sairat'})

WriteResult({ "nRemoved" : 1 })
> db.comedy.find()
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }
{ "_id" : ObjectId("59acf6ba5726b577df63e86b"), "name" : "Bill & Ted's Excellent
Adventure", "year" : 1989, "director" : "Stephen Herek", "cast" : [ "Keanu
Reeves", "Alex Winter" ] }

> db.comedy.count()
2
Now drop a collection
> db.comedy.drop()

true

> show dbs

admin 0.000GB

local 0.000GB

```

## 2.Observations

1. MongoDB uses SQL commands for basic operations
2. MongoDB is one of the documents based database, the documents can be simple text based records as used to describe objects

## 3. Results:

Results are mentioned in the scripts run in Section 11 above.

## 4. Learning Outcomes Achieved

1. MondoDB as NoSQL database is discussed
2. Students Downloaded and Installed MongoDB Server, Students Configured MongoDB Database
3. SQL Queries for CRUD Operations were tested on the databases
4. CAP theorem and BASE properties of NoSQL database are discussed in connection with MongoDB for Big Data analytics

## 5. Conclusion:

1. **Applications of the studied technique in industry**
  - a. MongoDB is one of the commonly used document store
  - b. BASE properties are of relevance in view of Big Data Analytics
2. **Engineering Relevance**
  - a. Distributed databases for storage of large volumes of data
  - b. NoSQL for Distributed Databases

### **3. Skills Developed**

- a. Configuration of MongoDB
- b. CRUD commands on MongoDB

## **References :**

- [1] <http://nosql-database.org/> "NoSQL DEFINITION: Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontally scalable"
- [2] Leavitt, Neal (2010). "Will NoSQL Databases Live Up to Their Promise?" (PDF). IEEE Computer.
- [3] Mohan, C. (2013). History Repeats Itself: Sensible and NonsenseSQL Aspects of the NoSQL Hoopla. Proc. 16th Int'l Conf. on Extending Database Technology.
- [4] "NOSQL meetup Tickets, Thu, Jun 11, 2009 at 10:00 AM". Eventbrite.com. Retrieved 2017-03-06.
- [5] "Amazon Goes Back to the Future With 'NoSQL' Database". WIRED. 2012-01-19. Retrieved 2017-03-06.
- [6] "RDBMS dominate the database market, but NoSQL systems are catching up". DB-Engines.com. 21 Nov 2013. Retrieved 24 Nov 2013.
- [7] "NoSQL (Not Only SQL)". NoSQL database, also called Not Only SQL
- [8] Fowler, Martin. "NoSQL Definition". many advocates of NoSQL say that it does not mean a "no" to SQL, rather it means Not Only SQL

## **Viva Questions**

- 4. What is NoSQL?
- 5. What is a Document Database?
- 6. Give Examples of other Document Databases?

## **Teachers Interaction with Students**

- 1. Concept of NoSQL
- 2. MongoDB Configuration
- 3. ACID and BASE differentiation



Hope Foundation's  
**Finolex Academy of Management and Technology, Ratnagiri**  
Information Technology Department

Subject name: Cloud Computing		Subject Code: BEITC702	
Class	BE IT	Semester – VIII (CBGS)	Academic year: 2018-19
Name of Student	<b>QUIZ Score :</b>		
Roll No	Assignment/Experiment No.		
<b>Title:</b> Python Program deployment on Google App Engine			

**1. Course objectives applicable**

**COB3.**Understand various factors for cloud deployment, security aspects at various levels of cloud computing environments.

**COB4.**Study cloud architecture, explain difference with traditional approach and programming support for cloud.

**2. Course outcomes applicable:**

**CO3** -Explain various factors for cloud deployment and key security issues at virtualization as well as network and service level

**CO4**-Explain cloud architecture, give basic consideration for design cloud computing environment. Explain difference between traditional and cloud programming model. Design, program and deploy application on public cloud such as Google App Engine, AWS, Microsoft Azure

**3. Learning Objectives:**

1. To write a python script for a simple web page
2. Create Google App Engine Upload Environment
3. Upload Python Script on GAE
4. Test Python Script on Google Cloud

**4. Practical applications of the assignment/experiment:**

Google compute cloud is a major IaaS provider, python is one of the supported language. This lab is an example of publishing a service on cloud.

**5. Prerequisites:**

1. Knowledge of Google compute cloud and python programming
2. Internet Access
3. Python compiler
4. In case on online deployment, GCC Access

**6. Hardware Requirements:**

1. Internet Access with Browser
2. PC to install Python Compiler

**7. Software Requirements:**

Browser like Chrome, Internet Explorer Edge, Python Compiler Setup

**8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)**

1. What model GCC follows ?
2. Which version of Python compiler is used?

3. What is an Oauth?

**9. Experiment/Assignment Evaluation:**

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A <u>or</u> any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
<b>Date of performance (DOP)</b>		<b>Total marks obtained</b>	<b>10</b>
<b>Date of checking (DOC)</b>		<b>Signature of teacher</b>	

## **9. Theory: <Preferably given as handwritten work for students>**

Google App Engine (often referred to as GAE or simply App Engine) is a web framework and cloud computing platform for developing and hosting web applications in Google-managed data centers. Applications are sandboxed and run across multiple servers.[1] App Engine offers automatic scaling for web applications—as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.[2]

Google App Engine is free up to a certain level of consumed resources. Fees are charged for additional storage, bandwidth, or instance hours required by the application.[3] It was first released as a preview version in April 2008 and came out of preview in September 2011.

Supported programming languages include Python, Ruby, Java (and, by extension, other JVM languages such as Kotlin, Groovy, JRuby, Scala, Clojure), Go, and PHP. Node.js is also available in the flexible environment. Google has said that it plans to support more languages in the future, and that the Google App Engine has been written to be language independent.[4] C# is also supported.[5] Arbitrary Docker containers are also supported.[6]

Python web frameworks that run on Google App Engine include Django, CherryPy, Pyramid, Flask, web2py and webapp2,[7] as well as a custom Google-written webapp framework and several others designed specifically for the platform that emerged since the release.[8] Any Python framework that supports the WSGI using the CGI adapter can be used to create an application; the framework can be uploaded with the developed application. Third-party libraries written in pure Python may also be uploaded.[9][10]

Google App Engine supports many Java standards and frameworks. Core to this is the servlet 2.5 technology using the open-source Jetty Web Server,[11] along with accompanying technologies such as JSP. JavaServer Faces operates with some workarounds. A newer release of App Engine Standard Java in Beta supports Java8, Servlet 3.1 and Jetty9.

### **Other Parallel technologies:**

- |     |                       |
|-----|-----------------------|
| 1.  |                       |
| 2.  | AppScale              |
| 3.  | AWS Elastic Beanstalk |
| 4.  | Bluemix               |
| 5.  | Cocaine (PaaS)        |
| 6.  | Engine Yard           |
| 7.  | Force.com             |
| 8.  | GoGrid                |
| 9.  | Google Apps Script    |
| 10. | Google Cloud Platform |
| 11. | Heroku                |
| 12. | Jelastic              |
| 13. | Microsoft Azure       |
| 14. | OpenShift             |
| 15. | Rackspace Cloud       |
| 16. | Skytap                |
| 17. | VMware                |
| 18. | Amazon Web Services   |

**10. Precautions :**

1. Use google account for login
2. Internet should be connected in order to loadLogin using Super user privileges

**11. Installation Steps / Performance Steps -**

**Running python on google app engine**

**1-use curl command to download google sdk curl  
https://sdk.cloud.google.com | bash**

**2-cd goo...(It will found in present directory)**

**3- run a script ./google-cloud-sdk/install.sh .**

**4-follow instructions**

**keep pressing yes**

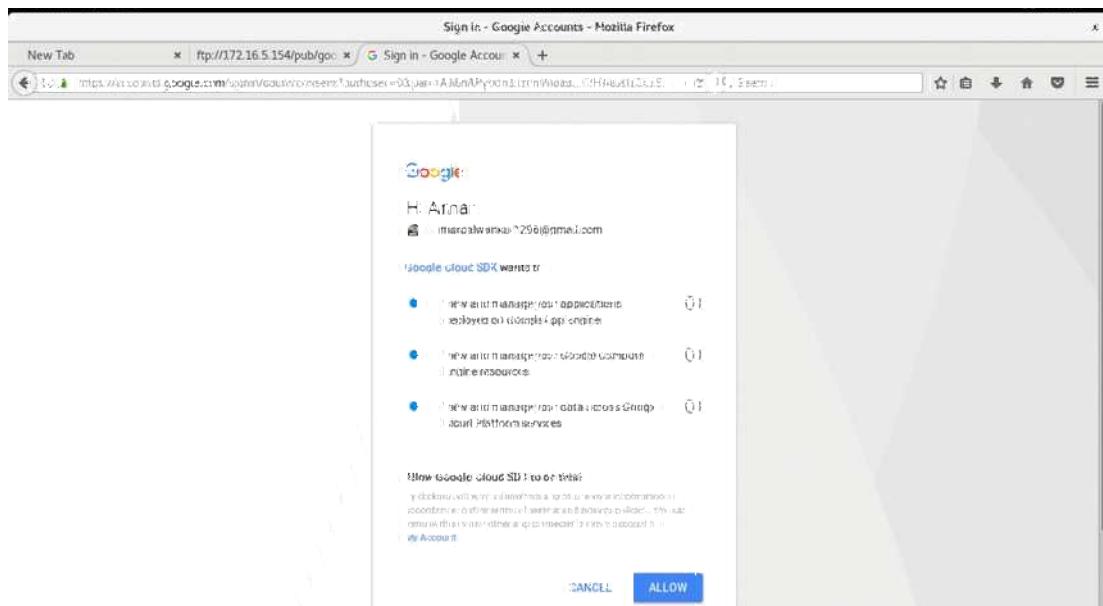
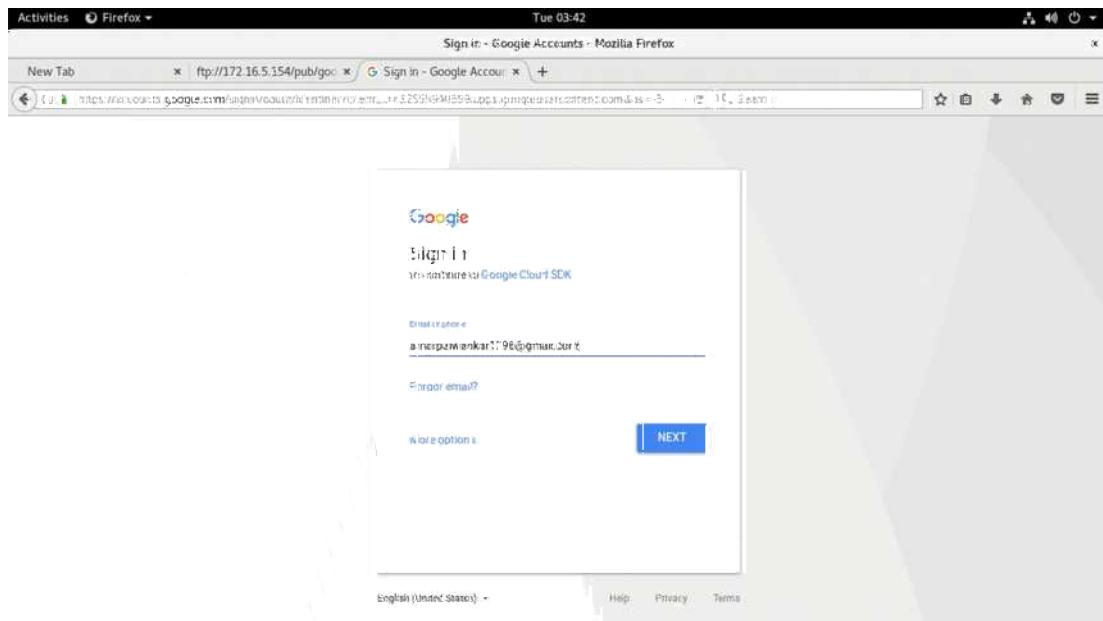
**5-after done,close terminal and reopen and enter in google-cloud-sdk directory**

**cd google-cloud-sdk/**

**6.Authenticating with the Cloud SDK**

**Before doing anything else, run this command: gcloud auth login**

**Now it will open a browser and will ask you to login**



Activities Firefox Tue 03:44

You are now authenticated with the Google Cloud SDK | Cloud SDK Documentation | Google Cloud Platform - Mozilla Firefox

New Tab x ftp://172.16.5.154/pub/goo x You are now authenti... +

https://console.cloud.google.com/usingauth\_stacksdk/python

Google Cloud Platform

Who's Online Products Solutions + Launchpad Training Custom... Documentation Support Partners

CONSOLIDATE

TOPICS

here are some links to help you get started with Google Cloud Platform services.

- Build a web app and host it on Google App Engine.  
To get started, follow the walkthrough in the Google Cloud Platform Tutorials: [Try Google App Engine Now](#)
- Launch large compute clusters on Google Compute Engine.  
To get started, follow the [Compute Engine Quickstart Guide](#)
- Store vast amounts of data on Google Cloud Storage.  
To get started, follow the [Hello Google Cloud Storage Tutorial](#).
- Analyze Big Data from cloud via Google BigQuery.  
To get started, follow the [sq Command-Line Tool Quickstart](#).
- Store and manage data using a MySQL database with Google Cloud SQL.  
To get started, see [Managing Instances Using the Cloud SDK](#).
- Make your specifications and services available to your users with Google Cloud DNS.  
To get started, see [Getting started with Google Cloud DNS](#).

Feedback

Send feedback

https://console.cloud.google.com/start/appengine

Activities Firefox Tue 03:44

Try App Engine now - My Project 11 - Mozilla Firefox

New Tab x ftp://172.16.5.154/pub/goo x Try App Engine now - My Project 11 +

https://console.cloud.google.com/appengine/start?sa=1204007948,-1281048115143249023&mt=1&project=MyProject11

Dismiss

Google Cloud Platform My Project 11

Welcome to App Engine

Your first app

Learn how to build and deploy on App Engine with a simple "Hello World" app. If you're new to App Engine, then start here.

Java Python PHP

Node.js Java Python PHP

Go Ruby .NET

App Engine Docs

User interface overview (PDF) API Reference and libraries (PDF) Standard Environment (PDF) Flexible Environment (PDF)

Regions

Cloud Shell

Cloud Build

Feedback

Choose a language (Python)

Now Open a google cloud shell and enter following commands(Just copy there itself)

```
TUTORIALDIR=~/src/my-project-11-179305/python_gae_quickstart-2017-09-12-03-50
```

```
git clone https://github.com/GoogleCloudPlatform/python-docs-samples $TUTORIALDIR  
cd $TUTORIALDIR/appengine/standard/hello_world
```

Now Open a main.py file and make changes and finally save it

Now its time to deploy our web app.

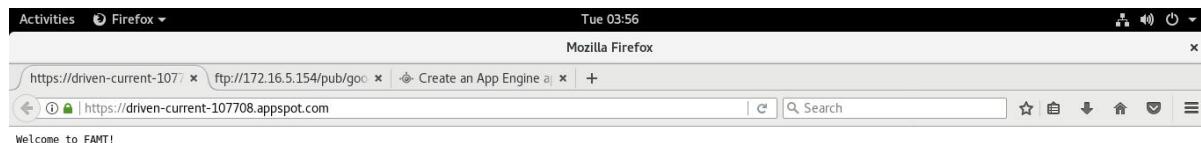
```
gcloud app deploy app.yaml --project driven-current-107708
```

Now app is deployed ,access it with displayed url.  
O/P

## 12.Observations

1. MongoDB uses SQL commands for basic operations
2. MongoDB is one of the documents based database, the documents can be simple text based records as used to describe objects

## 13. Results:



## 14. Learning Outcomes Achieved

1. A python web page was created by student and tested locally
2. Google App Engine Connectivity on Fedora Command prompt was done
3. Google Account was connected and the python page was transferred to the Google App Engine

4. The page was accessed on the Appsoft web portal

## 15. Conclusion:

1. **Applications of the studied technique in industry**
  - a. Google App Engine is one of the prominent IaaS
  - b. Python Script is a common script language used in cloud programming
2. **Engineering Relevance**
  - a. Deployment of programs on Cloud Infrastructure was done
  - b. Single Sign on to google services was used for cloud access
3. **Skills Developed**
  - a. Cloud App Upload
  - b. Cloud App hosting and configuration

## References :

1. "Python Runtime Environment - Google App Engine - Google Code". Code.google.com. 1999-02-22. Retrieved 2012-02-14.
2. Sanderson, Dan (2009). Programming Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure. O'Reilly Media. ISBN 978-0-596-52272-8.
3. "Quotas - Google App Engine - Google Code". Code.google.com. 1999-02-22. Retrieved 2012-02-14.
4. Sanderson, Dan (2010). Programming Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure. O'Reilly Media. ISBN 978-0-596-52272-8.
5. Krill, Paul. "Google App Engine adds C#, Node.js, and Ruby options".
6. "Building Custom Runtimes - Custom runtimes for the App Engine flexible environment - Google Cloud Platform". Google Cloud Platform.
7. "Welcome to webapp2! — webapp2 v2.5.1 documentation". Webapp-improved.appspot.com. Retrieved 2012-02-14.
8. "AppEngine Frameworks - tipfy - The almighty little framework for Google App Engine - Google Project Hosting". Code.google.com. Retrieved 2012-02-14.
9. "What Is Google App Engine? - Google App Engine - Google Code". Code.google.com. 1999-02-22. Retrieved 2012-02-14.
10. "webapp Overview - Google App Engine - Google Code". Code.google.com. 1999-02-22. Retrieved 2012-02-14.
11. "Google Chose Jetty for App Engine". Infoq.com. 2012-07-13. Retrieved 2012-07-17.

## Viva Questions

4. What is IaaS?
5. What is a Google App Engine?
6. Which version of Python SDK was used?
7. What is SSO? Give its relevance to this lab.

## Teachers Interaction with Students

1. Concept of IaaS
2. GAE Configuration
3. SSO and its relevance



**Finolex Academy of Management and Technology, Ratnagiri**  
**Department of Information Technology**

<b>Subject:</b>	<b>Cloud Computing (BEITL702)</b>		
<b>Class:</b>	<b>BE IT / Semester – VII (CBGS) / Academic year: 2017-2018</b>		
<b>Student Name</b>			
<b>Roll No:</b>		<b>Date of performance (DOP) :</b>	
<b>Assignment/Experiment No:</b>	<b>7</b>	<b>Date of checking (DOC) :</b>	
<b>Title:</b>	Microsoft Azure IaaS, PaaS, SaaS Operations		
<b>Marks</b>		<b>Teacher's Signature:</b>	

**1. Aim:** To demonstrate Azure WebRole Deployment on Azure Emulator

**2. Prerequisites:**

1. Knowledge of C#.NET, ASP.NET
2. Internet Access
3. Azure Account

**3. Hardware Requirements:**

1. Internet Access with Browser
2. PC with 4GB RAM

**4. Software Requirements:**

Visual Studio 2015 Community Edition

**5. Learning Objectives:**

1. To write a Web Role and Compile in Visual Studio.NET
2. Deploy the web role on Azure Compute Emulator
3. Study the web role deployment states
4. Configure Web Application for Web role Access

**6. Course Objectives Applicable:** CO1, CO4

**7. Program Outcomes Applicable:** PO1, PO2, PO4, PO5, PO9

**8. Program Education Objective Applicable:** PEO 2, PEO 3, PEO6

## **9. Theory: <Preferably given as handwritten work for students>**

Microsoft Azure (formerly Windows Azure) /'æzər/ is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service and infrastructure as a service and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.

Azure was announced in October 2008 and released on February 1, 2010 as "Windows Azure" before being renamed "Microsoft Azure" on March 25, 2014.[1][2]

### **Other Examples**

1. Amazon Web Services
2. IBM Bluemix
3. Cloud computing
4. Comparison of file hosting services
5. Google Cloud Platform
6. Microsoft Azure Web Sites
7. Predix (software)

### **What is a Web Role?**

Web Role is a Cloud Service role in Azure that is configured and customized to run web applications developed on programming languages / technologies that are supported by Internet Information Services (IIS), such as ASP.NET, PHP, Windows Communication Foundation and Fast CGI.

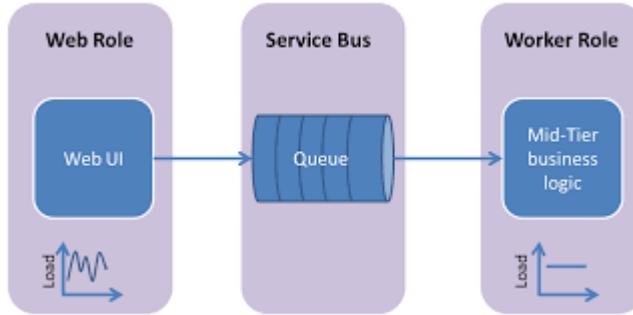
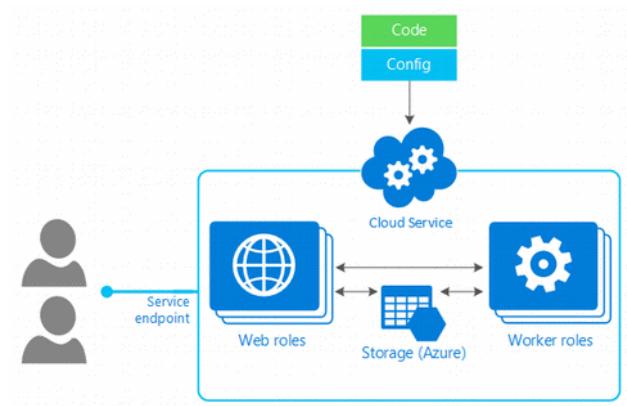
### **What is a Worker Role?**

Worker Role is any role in Azure that runs applications and services level tasks, which generally do not require IIS. In Worker Roles, IIS is not installed by default. They are mainly used to perform supporting background processes along with Web Roles and do tasks such as automatically compressing uploaded images, run scripts when something changes in database, get new messages from queue and process and more.

### **Differences between Web and Worker Roles**

The main difference between the two is that a Web Role support and runs Internet Information Services (IIS), while an instance of a Worker Role does not. As being deployed and delivered through Azure Service Platform, both can be managed in the same way and can be deployed on a similar Azure Instance. In most scenarios, Web Role and Worker Role instance work together and are often used by an application simultaneously.

For example, a web role instance might accept requests from users, then pass them to a worker role instance for processing



## 10. Precautions :

1. Shut Down the webrole if you are deploying on Azure Cloud as it will keep on charging in your monthly bill

## 11. Installation Steps / Performance Steps –

### Web Role Code :

```

using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.Diagnostics;
using Microsoft.WindowsAzure.ServiceRuntime;

namespace BE_IT_CC
{
    public class WebRole : RoleEntryPoint
    {
        int interloop=7;
        public override bool OnStart()
        {
            // For information on handling configuration changes
            // see the MSDN topic at http://go.microsoft.com/fwlink/?LinkId=166357.

            Console.WriteLine("This Worker Role has " + interloop + "Loops");
        }
    }
}

```

```

        return base.OnStart();
    }
}
}

```

### Web Page :

```

<!-- ko with: home -->
<div class="jumbotron">
    <h1>ASP.NET</h1>
    <p class="lead">ASP.NET is a free web framework for building great Web sites and
    Web applications using HTML, CSS, and JavaScript.</p>
    <p><a href="http://asp.net" class="btn btn-primary btn-lg">Learn more
    &raquo;</a></p>
</div>
<div class="row">
    <div class="col-md-4">
        <h2>Your information</h2>
        <p>This section shows how you can call ASP.NET Web API to get the user
        details.</p>
        <p data-bind="text: myHometown"></p>

        <p><a class="btn btn-default" href="http://go.microsoft.com/fwlink/?LinkId=273732">Learn more &raquo;</a></p>
    </div>

    <div class="col-md-4">
        <h2>Getting started</h2>
        <p>
            ASP.NET Single Page Application (SPA) helps you build applications that
            include significant client-side interactions using HTML, CSS, and JavaScript.
            It's now easier than ever before to get started writing highly
            interactive web applications.
        </p>
        <p><a class="btn btn-default" href="http://go.microsoft.com/fwlink/?LinkId=273732">Learn more &raquo;</a></p>
    </div>

    <div class="col-md-4">
        <h2>Web Hosting</h2>
        <p>You can easily find a web hosting company that offers the right mix of
        features and price for your applications.</p>
        <p><a class="btn btn-default" href="http://go.microsoft.com/fwlink/?LinkId=301867">Learn more &raquo;</a></p>
    </div>
</div>

<!-- /ko -->

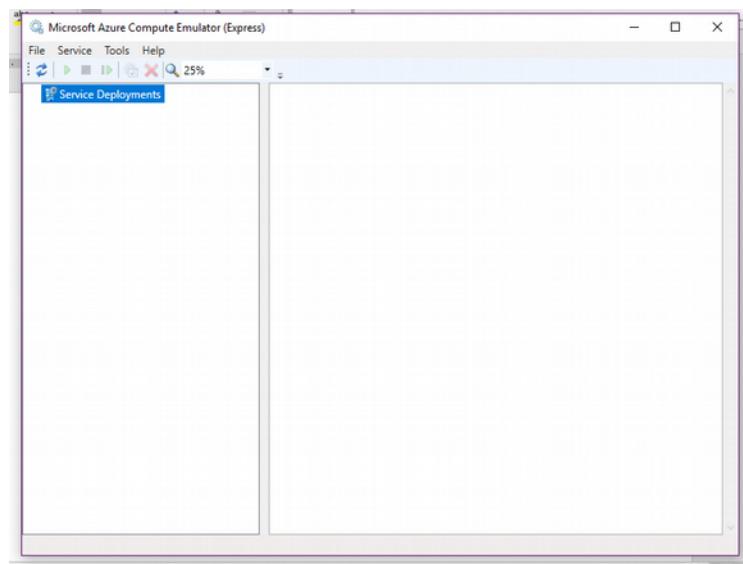
```

## 12.Observations

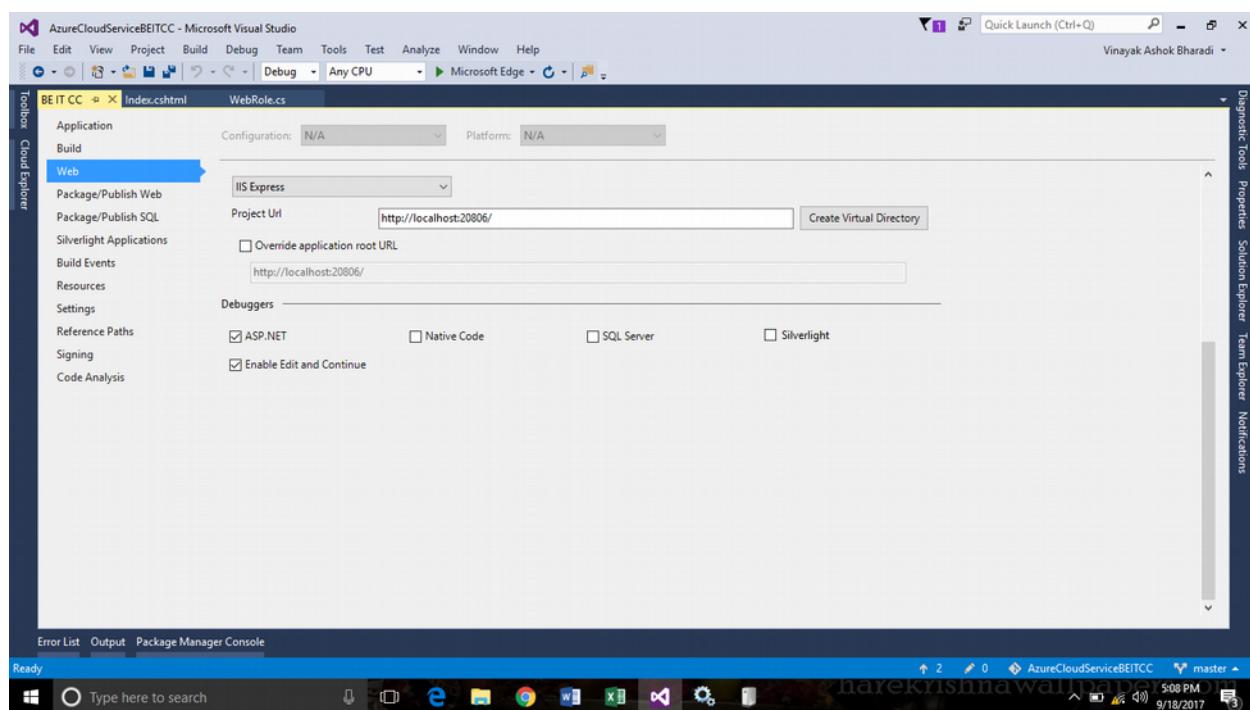
1. C# .Net and ASP.NET are used for coding, the ASP web page is hosted by IIS
2. Worker Role is emulated on Azure Compute Emulator

## 13. Results:

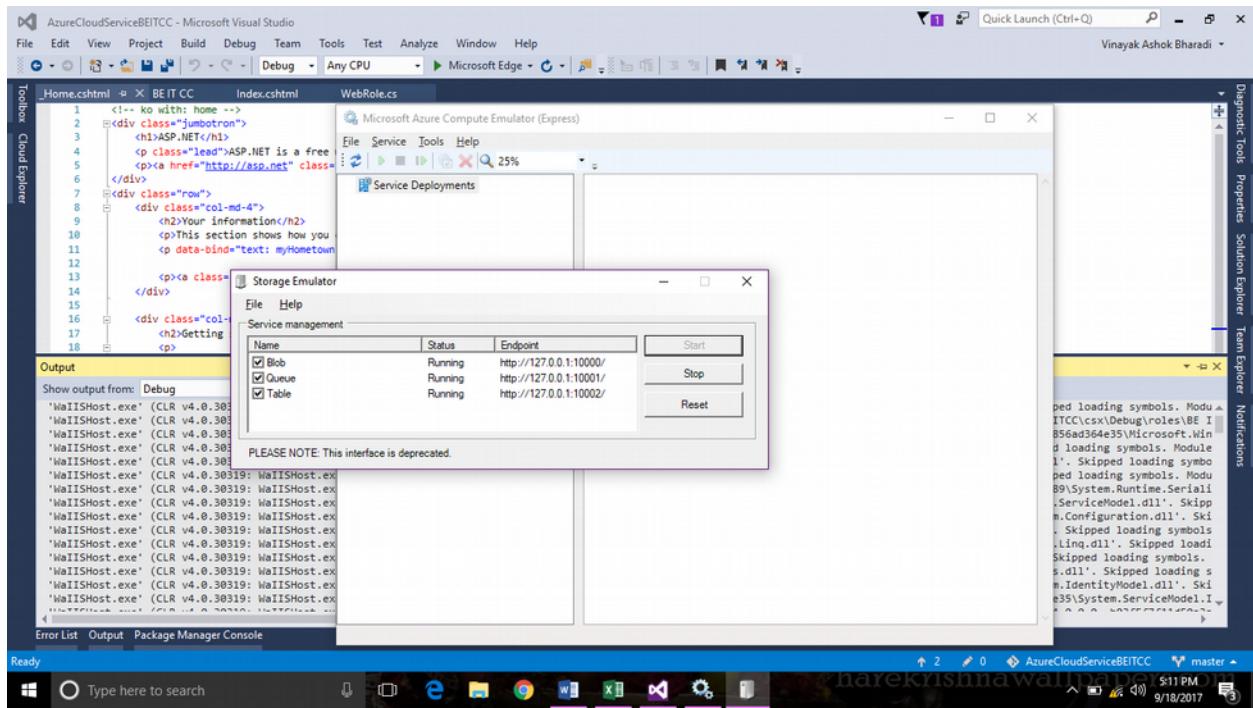
### 1. Azure Compute Emulator Before Deployment of the Worker Role



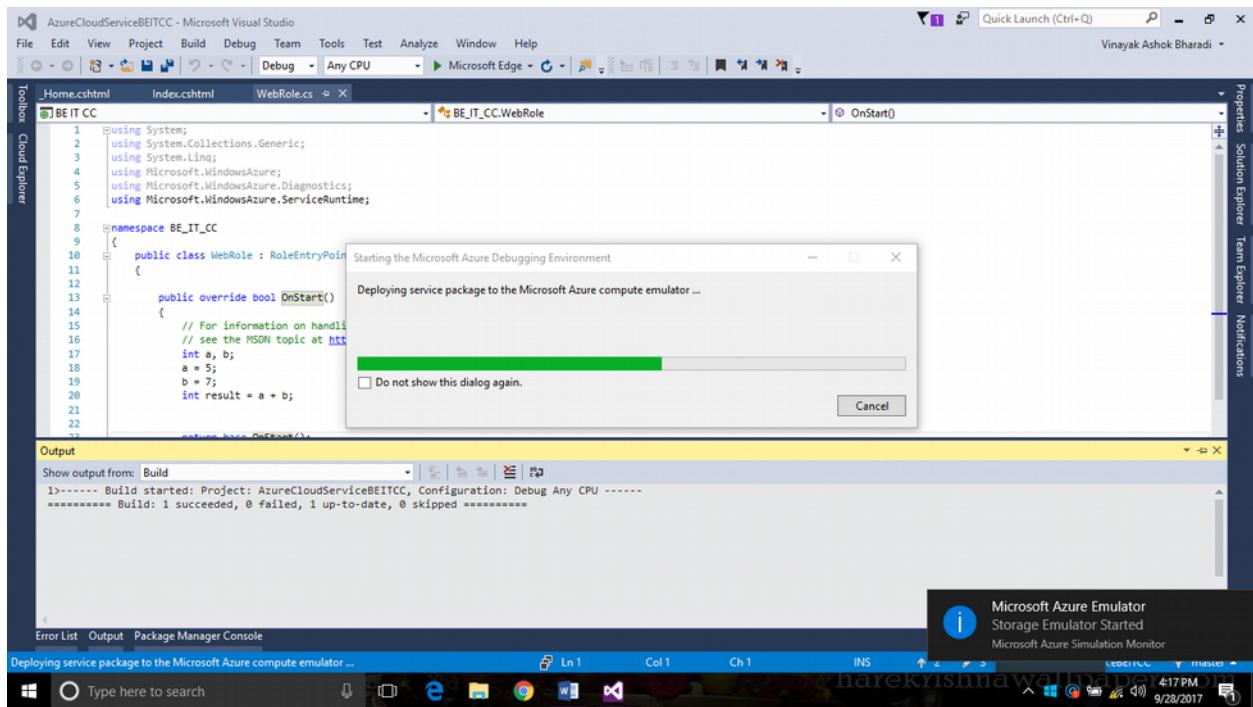
### 2. Worker Role Preferences



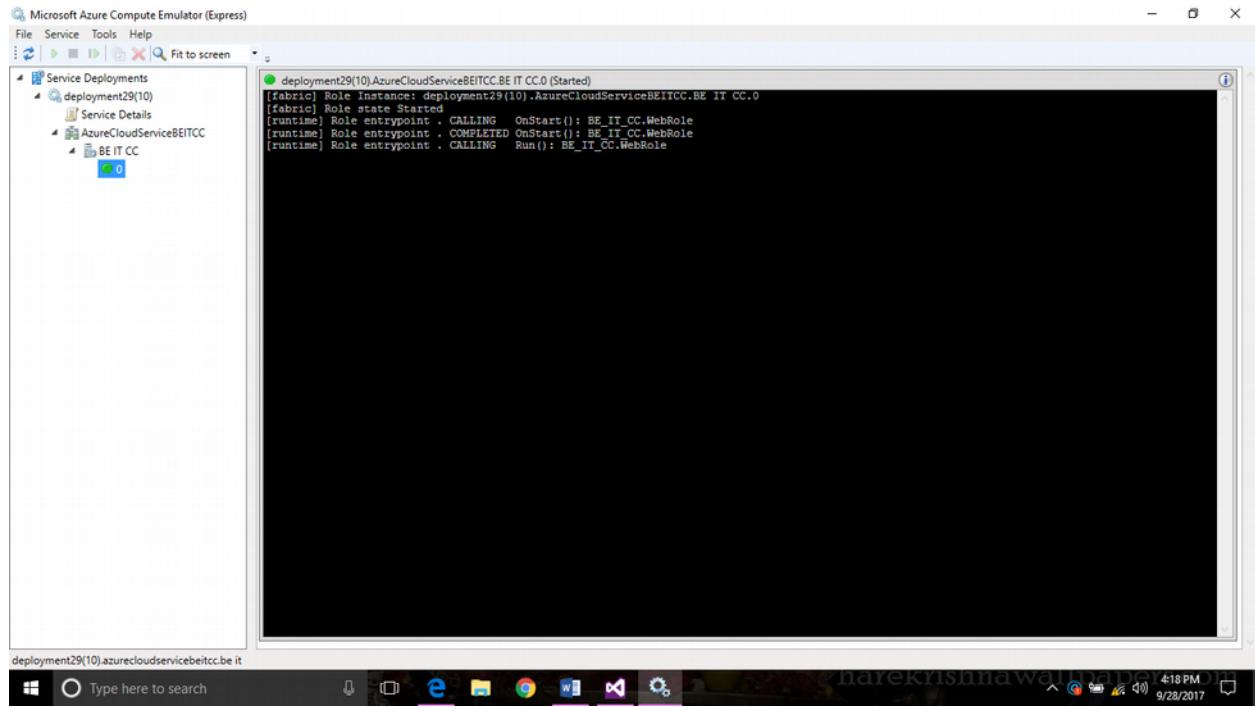
### 3. Azure Storage Emulator Stats



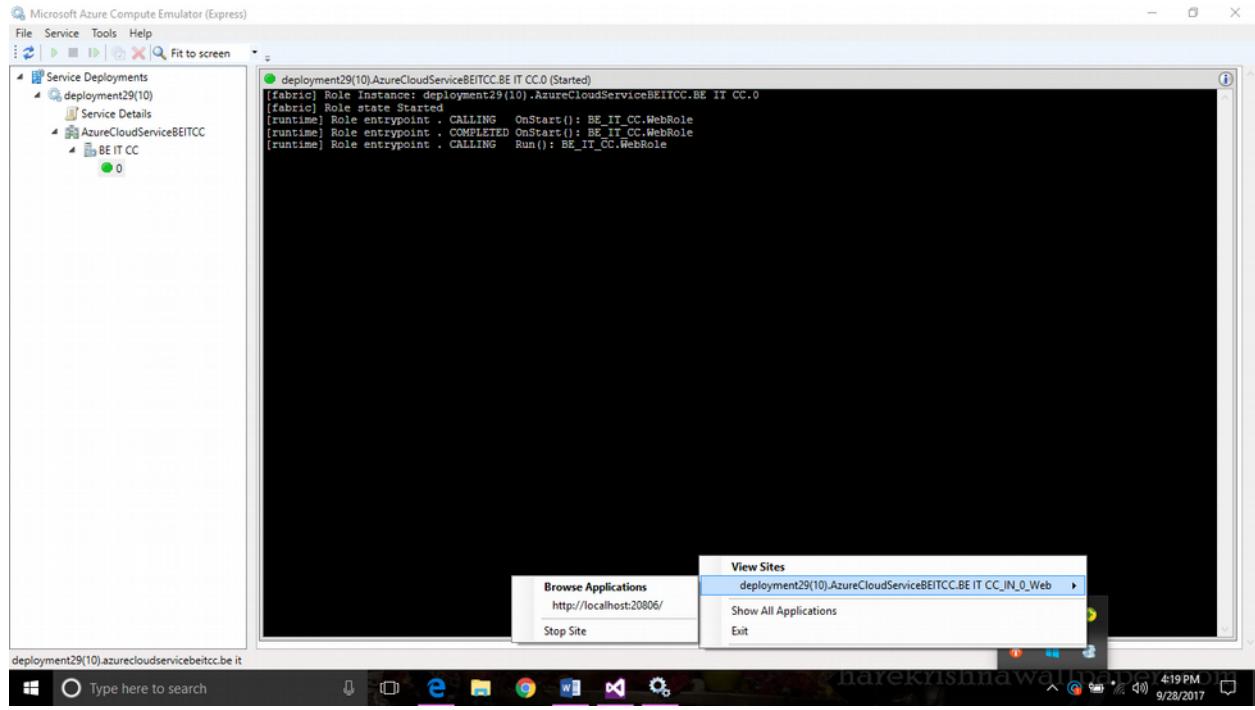
#### 4. Deployment of Web Role



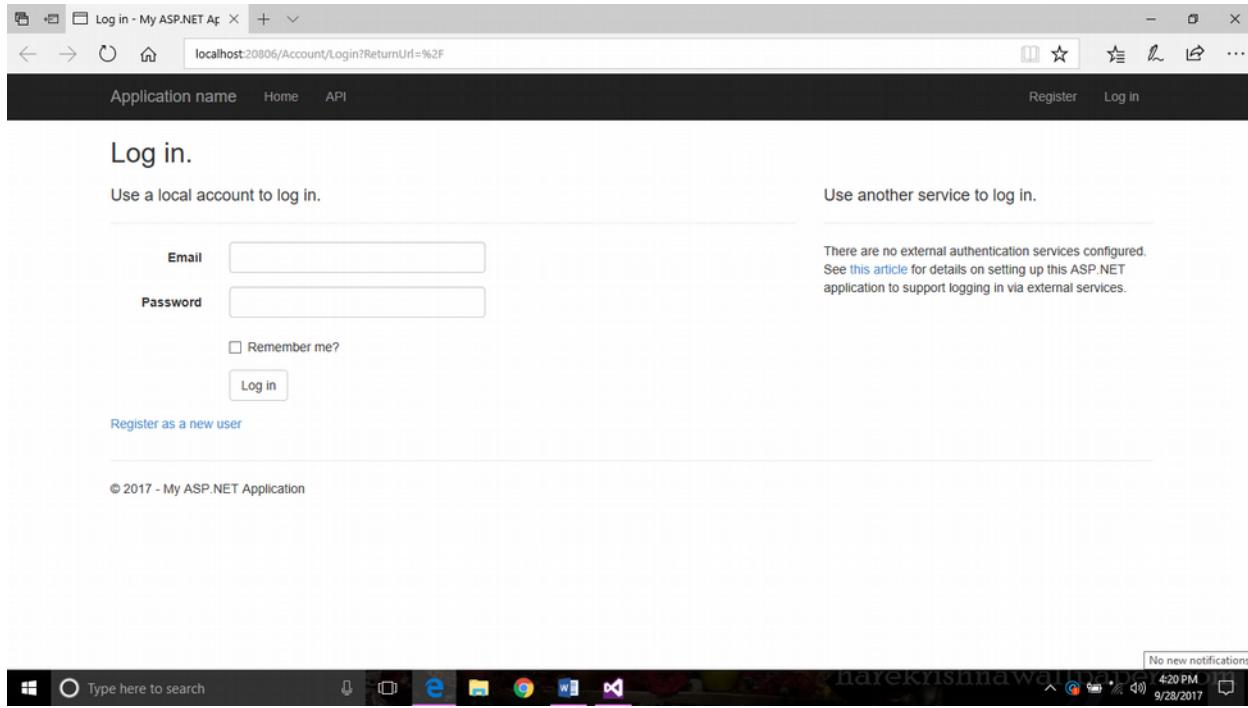
## 5. Worker Role Started



## 6. IIS Server Showing Web Application



## 7. Web Application



### System Log :

```
'WaIISHost.exe' (CLR v4.0.30319: WaIISHost.exe): Loaded
'C:\WINDOWS\Microsoft.Net\assembly\GAC_64\mscorlib\v4.0_4.0.0.0__b77a5c561934e089\mscorlib.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
Microsoft.WindowsAzure.ServiceRuntime Information: 100 : Role environment . INITIALIZING
Microsoft.WindowsAzure.ServiceRuntime Information: 100 : Role environment . INITIALED
RETURNED. HRESULT=0
'iisexpress.exe' (CLR v4.0.30319: DefaultDomain): Loaded
'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Web.ApplicationServices\v4.0_4.0.0.0__31bf3856ad364e35\System.Web.ApplicationServices.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
Microsoft.WindowsAzure.ServiceRuntime Information: 101 : Role environment . INITIALIZED
Microsoft.WindowsAzure.ServiceRuntime Information: 200 : Role entrypoint . CALLING
OnStart(): BE_IT_CC.WebRole
Microsoft.WindowsAzure.ServiceRuntime Information: 202 : Role entrypoint . COMPLETED
OnStart(): BE_IT_CC.WebRole
Microsoft.WindowsAzure.ServiceRuntime Verbose: 500 : Role instance status check starting
Microsoft.WindowsAzure.ServiceRuntime Information: 203 : Role entrypoint . CALLING Run()
BE_IT_CC.WebRole
Microsoft.WindowsAzure.ServiceRuntime Verbose: 502 : Role instance status check succeeded:
Ready
The thread 0x428 has exited with code 0 (0x0).
Microsoft.WindowsAzure.ServiceRuntime Verbose: 500 : Role instance status check starting
Microsoft.WindowsAzure.ServiceRuntime Verbose: 502 : Role instance status check succeeded:
Ready
The thread 0xd60 has exited with code 0 (0x0).
Microsoft.WindowsAzure.ServiceRuntime Verbose: 500 : Role instance status check starting
Microsoft.WindowsAzure.ServiceRuntime Verbose: 502 : Role instance status check succeeded:
Ready
Microsoft.WindowsAzure.ServiceRuntime Verbose: 500 : Role instance status check starting
Microsoft.WindowsAzure.ServiceRuntime Verbose: 502 : Role instance status check succeeded:
Ready
```

```

Microsoft.WindowsAzure.ServiceRuntime Verbose: 500 : Role instance status check starting
Microsoft.WindowsAzure.ServiceRuntime Verbose: 502 : Role instance status check succeeded:
Ready
The program '[7844] WaIISHost.exe' has exited with code 0 (0x0).
The program '[9624] iisexpress.exe' has exited with code -1 (0xffffffff).

```

## 14. Learning Outcomes Achieved

1. Successful Coding for a Web Role and Compile in Visual Studio.NET
2. Deployment of the web role on Azure Compute Emulator
3. Analysis of the web role deployment states
4. Configuration of Web Application for Web role Access

## 15. Conclusion:

1. Applications of the studied technique in industry
  - a. Implementation of compute tasks over the cloud
  - b. Implementation of SaaS based software development
2. Engineering Relevance
  - a. Concept of SaaS is implemented using web and worker roles
  - b. Using MS Azure framework for developing web role
3. Skills Developed
  - a. Web role coding using C#.NET
  - b. Testing and deployment of web roles

## 16. Experiment/Assignment Evaluation

SR	Parameters	Weight	Excellent	Good	Average	Poor	Not as per requirement
		Scale Factor ->	5	4	3	2	0
1	Technical Understanding	25					
2	Performance / Execution	25					
3	Question Answers	20					
4	Punctuality	20					
5	Presentation	10					
Total out of 100 -->		$\Sigma (\text{Weight} * \text{Scale Factor}) / 5 =$ _____					

## References :

[1] "Upcoming Name Change for Windows Azure". Microsoft Azure. March 24, 2014. Archived from the original on March 24, 2014. Retrieved August 29, 2016.

- [2] Tharakan, Anya George and Dastin, Jeffery (October 20, 2016). "Microsoft shares hit high as cloud business flies above estimates". Rueters. Thomson Reuters. Retrieved October 21, 2016.
- [3] Directory of Azure Cloud Services, Microsoft.com
- [4] "How to monitor Microsoft Azure VMs". Datadog. Retrieved September 16, 2016.
- [5] "Meet Windows Azure event June 2012". Weblogs.asp.net. June 7, 2012. Retrieved June 27,
- [6] <https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-choose-me>

## Viva Questions

1. What is a WebRole?
2. What is a IaaS?
3. Give Examples of other Cloud Vendors?
4. What is worker role, how it's different than webrole?

## Teachers Interaction with Students

1. Concept of IaaS
2. Azure Cloud Configuration
3. Cloud Application Deployment on cloud

 <b>Hope Foundation's</b> <b>Finolex Academy of Management and Technology, Ratnagiri</b> <b>Information Technology Department</b>			
Subject name: Cloud Computing			Subject Code: BEITC702
Class	BE IT	Semester – VIII (CBGS)	Academic year: 2018-19
Name of Student	<b>QUIZ Score :</b>		
Roll No		Assignment/Experiment No.	
<b>Title:</b> To demonstrate Azure Web role and Worker role			

<p><b>1. Course objectives applicable</b></p> <p><b>COB1.</b>Get familiar with cloud computing fundamentals, Virtualization, and types of virtualization</p> <p><b>COB4.</b>Study cloud architecture, explain difference with traditional approach and programming support for cloud.</p>
<p><b>2. Course outcomes applicable:</b></p> <p><b>CO1 :</b> Define cloud computing, list various avenues for cloud computing applications, Explain different cloud computing fundamental and virtualization techniques and its role in cloud computing.</p> <p><b>CO4-</b>Explain cloud architecture, give basic consideration for design cloud computing environment. Explain difference between traditional and cloud programming model. Design, program and deploy application on public cloud such as Google App Engine, AWS, Microsoft Azure</p>
<p><b>3. Learning Objectives:</b></p> <ol style="list-style-type: none"> <li>1. To write a Web Role and Compile in Visual Studio.NET</li> <li>2. Configure worker role</li> <li>3. Deploy the web and worker role on Cloud Compute and Storage emulator</li> <li>4. Asses Runtime progress</li> </ol>
<p><b>4. Practical applications of the assignment/experiment:</b></p> <p>Compute and Storage are the major resource groups used by cloud services</p>
<p><b>5. Prerequisites:</b></p> <ol style="list-style-type: none"> <li>1. Knowledge of Azure compute cloud and C# programming</li> <li>2. Internet Access</li> <li>3. VS 2017 compiler</li> <li>4. Azure Components in VS 2017</li> </ol>
<p><b>6. Hardware Requirements:</b></p> <ol style="list-style-type: none"> <li>1. Internet Access with Browser</li> <li>2. PC to install VS2017 Compiler</li> </ol>
<p><b>7. Software Requirements:</b></p> <p>Browser like Chrome, Internet Explorer Edge, VS 2017 Compiler Setup</p>
<p><b>8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)</b></p> <ol style="list-style-type: none"> <li>1. What is web and worker role?</li> </ol>

2. What is an Emulator?

**9. Experiment/Assignment Evaluation:**

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A <u>or</u> any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2
<b>Date of performance (DOP)</b>		<b>Total marks obtained</b>	<b>10</b>
<b>Date of checking (DOC)</b>		<b>Signature of teacher</b>	

## 9. Theory: <Preferably given as handwritten work for students>

Microsoft Azure (formerly Windows Azure) is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service and infrastructure as a service and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.

Web and Worker roles are nearly identical:

More control also means less ease of use. Unless you need the additional control options, it's typically quicker and easier to get a web application up and running in the Web Apps feature of App Service compared to Azure Cloud Services.

There are two types of Azure Cloud Services roles. The only difference between the two is how your role is hosted on the VMs:

- **Web role:** Automatically deploys and hosts your app through IIS.
- **Worker role:** Does not use IIS, and runs your app standalone.

For example, a simple application might use just a single web role, serving a website. A more complex application might use a web role to handle incoming requests from users, and then pass those requests on to a worker role for processing. (This communication might use [Azure Service Bus](#) or [Azure Queue storage](#).)

As the preceding figure suggests, all the VMs in a single application run in the same cloud service. Users access the application through a single public IP address, with requests automatically load balanced across the application's VMs. The platform [scales and deploys](#) the VMs in an Azure Cloud Services application in a way that avoids a single point of hardware failure.

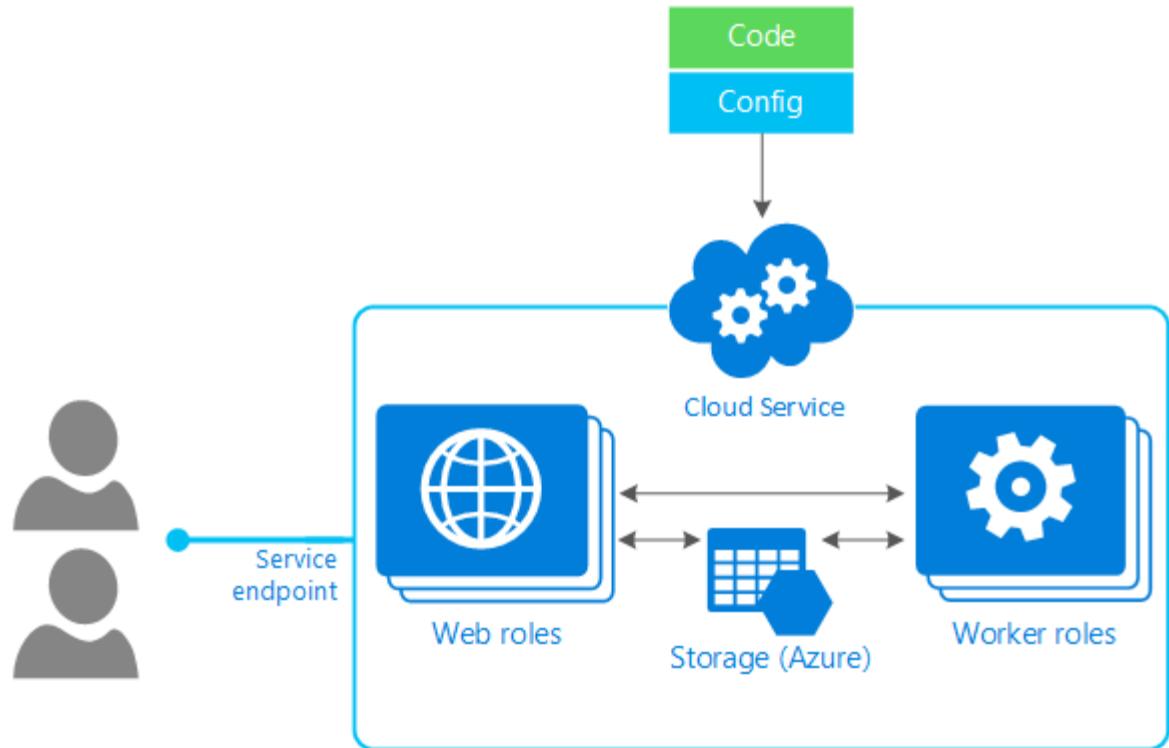
Even though applications run in VMs, it's important to understand that Azure Cloud Services provides PaaS, not infrastructure as a service (IaaS). Here's one way to think about it. With IaaS, such as Azure Virtual Machines, you first create and configure the environment your application runs in. Then you deploy your application into this environment. You're responsible for managing much of this world, by doing things such as deploying new patched versions of the operating system in each VM. In PaaS, by contrast, it's as if the environment already exists. All you have to do is deploy your application. Management of the platform it runs on, including deploying new versions of the operating system, is handled for you.

### Scaling and management

With Azure Cloud Services, you don't create virtual machines. Instead, you provide a configuration file that tells Azure how many of each you'd like, such as "three web role instances" and "two worker role instances." The platform then creates them for you. You still choose [what size](#) those backing VMs should be, but you don't explicitly create them yourself. If your application needs to handle a greater load, you can ask for more VMs, and Azure creates those instances. If the load decreases, you can shut down those instances and stop paying for them.

An Azure Cloud Services application is typically made available to users via a two-step process. A developer first [uploads the application](#) to the platform's staging area. When the developer is ready to make the application live, they use the Azure portal to swap staging with production. This [switch between staging](#)

[and production](#) can be done with no downtime, which lets a running application be upgraded to a new version without disturbing its users.



- **10. Precautions :**

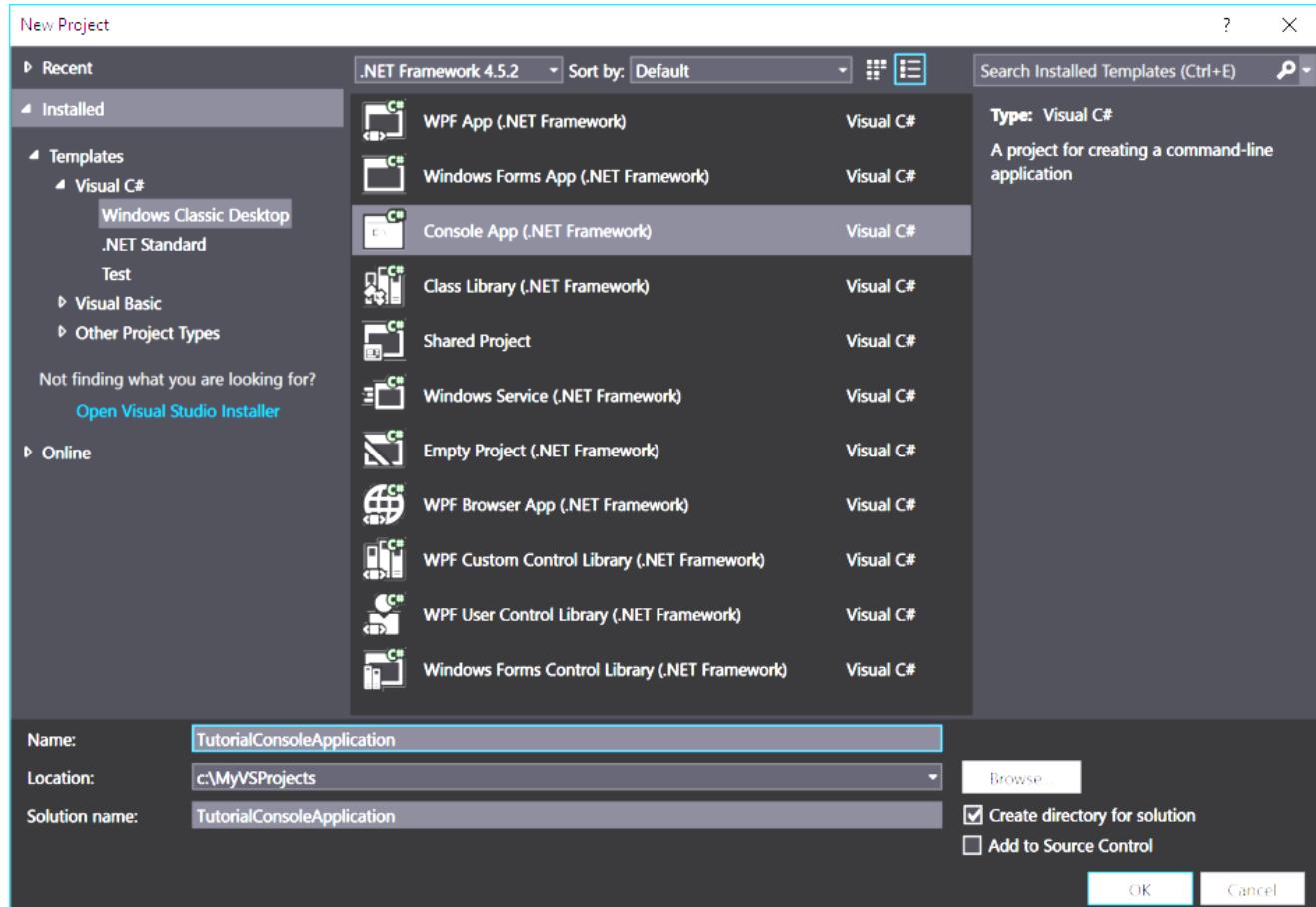
1. Shut Down the webrole if you are deploying on Azure Cloud as it will keep on charging in your monthly bill
2. Dispose the blob storage if not in use

### **11. Installation Steps / Performance Steps –**

#### **Create a Windows console application project**

In Visual Studio, create a new Windows console application. The following steps show you how to create a console application in Visual Studio 2017, however, the steps are similar in other versions of Visual Studio.+

1. Select **File > New > Project**
2. Select **Installed > Templates > Visual C# > Windows Classic Desktop**
3. Select **Console App (.NET Framework)**
4. Enter a name for your application in the **Name:** field
5. Select **OK**



All code examples in this tutorial can be added to the `Main()` method of your console application's `Program.cs` file.

You can use the Azure Storage Client Library in any type of .NET application, including an Azure cloud service or web app, and desktop and mobile applications. In this guide, we use a console application for simplicity.

## Use NuGet to install the required packages

There are two packages you need to reference in your project to complete this tutorial:

- [Microsoft Azure Storage Client Library for .NET](#): This package provides programmatic access to data resources in your storage account.
- [Microsoft Azure Configuration Manager library for .NET](#): This package provides a class for parsing a connection string in a configuration file, regardless of where your application is running.

You can use NuGet to obtain both packages. Follow these steps:

1. Right-click your project in **Solution Explorer** and choose **Manage NuGet Packages**.

2. Search online for "WindowsAzure.Storage" and click **Install** to install the Storage Client Library and its dependencies.
3. Search online for "WindowsAzure.ConfigurationManager" and click **Install** to install the Azure Configuration Manager.

```

4. using System;
5. using System.Collections.Generic;
6. using System.Diagnostics;
7. using System.Linq;
8. using System.Net;
9. using System.Threading;
10. using Microsoft.ServiceBus;
11. using Microsoft.ServiceBus.Messaging;
12. using Microsoft.WindowsAzure;
13. using Microsoft.WindowsAzure.ServiceRuntime;
14.
15. namespace WorkerRoleWithSBQueue1
16. {
17.     public class WorkerRole : RoleEntryPoint
18.     {
19.         // The name of your queue
20.         const string QueueName = "ProcessingQueue";
21.
22.         // QueueClient is thread-safe. Recommended that you cache
23.         // rather than recreating it on every request
24.         QueueClient Client;
25.         ManualResetEvent CompletedEvent = new ManualResetEvent(false);
26.
27.         public override void Run()
28.         {
29.             Trace.WriteLine("Starting processing of messages");
30.             Trace.WriteLine("Startet the Worker Role!!!");
31.
32.             // Initiates the message pump and callback is invoked for each
33.             // message that is received, calling close on the client will stop the pump.
34.             Client.OnMessage((receivedMessage) =>
35.             {
36.                 try
37.                 {
38.                     // Process the message
39.                     Trace.WriteLine("Processing Service Bus message: " +
40.                     receivedMessage.SequenceNumber.ToString());
41.                 }
42.                 catch
43.                 {
44.                     // Handle any message processing specific exceptions
45.                     here
46.                 }
47.             });
48.
49.         public override bool OnStart()
50.         {
51.             // Set the maximum number of concurrent connections
52.             ServicePointManager.DefaultConnectionLimit = 12;
53.
54.             // Create the queue if it does not exist already

```

```

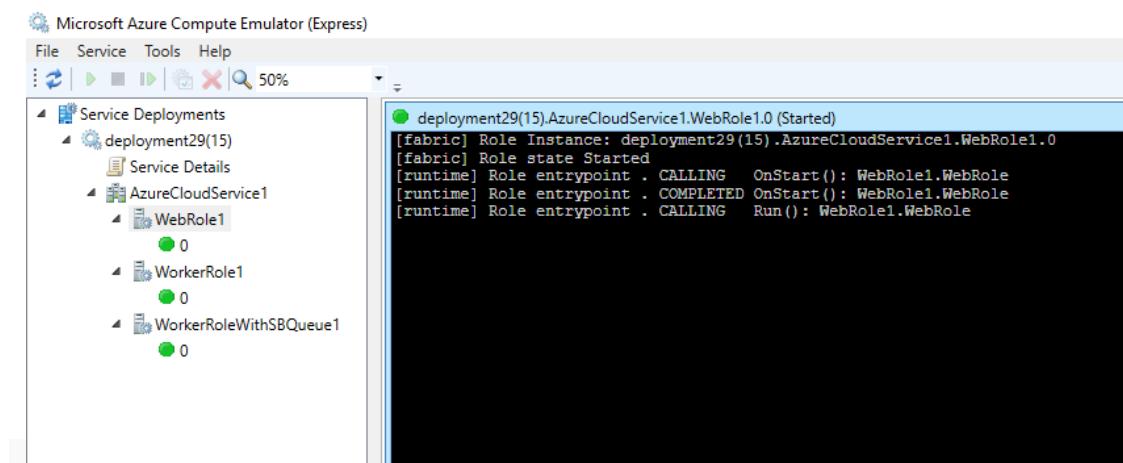
55.         string connectionString =
56.             CloudConfigurationManager.GetSetting("Microsoft.ServiceBus.ConnectionString");
57.             var namespaceManager =
58.                 NamespaceManager.CreateFromConnectionString(connectionString);
59.                     if (!namespaceManager.QueueExists(QueueName))
60.                     {
61.                         namespaceManager.CreateQueue(QueueName);
62.                     }
63.                     // Initialize the connection to Service Bus Queue
64.                     Client = QueueClient.CreateFromConnectionString(connectionString,
65.                         QueueName);
66.                     Trace.WriteLine("On Start Event!! - FAMT BE IT");
67.                     return base.OnStart();
68.                 }
69.                 public override void OnStop()
70.                 {
71.                     // Close the connection to Service Bus Queue
72.                     Client.Close();
73.                     CompletedEvent.Set();
74.                     base.OnStop();
75.                 }
76. }

```

## 12.Observations

1. C# .Net and ASP.NET are used for coding, the ASP web page is hosted by IIS
2. Worker Role is emulated on Azure Compute Emulator

## 13. Results:



```
[fabric] Role Instance: deployment29(15).AzureCloudService1.WorkerRole1.0
[fabric] Role state Started
[runtime] Role entrypoint . CALLING OnStart(): WorkerRole1.WorkerRole
[WaWorkerHost.exe] WorkerRole1 has been started
[runtime] Role entrypoint . COMPLETED OnStart(): WorkerRole1.WorkerRole
[runtime] Role entrypoint . CALLING Run(): WorkerRole1.WorkerRole
[WaWorkerHost.exe] WorkerRole1 is running
[WaWorkerHost.exe] Working
[fabric] Role state Unresponsive
[fabric] Role state Stopped
[fabric] Role state Busy
```

```
[fabric] Role Instance:
deployment29(15).AzureCloudService1.WorkerRoleWithSBQueue1.0
[fabric] Role state Busy
[runtime] Role entrypoint . CALLING OnStart():
WorkerRoleWithSBQueue1.WorkerRole
Loaded "Microsoft.WindowsAzure.ServiceRuntime, Version=2.7.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35"
Getting "Microsoft.ServiceBus.ConnectionString" from ServiceRuntime: PASS
(Endpoint=sb://[your
namespace].servicebus.windows.net;SharedAccessKeyName=RootManageSharedAccessK
ey;SharedAccessKey=[your key]). 
[fabric] Role state Unresponsive
```

```
[fabric] Role state Stopped  
[fabric] Role state Suspended  
[fabric] Role state Busy
```

## 14. Learning Outcomes Achieved

1. Students coded Web Role and Compiled in Visual Studio.NET
2. Students Configured worker role
3. Students Deployed the web and worker role on Cloud Compute and Storage emulator
4. The Runtime progress of Web and worker role was assesed

## 15. Conclusion:

1. **Applications of the studied technique in industry**
  - a. Worker role act as process running on compute cloud
  - b. Web roles are similar to web services
2. **Engineering Relevance**
  - a. Web and worker role are main compute related objects
  - b. All leading cloud storage providers have parallel technology for web and worker role
3. **Skills Developed**
  - a. Configuration of cloud components
  - b. Assessment of progress of web and worker role deployment

## References :

- [1] "<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-dotnet-how-to-use-blobs>
- [2] <https://docs.microsoft.com/en-us/azure/storage/common/storage-samples-dotnet>

## Teachers Interaction with Students

1. Concept of Cloud based resources
2. Azure compute and storage cloud emulation
3. Cloud Application Deployment on emulator



**Finolex Academy of Management and Technology, Ratnagiri**  
**Department of Information Technology**

<b>Subject:</b>	<b>Cloud Computing (BEITL702)</b>		
<b>Class:</b>	<b>BE IT / Semester – VII (CBGS) / Academic year: 2017-2018</b>		
<b>Student Name</b>			
<b>Roll No:</b>		<b>Date of performance (DOP) :</b>	
<b>Assignment/Experiment No:</b>	<b>9</b>	<b>Date of checking (DOC) :</b>	
<b>Title:</b> Deep Web and Big Data			
	<b>Marks</b>		<b>Teacher's Signature:</b>

**1. Aim:** To demonstrate Azure Blob Storage management

**2. Prerequisites:**

1. Knowledge of C#.NET, ASP.NET
2. Internet Access
3. Azure Account

**3. Hardware Requirements:**

1. Internet Access with Browser
2. PC with 4GB RAM

**4. Software Requirements:**

Visual Studio 2015 Community Edition

**5. Learning Objectives:**

1. To Understand the concept of Big Data
2. To understand the concept of Deep Web
3. Find the correlation between the deep web and big data
4. To discuss Future aspects of data analysis

**6. Course Objectives Applicable:** CO3, CO5

**7. Program Outcomes Applicable:** PO1, PO2, PO4, PO5, PO10

**8. Program Education Objective Applicable:** PEO 2, PEO 3, PEO6

## **14. Learning Outcomes Achieved**

### **15. Conclusion:**

- 1. Applications of the studied technique in industry**
  - a.
- 2. Engineering Relevance**
  - a.
- 3. Skills Developed**
  - a.

## **16. Experiment/Assignment Evaluation**

SR	Parameters	Weight	Excellent	Good	Average	Poor	Not as per requirement
		Scale Factor ->	5	4	3	2	0
1	Technical Understanding	25					
2	Performance / Execution	25					
3	Question Answers	20					
4	Punctuality	20					
5	Presentation	10					
	Total out of 100 -->		$\Sigma (\text{Weight} * \text{Scale Factor}) /5 = \underline{\hspace{1cm}}$				

### **References :**





**Finolex Academy of Management and Technology, Ratnagiri**  
**Department of Information Technology**

<b>Subject:</b>	<b>Cloud Computing (BEITL702)</b>		
<b>Class:</b>	<b>BE IT / Semester – VII (CBGS) / Academic year: 2017-2018</b>		
<b>Student Name</b>			
<b>Roll No:</b>		<b>Date of performance (DOP) :</b>	
<b>Assignment/Experiment No:</b>	<b>10</b>	<b>Date of checking (DOC) :</b>	
<b>Title:</b> Microsoft Azure NoSQL (MongoDB)			
<b>Marks</b>		<b>Teacher's Signature:</b>	

**1. Aim:** To demonstrate Azure Blob Storage management

**2. Prerequisites:**

1. Knowledge of Azure Cloud
2. Internet Access
3. MongoDB

**3. Hardware Requirements:**

1. Internet Access with Browser
2. PC with 4GB RAM

**4. Software Requirements:**

Microsoft Azure Cloud Access

**5. Learning Objectives:**

1. To Understand the concept of NoSQL
2. To understand the concept of MongoDB
3. To explain how Azure is providing security to MongoDB

**6. Course Objectives Applicable:** CO3, CO4

**7. Program Outcomes Applicable:** PO1, PO2, PO4, PO5, PO10

**8. Program Education Objective Applicable:** PEO 2, PEO 3, PEO6

## **14. Learning Outcomes Achieved**

### **15. Conclusion:**

- 1. Applications of the studied technique in industry**
  - a.
- 2. Engineering Relevance**
  - a.
- 3. Skills Developed**
  - a.

## **16. Experiment/Assignment Evaluation**

SR	Parameters	Weight	Excellent	Good	Average	Poor	Not as per requirement
		Scale Factor ->	5	4	3	2	0
1	Technical Understanding	25					
2	Performance / Execution	25					
3	Question Answers	20					
4	Punctuality	20					
5	Presentation	10					
	Total out of 100 -->		$\Sigma (\text{Weight} * \text{Scale Factor}) /5 = \underline{\hspace{1cm}}$				

### **References :**

