

**Microsoft Azure**



**Splunk**



**Docker**



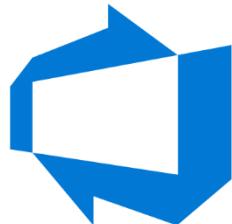
**Kubernetes**



**Azure Kubernetes Services**



**Azure DevOps**



**Azure DevOps**

**SQL**



## **AGENDA**

### **Azure**

- ★ **What is Cloud Computing?**
- ★ **What is Microsoft Azure?**
- ★ **Microsoft Azure Fundamentals**
- ★ **Microsoft Azure Services**
- ★ **Accessing Microsoft Azure Portal**

## ★ Creating Various Services

- **Virtual Machine in Linux and Windows**
- **Storage Account**
- **Scale Sets**
- **Virtual Machine Deep Dive**
- **SQL Database**
- **App Services**
- **Virtual Networks**
  - Subnets
  - Network security groups
- **Custom Website in Azure App Service**
- **Custom Website in Ubuntu 18.04**
- **File share and Service Mounting in Linux**

## **SPLUNK**

### ★ What is Splunk?

### ★ Splunk Installation in Linux and Windows

### ★ Loading data into Splunk

### ★ Operations in Splunk

- **Searching**
  - Visualization
  - Creating Dashboard
- **Querying**
- **Adding Index**
- **Adding Role**
- **Adding User**

### ★ Forwarders

- **Heavy Forwarder**
- **Universal Forwarder**

### ★ Pivots

## **DOCKER**

- ★ Docker Installation
- ★ Docker Concepts
- ★ Docker hub
- ★ Docker Images
- ★ Docker Containers
- ★ Docker Architecture
- ★ Docker File
- ★ Docker Public Repositories
- ★ Docker Managing Ports
- ★ Docker Private Registry
- ★ Docker Storage
- ★ Docker Volumes
- ★ Docker Networking
- ★ Docker Toolbox
- ★ Docker Logging

## **KUBERNETES**

- ★ Kubernetes Concepts
- ★ Minikube
- ★ Kubernetes Pods
- ★ Kubernetes Deployments
- ★ Kubernetes Services
- ★ Kubernetes Labels
- ★ Health Checks
- ★ Liveness Probe
- ★ Kubernetes Secrets
- ★ Kubernetes Volumes
- ★ Microservices Architecture

- ★ **High Availability**
- ★ **Taints & Tolerants**

## **AZURE KUBERNETES SERVICES**

- ★ **Azure Kubernetes Services Concepts**
  - **Kubernetes Cluster Architecture**
  - **Control Plane**
  - **Nodes & Node Pools**
  - **Resource & Reservations**
  - **YAML Manifests**
- ★ **Azure CLI**
  - **Creating a AKS cluster for a Voting Application**
  - **Deploying the Application**
  - **Testing the Application**
  - **Running the Application**

## **AZURE DEVOPS**

- ★ **Azure DevOps Concepts**
- ★ **Azure Pipelines**
- ★ **Azure Boards**
- ★ **Azure Repos**
- ★ **Azure Artifacts**
- ★ **Azure Resource Manager Template**
- ★ **Web Hooks and API Integration**
- ★ **UI Integration using DevOps**
- ★ **Version Control**
- ★ **Springs**

## **SQL**

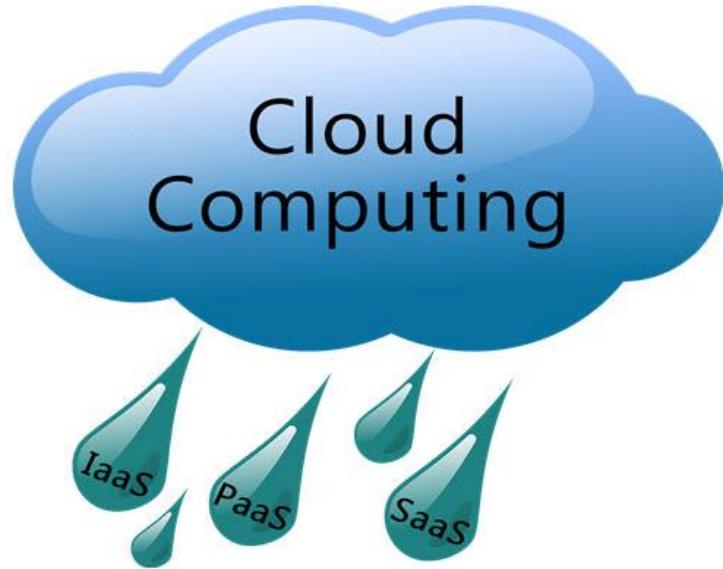
- ★ **Introduction to SQL and SSMS**

- ★ **Ranking**
- ★ **Built-in Functions & User defined Functions**
- ★ **Pivoting**
- ★ **Stored Procedures**
- ★ **Server Login & Permissions**
- ★ **Importing & Exporting of Data**
- ★ **Backups**
- ★ **Data Restoration & Data Recovery Methods**
- ★ **Indexes**
- ★ **Deadlock Scenario**
- ★ **Isolation Levels**
- ★ **Query store in memory**
- ★ **Exporting & Importing data b/w Azure & SSMS**
- ★ **Azure Synapse**

## **MICROSOFT AZURE**

### **Introduction to Cloud Computing**

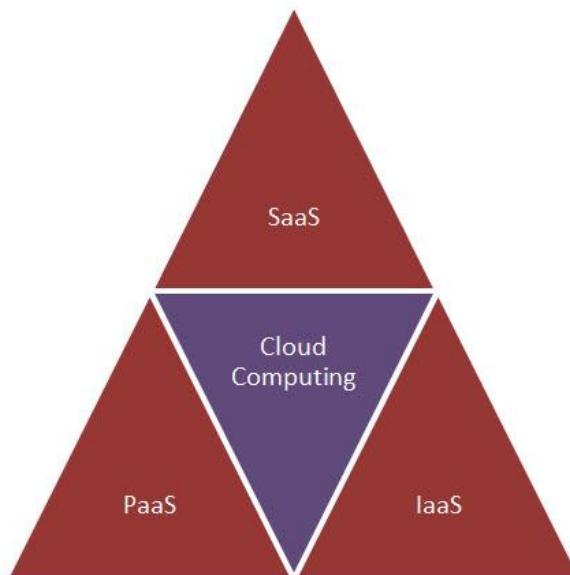
**Cloud Computing** is defined as storing and accessing of data and computing services over the internet. It doesn't store any data on your personal computer. It is the on-demand availability of computer services like servers, data storage, networking, databases, etc. The main purpose of cloud computing is to give access to data centers to many users. Users can also access data from a remote server.



## **Cloud Computing Services**

The three major Cloud Computing Offerings are

- **Software as a Service (SaaS)**
- **Platform as a Service (PaaS)**
- **Infrastructure as a Service (IaaS)**

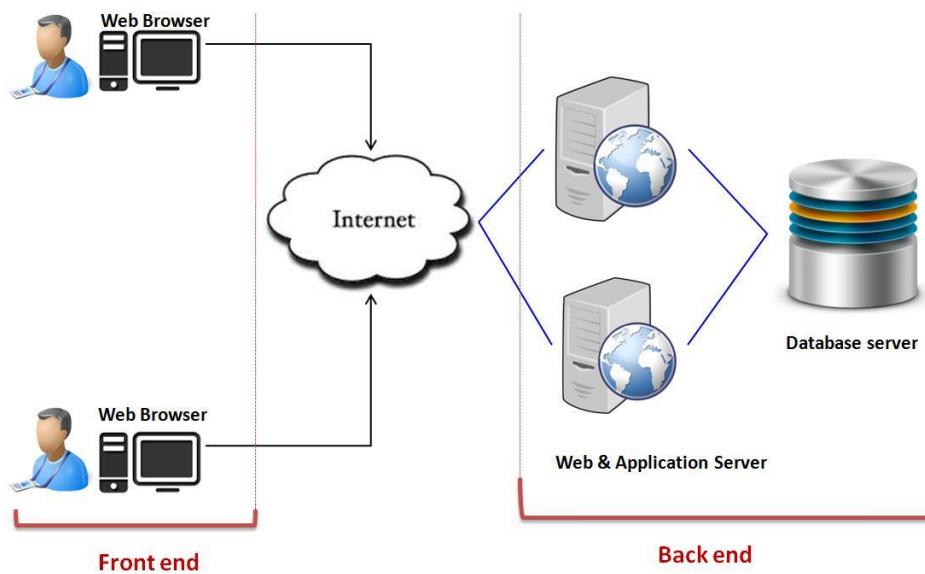


## Cloud Computing Architecture

**Cloud Computing Architecture** is a combination of components required for a Cloud Computing service. A Cloud computing architecture consists of several components like a front-end platform, a back-end platform or servers, a network or internet service, and a cloud-based delivery service.

Let's have a look into Cloud Computing and see what Cloud Computing is made of. Cloud computing comprises of two components front end and back end. Front end consists client part of cloud computing system.

It comprises of interfaces and applications that are required to access the Cloud Computing or Cloud Programming platform.



## Azure Fundamentals

### What is Microsoft Azure?

The Azure cloud platform is more than 200 products and cloud services designed to help you bring new solutions to life—to solve today's challenges and create the future. Build, run and manage applications across multiple clouds, on-premises and at the edge, with the tools and frameworks of your choice.

## **What does Azure offer?**

With help from Azure, you have everything you need to build your next great solution. The following table lists several of the benefits that Azure provides, so you can easily invent with purpose.

**Be ready for the future:** Continuous innovation from Microsoft supports your development today and your product visions for tomorrow.

**Build on your terms:** You have choices. With a commitment to open source, and support for all languages and frameworks, build how you want and deploy where you want to.

**Operate hybrid seamlessly:** On-premises, in the cloud, and at the edge--we'll meet you where you are. Integrate and manage your environments with tools and services designed for a hybrid cloud solution.

**Trust your cloud:** Get security from the ground up, backed by a team of experts, and proactive compliance trusted by enterprises, governments, and startups.

## **What is the Azure portal?**

The Azure portal is a web-based, unified console that provides an alternative to command-line tools. With the Azure portal, you can manage your Azure subscription by using a graphical user interface. You can:

- Build, manage, and monitor everything from simple web apps to complex cloud deployments.
- Create custom dashboards for an organized view of resources.
- Configure accessibility options for an optimal experience.

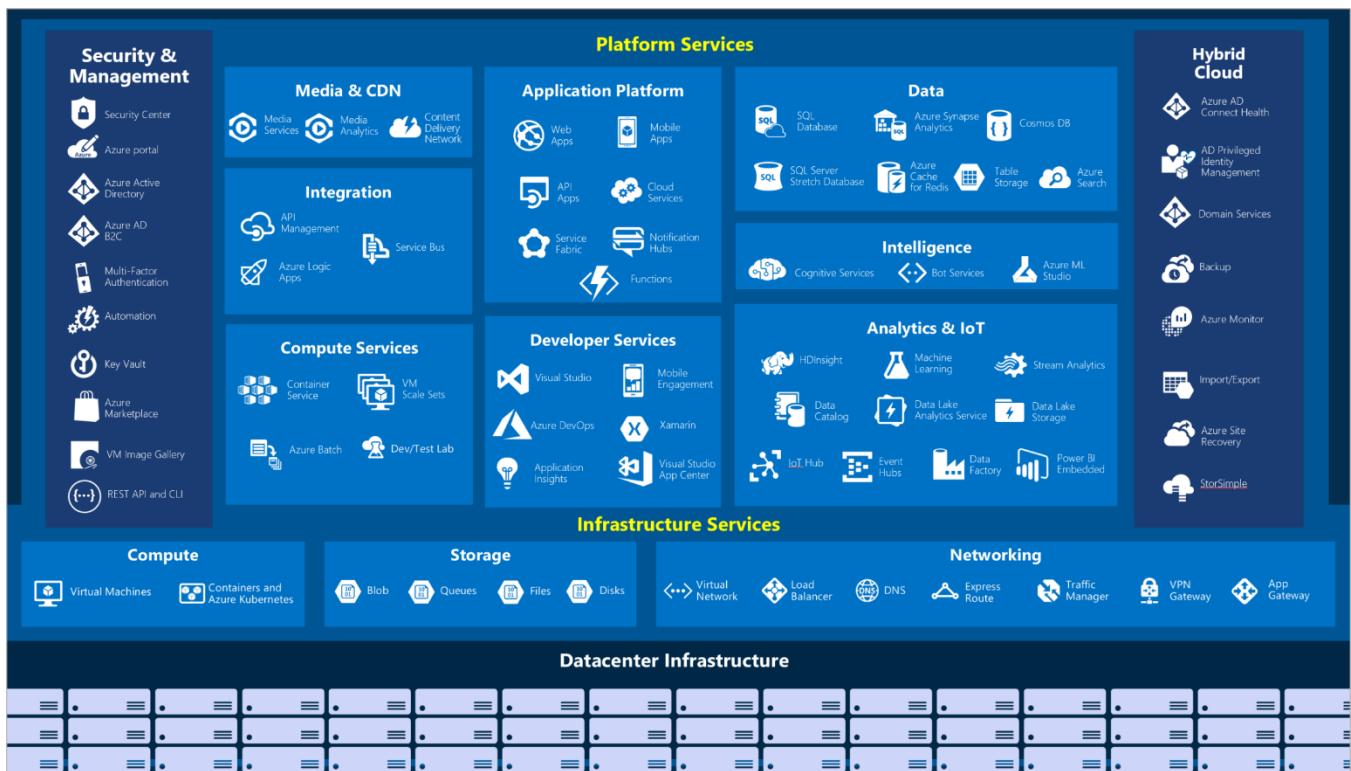
We can open the Azure portal using the link:

[Microsoft Azure portal - https://portal.azure.com](https://portal.azure.com)

The screenshot shows the Microsoft Azure portal interface. At the top, there's a blue header bar with the 'Microsoft Azure' logo, a search bar containing 'Search resources, services, and docs (G+ /)', and a user profile icon. Below the header, the main content area is titled 'Azure services'. It features a grid of service icons: 'Create a resource' (with a plus sign), 'Resource groups', 'Virtual machines', 'App Services', 'Storage accounts', 'SQL databases', 'Azure Database for PostgreSQL...', 'Azure Cosmos DB', 'Kubernetes services', and a large blue arrow pointing right labeled 'More services'. On the left side, there's a 'Navigate' sidebar with links for 'Subscriptions' (key icon), 'Resource groups' (cube icon), 'All resources' (grid icon), and 'Dashboard' (dash icon). A vertical scroll bar is visible on the right side of the main content area.

## Azure Services

- Compute
- Networking
- Storage
- Mobile
- Databases
- Web
- Internet of Things (IoT)
- Big data
- AI
- DevOps



## Getting started with Azure Account

You can purchase Azure access directly from Microsoft by signing up on the [Azure website](#) or through a Microsoft representative. You can also purchase Azure access through a Microsoft partner. Cloud Solution Provider partners offer a range of complete managed-cloud solutions for Azure

[https://docs.microsoft.com/en-us/learn/modules/create-an-azure-account/-](https://docs.microsoft.com/en-us/learn/modules/create-an-azure-account/)

## Creating a Virtual Machine in MS Azure

### Virtual Machine

Azure Virtual Machines (VM) is one of several types of [on-demand, scalable computing resources](#) that Azure offers. Typically, you choose a VM when you need more control over the computing environment than the other choices offer. This article gives you

information about what you should consider before you create a VM, how you create it, and how you manage it.

An Azure VM gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it. However, you still need to maintain the VM by performing tasks, such as configuring, patching, and installing the software that runs on it.

Azure virtual machines can be used in various ways. Some examples are:

- **Development and test** – Azure VMs offer a quick and easy way to create a computer with specific configurations required to code and test an application.
- **Applications in the cloud** – Because demand for your application can fluctuate, it might make economic sense to run it on a VM in Azure. You pay for extra VMs when you need them and shut them down when you don't.
- **Extended datacenter** – Virtual machines in an Azure virtual network can easily be connected to your organization's network.

## **Basic Linux Commands**

### **1. pwd command**

Use the **pwd** command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash (**/**). An example of an absolute path is **/home/username**.

### **2. cd command**

To navigate through the Linux files and directories, use the **cd** command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.

### **3. ls command**

The **ls** command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

### **4. cat command**

**cat** (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (sdout). To run this command, type **cat** followed by the file's name and its extension. For instance: **cat file.txt**.

## **5. cp command**

Use the **cp** command to copy files from the current directory to a different directory. For instance, the command **cp scenery.jpg /home/username/Pictures** would create a copy of **scenery.jpg** (from your current directory) into the **Pictures** directory.

## **6. mv command**

The primary use of the **mv** command is to move files, although it can also be used to rename files.

## **7. mkdir command**

Use **mkdir** command to make a new directory — if you type **mkdir Music** it will create a directory called **Music**.

## **8. rmdir command**

If you need to delete a directory, use the **rmdir** command. However, rmdir only allows you to delete empty directories.

## **9. rm command**

The **rm** command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to rmdir — use **rm -r**.

## **10. touch command**

The **touch** command allows you to create a blank new file through the Linux command line. As an example, enter **touch /home/username/Documents/Web.html** to create an HTML file entitled **Web** under the **Documents** directory.

## **11. locate command**

You can use this command to **locate** a file, just like the search command in Windows. What's more, using the **-i** argument along with this command will make it case-insensitive, so you can search for a file even if you don't remember its exact name.

## **12. find command**

Similar to the **locate** command, using **find** also searches for files and directories. The difference is, you use the **find** command to locate files within a given directory.

## **13. grep command**

Another basic Linux command that is undoubtedly helpful for everyday use is **grep**. It lets you search through all the text in a given file.

To illustrate, **grep blue notepad.txt** will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully.

## **14. sudo command**

Short for “**SuperUser Do**”, this command enables you to perform tasks that require administrative or root permissions. However, it is not advisable to use this command for daily use because it might be easy for an error to occur if you did something wrong.

## **15. df command**

Use **df** command to get a report on the system’s disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type **df -m**.

## **16. du command**

If you want to check how much space a file or a directory takes, the **du** (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the **-h** argument to the command line.

## **17. head command**

The **head** command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type **head -n 5 filename.ext**.

## **18. tail command**

This one has a similar function to the head command, but instead of showing the first lines, the **tail** command will display the last ten lines of a text file. For example, **tail -n filename.ext**.

## **19. diff command**

Short for difference, the **diff** command compares the contents of two files line by line.

## **20. tar command**

The **tar** command is the most used command to archive multiple files into a **tarball** — a common Linux file format that is similar to zip format, with compression being optional.

## **21. chmod command**

**chmod** is another Linux command, used to change the read, write, and execute permissions of files and directories. As this command is rather complicated, you can read [the full tutorial](#) in order to execute it properly.

## **22. chown command**

In Linux, all files are owned by a specific user. The **chown** command enables you to change or transfer the ownership of a file to the specified username. For instance, **chown linuxuser2 file.ext** will make **linuxuser2** as the owner of the **file.ext**.

## **23. jobs command**

**jobs** command will display all current jobs along with their statuses. A job is basically a process that is started by the shell.

## **24. kill command**

If you have an unresponsive program, you can terminate it manually by using the **kill** command. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

## **25. ping command**

Use the **ping** command to check your connectivity status to a server. For example, by simply entering **ping google.com**, the command will check whether you're able to connect to Google and also measure the response time.

## **26. wget command**

The Linux command line is super useful — you can even download files from the internet with the help of the **wget** command. To do so, simply type **wget** followed by the download link.

## **27. uname command**

The **uname** command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

## **28. top command**

As a terminal equivalent to Task Manager in Windows, the **top** command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

## **29. history command**

When you've been using Linux for a certain period of time, you'll quickly notice that you can run hundreds of commands every day. As such, running **history** command is particularly useful if you want to review the commands you've entered before.

## **30. man command**

Confused about the function of certain Linux commands? Don't worry, you can easily learn how to use them right from Linux's shell by using the **man** command. For instance, entering **man tail** will show the manual instruction of the tail command.

### **31. echo command**

This command is used to move some data into a file. For example, if you want to add the text, "Hello, my name is John" into a file called name.txt, you would type **echo Hello, my name is John >> name.txt**

### **32. zip, unzip command**

Use the **zip** command to compress your files into a zip archive, and use the **unzip** command to extract the zipped files from a zip archive.

### **33. hostname command**

If you want to know the name of your host/network simply type **hostname**. Adding a **-I** to the end will display the IP address of your network.

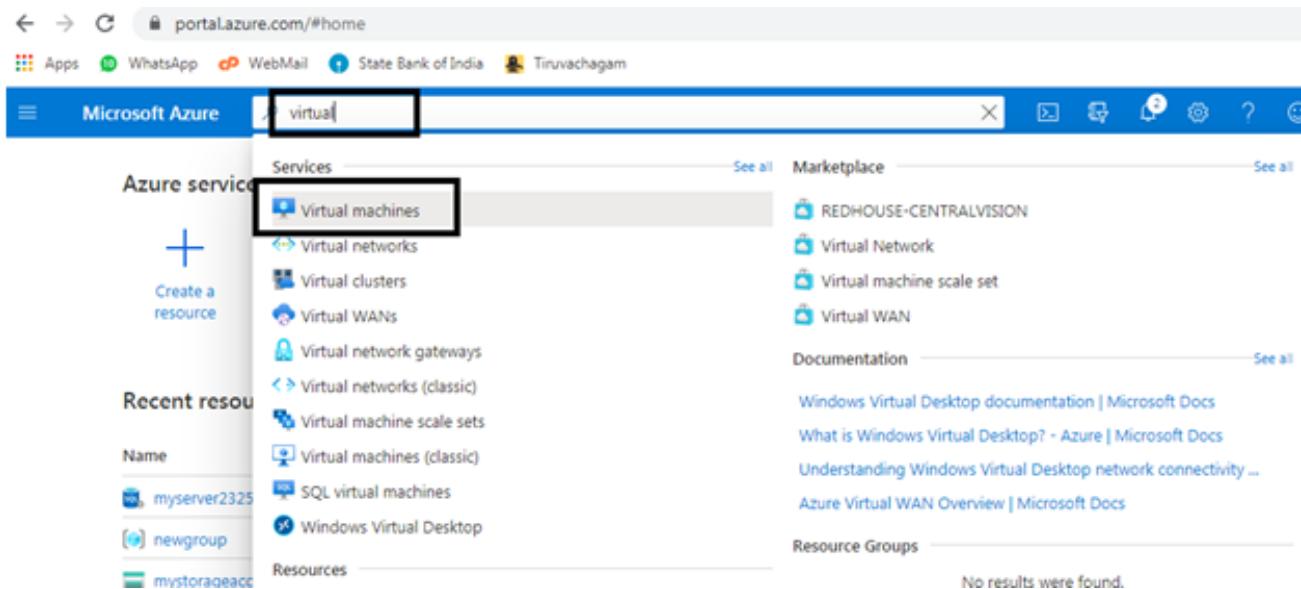
### **34. useradd, userdel command**

Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. **useradd** is used to create a new user, while **passwd** is adding a password to that user's account.

## **Creating a Virtual Machine in Ubuntu 18.04**

The below provided is the procedure for creating a virtual machine in Ubuntu 18.04

### **Step 1: Open the azure portal and search for virtual machines.**



The screenshot shows the Microsoft Azure portal interface. The search bar at the top contains the text "virtual". Below the search bar, the "Services" section is visible, with "Virtual machines" highlighted by a black rectangle. Other service options include "Virtual networks", "Virtual clusters", "Virtual WANs", "Virtual network gateways", "Virtual networks (classic)", "Virtual machine scale sets", "Virtual machines (classic)", "SQL virtual machines", and "Windows Virtual Desktop". To the right of the services, there are sections for "Marketplace", "Documentation", and "Resource Groups". The "Marketplace" section lists items like "REDHOUSE-CENTRALVISION", "Virtual Network", "Virtual machine scale set", and "Virtual WAN". The "Documentation" section links to "Windows Virtual Desktop documentation | Microsoft Docs", "What is Windows Virtual Desktop? - Azure | Microsoft Docs", "Understanding Windows Virtual Desktop network connectivity ...", and "Azure Virtual WAN Overview | Microsoft Docs". The "Resource Groups" section indicates "No results were found."

### **Step 2: Select Add and Virtual machine**

← → C portal.azure.com/#blade/HubsExtension/BrowseResourceBlade/resourceType/Microsoft.Computer

Apps WhatsApp WebMail State Bank of India Tiruvachagam

Microsoft Azure Search resources, services, and docs (G+)

Home > Virtual machines

ValueMomentum Inc.

+ Add Reservations Edit columns Refresh Try preview Assign tags ▶

+ Virtual machine

+ Start with a preexisting browser! This experience is faster and has improved sorting and filtering capability support for some columns such as maintenance status.

**Subscriptions:** All 2 selected – Don't see a subscription? Open Directory + Subscription settings

Filter by name... All subscriptions All resource groups All types [ ]

### Step 3: Create a new resource group

#### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize all your resources.

Subscription \* ⓘ

Visual Studio Enterprise Subscription – MPN

Resource group \* ⓘ

(None) Resource group

Create new

1

#### Instance details

Virtual machine name \* ⓘ

Region \* ⓘ

Availability options ⓘ

Review + create

< Previous

A resource group is a container that holds related resources for an Azure solution.

Name \*

newgroup2

2

OK

Cancel

3

|| teams.microsoft.com is sharing your screen

#### **Step 4: Fill the details accordingly**

Instance details

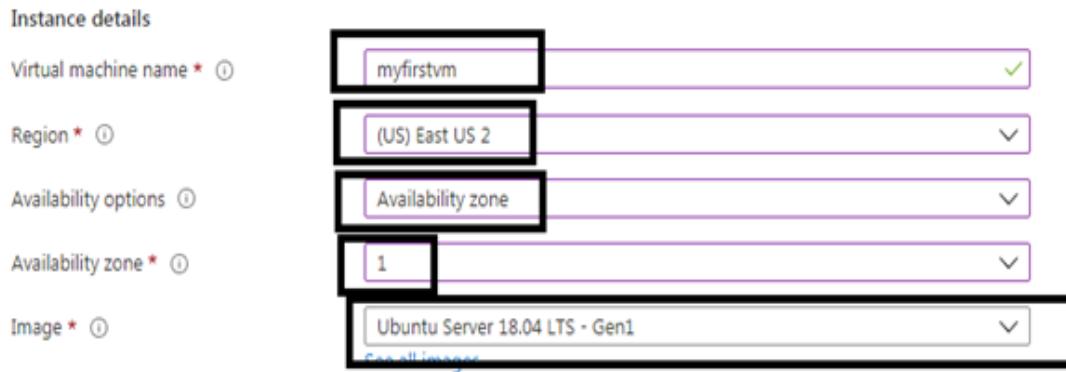
Virtual machine name \* ⓘ myfirstvm ✓

Region \* ⓘ (US) East US 2 ▾

Availability options ⓘ Availability zone ▾

Availability zone \* ⓘ 1 ▾

Image \* ⓘ Ubuntu Server 18.04 LTS - Gen1 ▾ See all images



#### **Step 5: Change the “Size” of the VM to “Standard B1s – 1vcpu, 1 Gi memory (501.80/month)”**

Azure Spot instance ⓘ

Size \* ⓘ Standard\_B1s - 1 vcpu, 1 GiB memory (₹501.80/month) ▾

[See all sizes](#)



#### **Step 6: Select the SSH public key and Create a new key pair**

## Create a virtual machine

Administrator account

Authentication type  SSH public key  Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username \*

SSH public key source

Key pair name \*

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular

**Review + create** **< Previous** **Next : Disks >** || teams.microsoft.com is sharing your screen. **Stop sharing** Hide

## Step 7: Change the OS disk type to Standard HDD

Basics **Disks** Networking Management Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

Disk options

OS disk type \*

Standard HDD is a standard provisioned disk. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.

Encryption type \*

Enable Ultra Disk compatibility

Data disks

**Review + create** **< Previous** **Next : Networking >** || teams.microsoft.com is sharing your screen. **Stop sharing**

## Step 8: Create a new virtual network in the Networking tab

### Step 9: Click Review & Create after networking and check for Validation to pass.

**Validation passed**

Username	azUREUSER
Key pair name	myfirstvm_key1
Public inbound ports	SSH
Azure Spot	No

**Disks**

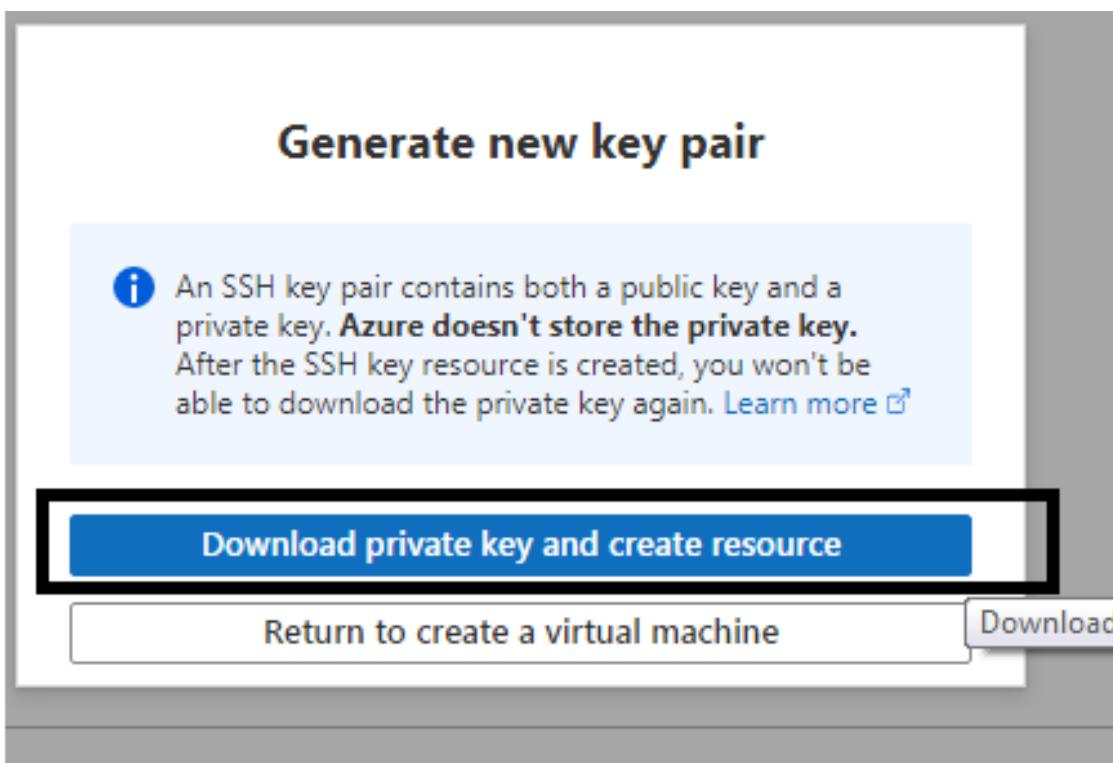
OS disk type	Standard HDD
Use managed disks	Yes
Use ephemeral OS disk	No

**Networking**

Virtual network	(new) newvnet1
Subnet	(new) default (10.0.1.0/24)

**Create**    < Previous    Next >    Download a template for automation

## Step 10: Download private key



## Step 11: Click on go to resource to check the resources.

The screenshot shows the "CreateVm-Canonical.UbuntuServer-18.04-LTS-20201230115021 | Overview" page. On the left, there is a navigation menu with items like "Deployment", "Overview", "Outputs", and "Template". The main area displays deployment details: "Deployment name: CreateVm-Canonical.UbuntuServer-18.04-LTS-2...", "Subscription: Visual Studio Enterprise Subscription - MPN", "Start time: 12/30/2020, 12:20:30 PM", "Correlation ID: 308c563c-c74a-4317-bec8-ed69f", and "Resource group: newgroup2". A prominent green checkmark icon with the text "Your deployment is complete" is centered in the main content area.

## Step 12: Check the resources

myfirstvm

Virtual machine

Search (Ctrl+ /)

Connect Start Restart Stop Capture Delete Refresh Open in mobile

Overview

Activity log

Access control (IAM)

Logs

Diagnose and solve problems

Networking

Connect

Essentials

Resource group (change)  
newgroup2

Status  
Running

Location  
East US 2 (Zone 1)

Subscription (change)  
Visual Studio Enterprise Subscription – MPN

Subscription ID  
2d1leaf32-f707-40db-a62c-2b7d297b42a6

Operating system  
Linux (ubuntu 18.04)

Size  
Standard B1s (1 vcpus, 1 GiB memory)

Public IP address  
20.81.200.87

Virtual network/subnet  
newvnet1/default

DNS name  
Configure

Availability score

### Step 13: Open the following Website:

<https://putty.org/>

You can download PuTTY [here](#).

Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel), but you can download standalone binaries below.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

**putty.exe (the SSH and Telnet client itself)**

32-bit:	<a href="#">putty.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )
64-bit:	<a href="#">putty.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )

**pscp.exe (an SCP client, i.e. command-line secure file copy)**

32-bit:	<a href="#">pSCP.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )
64-bit:	<a href="#">pSCP.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )

**psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)**

32-bit:	<a href="#">psFTP.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )
64-bit:	<a href="#">psFTP.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )

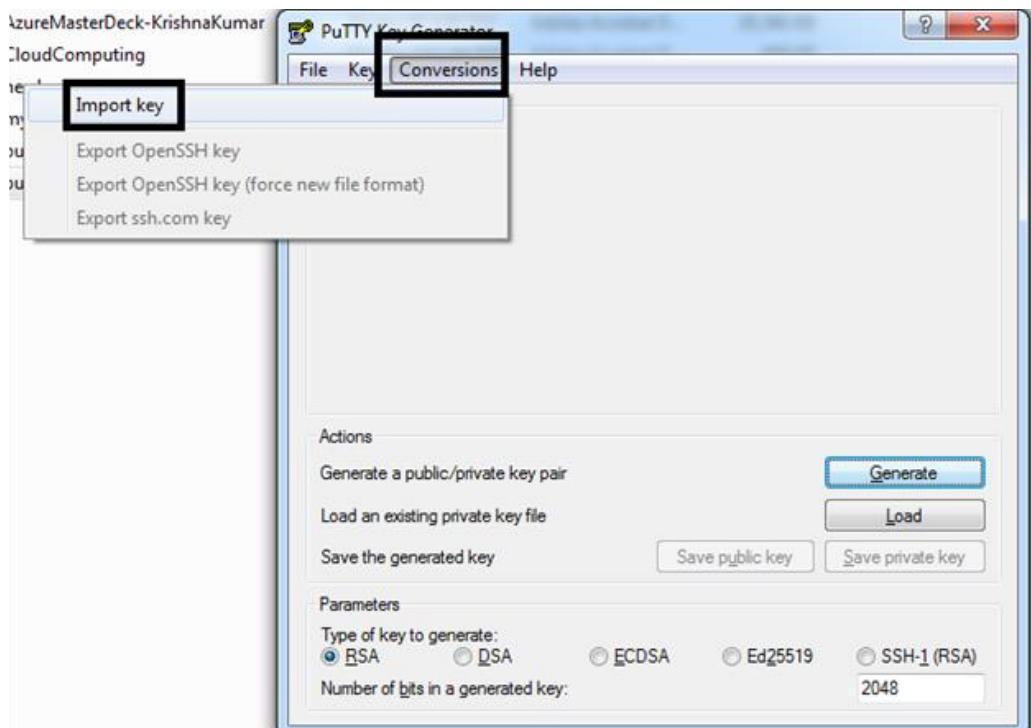
**puttytel.exe (a Telnet-only client)**

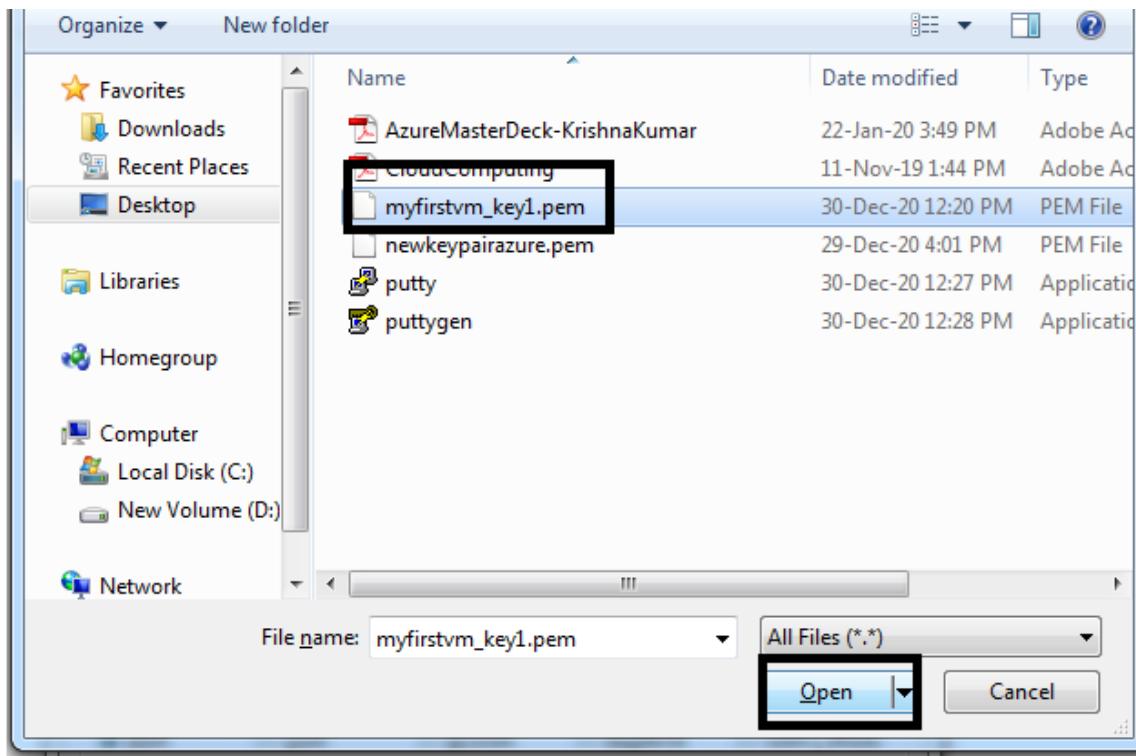
32-bit:	<a href="#">puttytel.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )
64-bit:	<a href="#">puttytel.exe</a>	(or by <a href="#">FTP</a> )	( <a href="#">signature</a> )

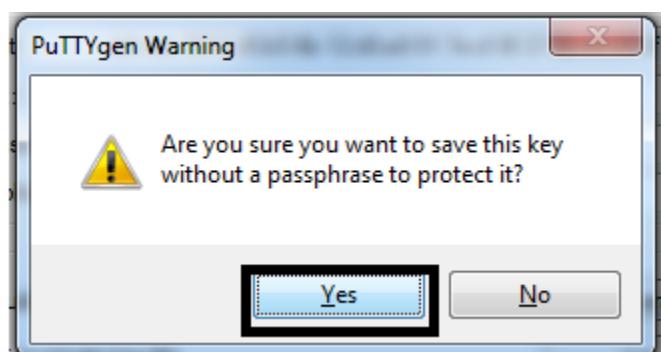
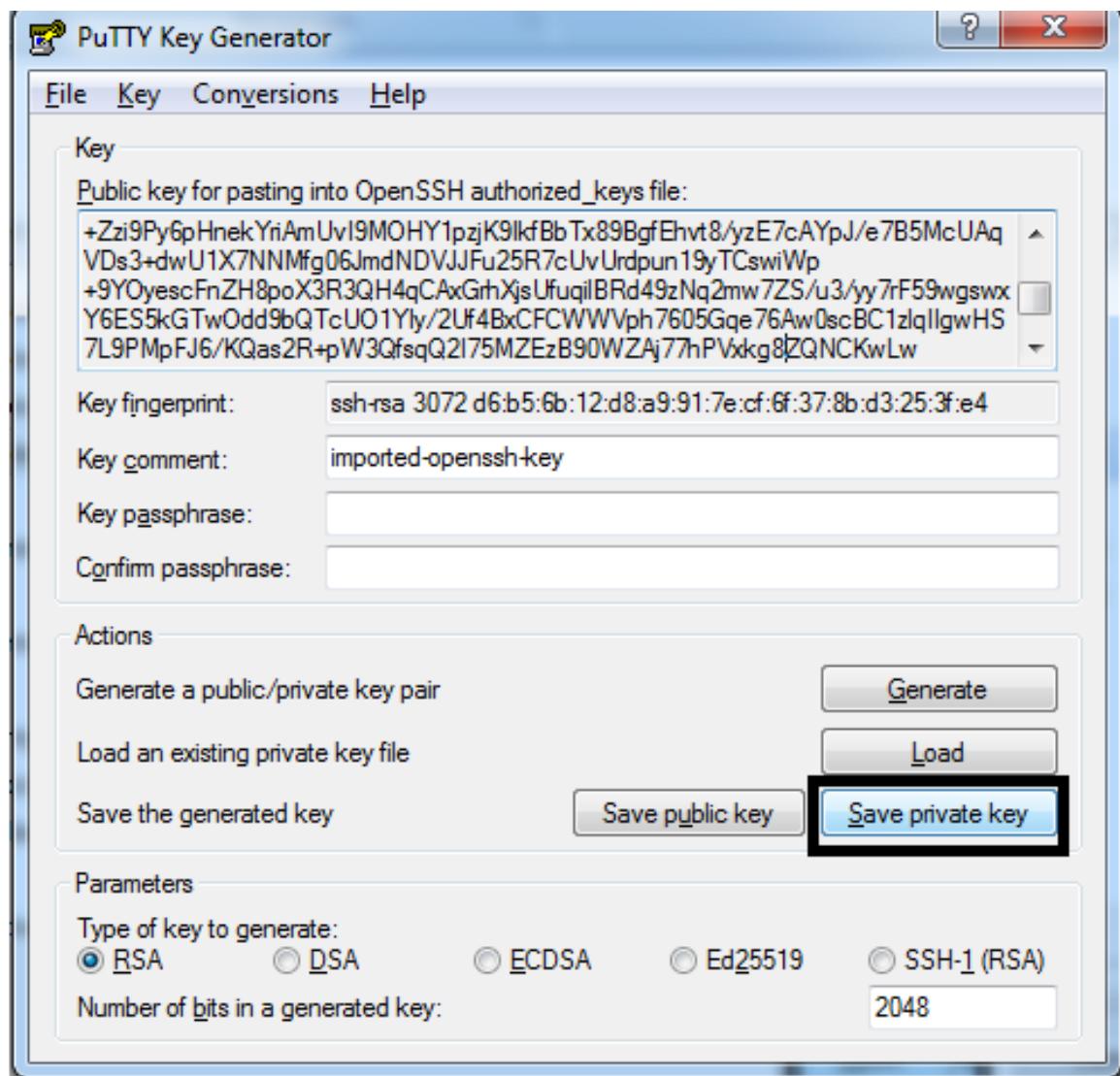
<b>pageant.exe (an SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink)</b>			
32-bit:	<a href="#">pageant.exe</a>	(or by FTP)	(signature)
64-bit:	<a href="#">pageant.exe</a>	(or by FTP)	(signature)
<b>puttygen.exe ( RSA and DSA key generation utility)</b>			
32-bit:	<a href="#">puttygen.exe</a>	(or by FTP)	(signature)
64-bit:	<a href="#">puttygen.exe</a>	(or by FTP)	(signature)
<b>putty.zip (a .ZIP archive of all the above)</b>			
32-bit:	<a href="#">putty.zip</a>	(or by FTP)	(signature)
64-bit:	<a href="#">putty.zip</a>	(or by FTP)	(signature)

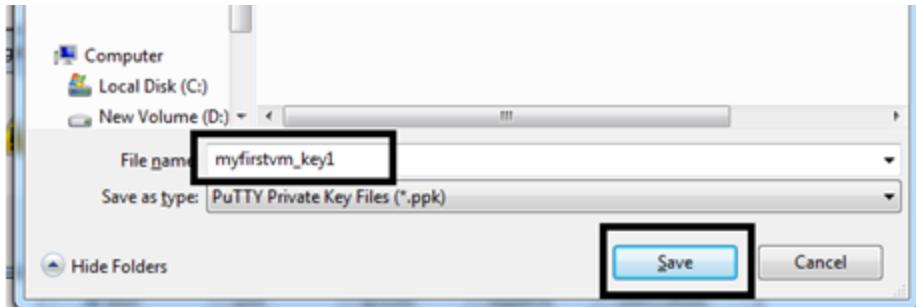
## Step 14: Open puttygen

Convert .pem into .ppk









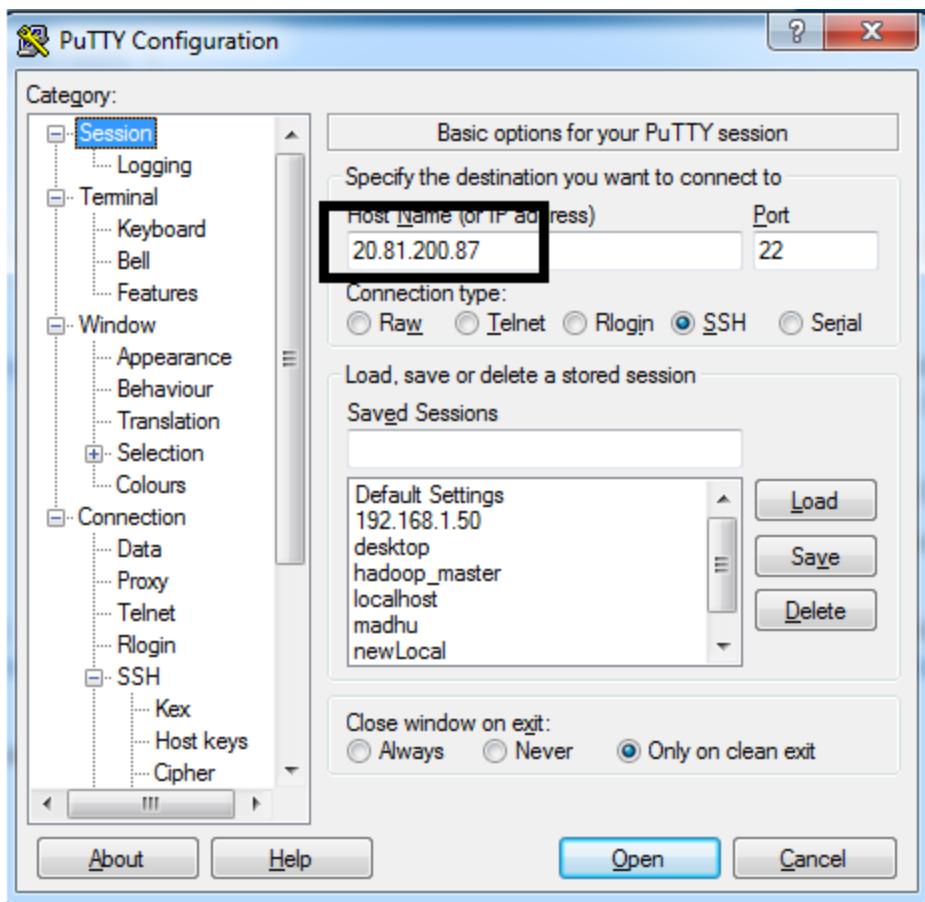
### Step 15: Close the Putty gen

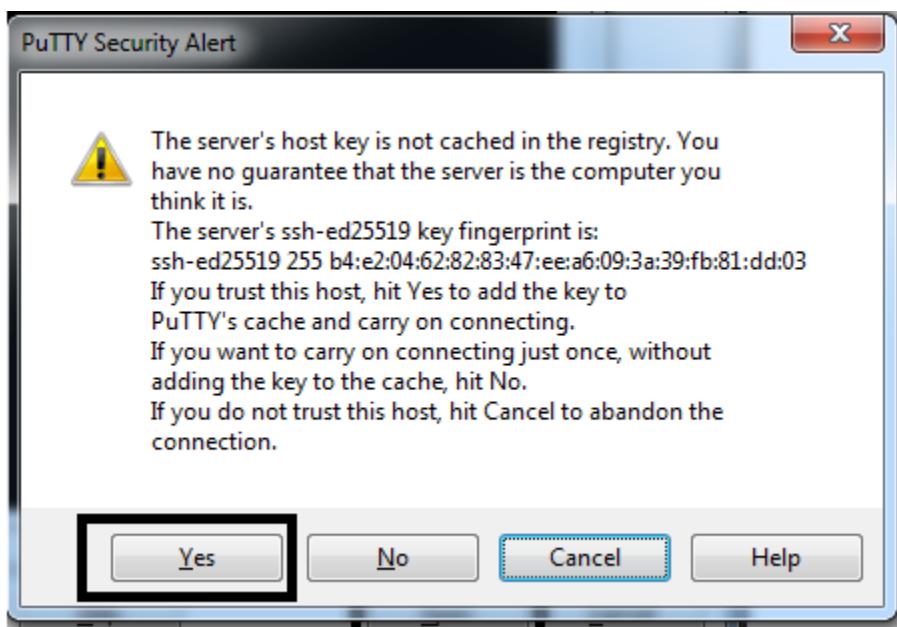
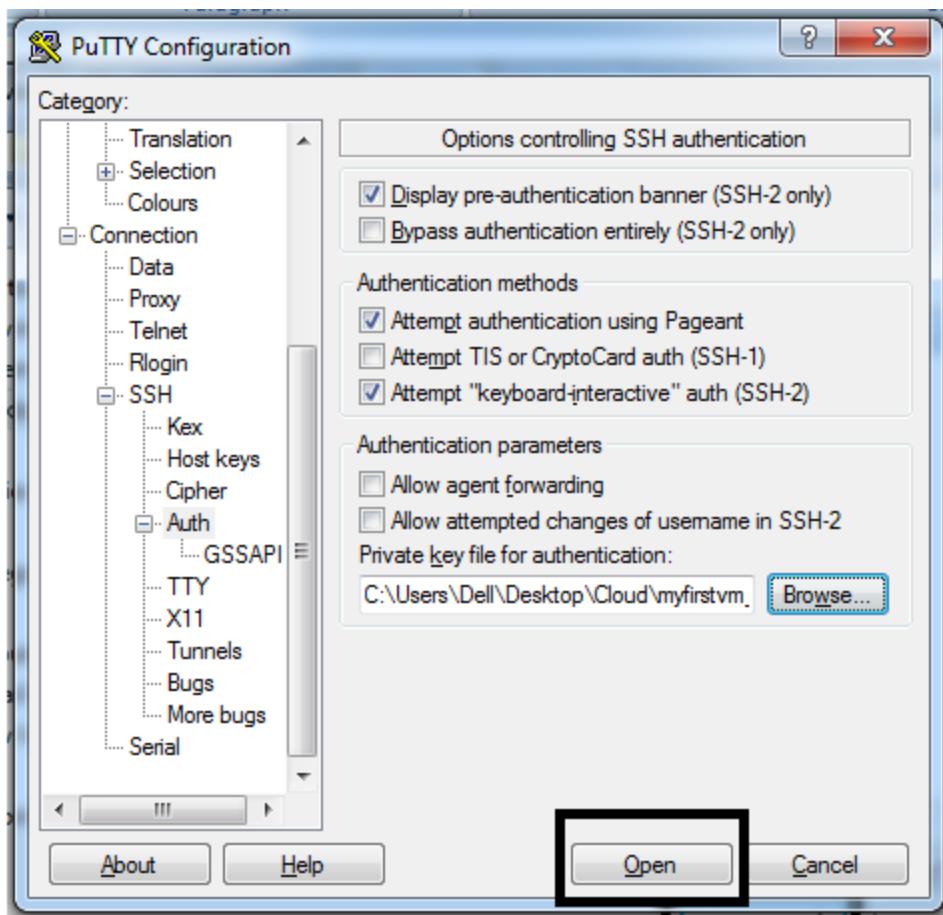
Open putty.exe

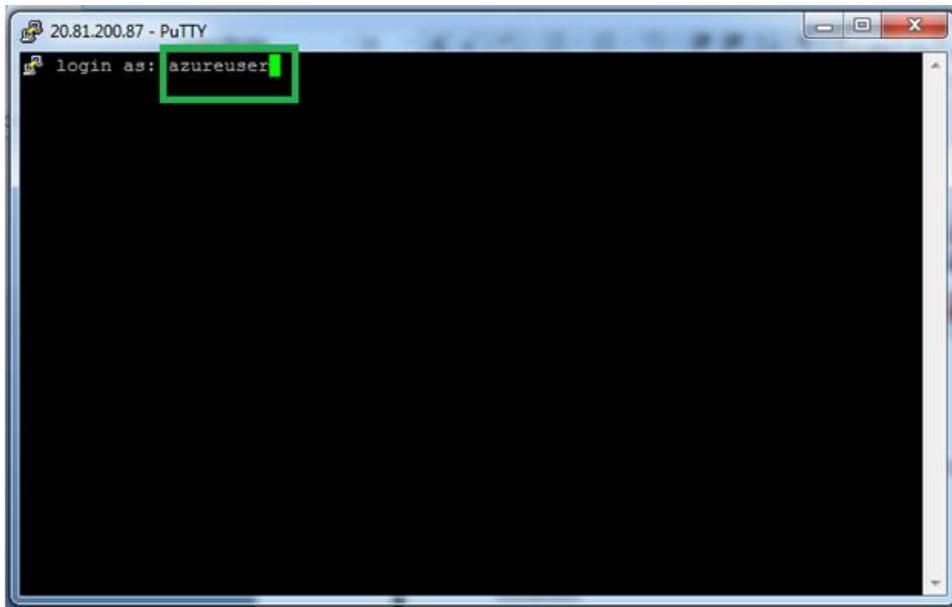
Copy the Public IP

A screenshot of the Azure portal showing a virtual machine named 'newgroup2'. The 'Essentials' section displays the following details:

Setting	Value
Resource group (change)	newgroup2
Status	Running
Location	East US 2 (Zone 1)
Subscription (change)	Visual Studio Enterprise Subscription – MPN
Subscription ID	2d1eaf32-f707-40db-a62c-2b7d297b42a6
Availability zone	1
Operating system	Linux (ubuntu 18.04)
Size	Standard_B1s (1 vCPU, 1 GiB memory)
Public IP address	20.81.200.87
Virtual network/subnet	newvnet1/default
DNS name	Configure







```
azureuser@myfirstvm: ~
System information as of Wed Dec 30 07:07:55 UTC 2020

System load: 0.0          Processes:      109
Usage of /: 4.5% of 28.90GB  Users logged in:  0
Memory usage: 20%          IP address for eth0: 10.0.1.4
Swap usage:  0%

0 packages can be updated.
0 updates are security updates.

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

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

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@myfirstvm:~$ ls
azureuser@myfirstvm:~$
```

Step 16: Exit and Enter

# Creating Virtual Machine in Windows machine

## Creating Windows Server VM:

Home > Virtual machines >

### Create a virtual machine

Resource group \* ⓘ

newgroup2

Virtual machine name \* ⓘ

windowsvm1

Region \* ⓘ

(US) East US 2

Availability options ⓘ

No infrastructure redundancy required

Image \* ⓘ

Windows Server 2012 R2 Datacenter - Gen1

Size \* ⓘ

Standard\_B2s - 2 vcpus, 4 GiB memory (₹2,007.21/month)

See all sizes

Standard\_B2s - 2 vcpus, 4 GiB memory (₹2,007.21/month)

See all sizes

### Select a VM size

Search by VM size... Display cost: Monthly vCPUs: All RAM (GiB): All Add filter

Showing 370 VM sizes. Subscription: Visual Studio Enterprise Subscription – MPN Region: East US 2 Current size: Standard\_B2s Image: Windows Server 2012 R2 Datacenter Learn more about VM sizes Group by series

VM Size ↑	Family ↑	vCPUs ↑	RAM (GiB) ↑	Data disks ↑	Max IOPS ↑	Temp storage (GiB) ↑	Premium support
B2s	General purpose	2	4	4	1280	8	Support
B1s	General purpose	1	1	2	320	4	Support
B2ms	General purpose	2	8	4	1920	16	Support

Select Prices presented are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Final charges will appear in your local currency in cost analysis and billing views. [View Azure pricing calculator.](#)

### Administrator account

Username \* ⓘ

azureuser

Password \* ⓘ

\*\*\*\*\*

Confirm password \* ⓘ

\*\*\*\*\*

## Next Disks:

## Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

### Disk options

OS disk type \* ⓘ

Standard HDD

The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.

Encryption type \*

(Default) Encryption at-rest with a platform-managed key

## Click Review + Create Button

Validation passed

Basics Disks Networking Management Advanced Tags Review + create

### PRODUCT DETAILS

Standard B2s

by Microsoft

[Terms of use](#) | [Privacy policy](#)

Pricing plan and rate apply ⓘ

2.7496 INR/hr

[Pricing for other VM sizes](#)

### TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional

[Create](#)

< Previous

Next >

[Download a template for automation](#)

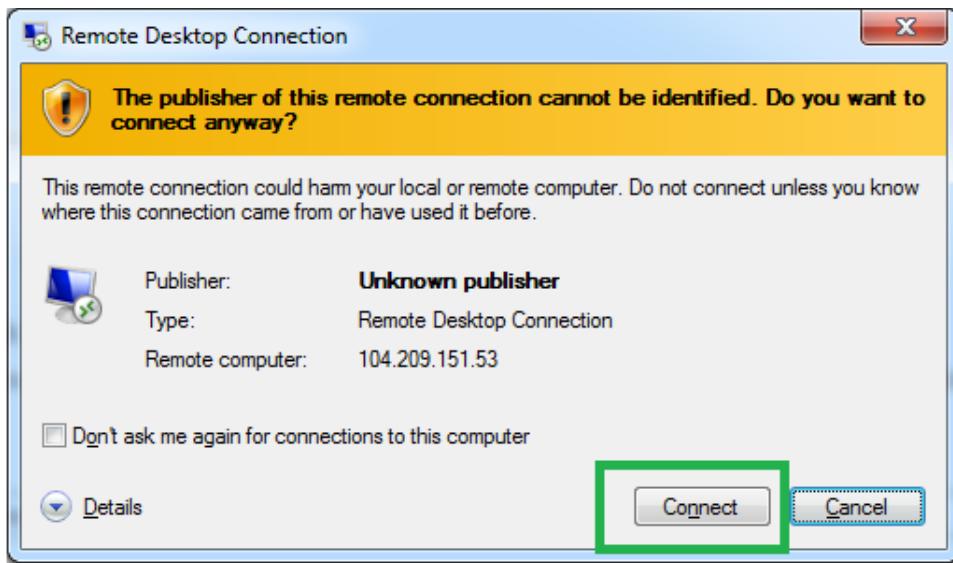
## Click on go to resource

The screenshot shows the Azure portal interface for a completed VM deployment. At the top, it says "CreateVm-MicrosoftWindowsServer.WindowsServer-201-20201230125951 | Overview". Below the header, there's a search bar and standard navigation buttons (Delete, Cancel, Redeploy, Refresh). A feedback message box says "We'd love your feedback! →". The main content area displays a green checkmark icon and the message "Your deployment is complete". It provides deployment details: Deployment name: CreateVm-MicrosoftWindowsServer.WindowsServer-201-20201230125951, Subscription: Visual Studio Enterprise Subscription – MPN, Resource group: newgroup2, Start time: 12/30/2020, 1:01:59 PM, Correlation ID: 155b2362-bb6a-4f2c-9446-94c7. There are sections for "Deployment details (Download)" and "Next steps" with options like "Setup auto-shutdown Recommended", "Monitor VM health, performance and network dependencies Recommended", and "Run a script inside the virtual machine Recommended". At the bottom, there are two buttons: "Go to resource" (highlighted with a green border) and "Create another VM". A status bar at the bottom right says "teams.microsoft.com is sharing your screen."

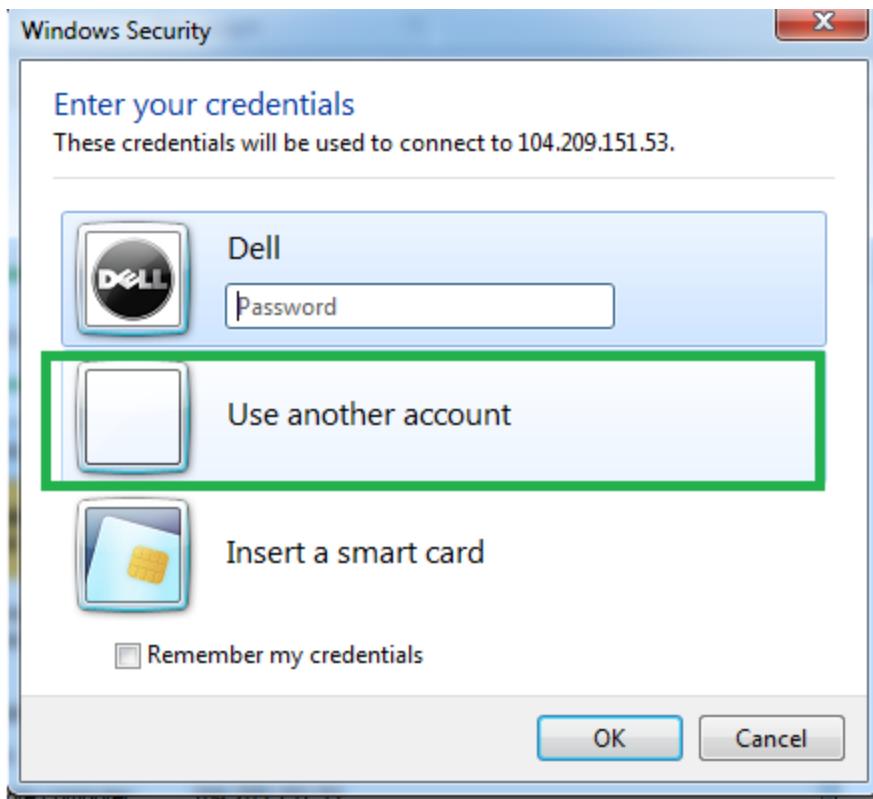
This screenshot shows the "Overview" page for a specific VM. At the top, there's a toolbar with "Connect" (highlighted with a green box), "Start", "Restart", "Stop", "Capture", "Delete", "Refresh", and "Open in mobile". Below the toolbar, the VM's configuration is listed: Operating system Windows, Size Standard B2s (2 vcpus, 4 GiB memory), Public IP address 104.209.151.53, Virtual network/subnet newvnet1/default, and DNS name (with a "Configure" link). On the left, a sidebar shows connection methods: RDP (highlighted with a green box), SSH, and Bastion. The location is listed as East US 2. Subscriptions include Visual Studio Enterprise Subscription – MPN and a specific subscription ID: 2d1leaf32-f707-40db-a62c-2b7d297b42a6.

## Download RDP file

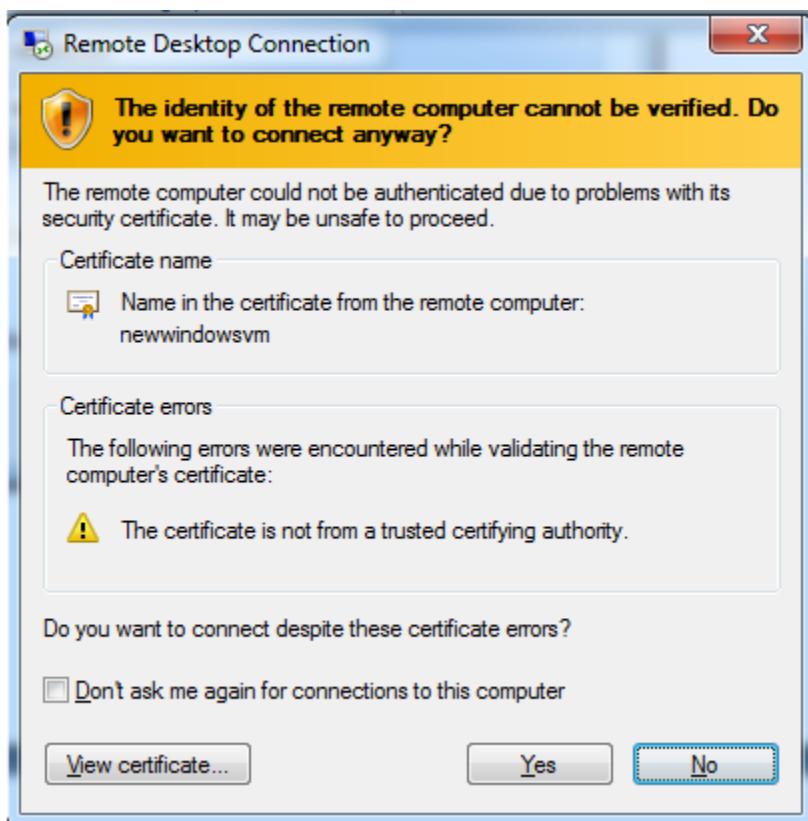
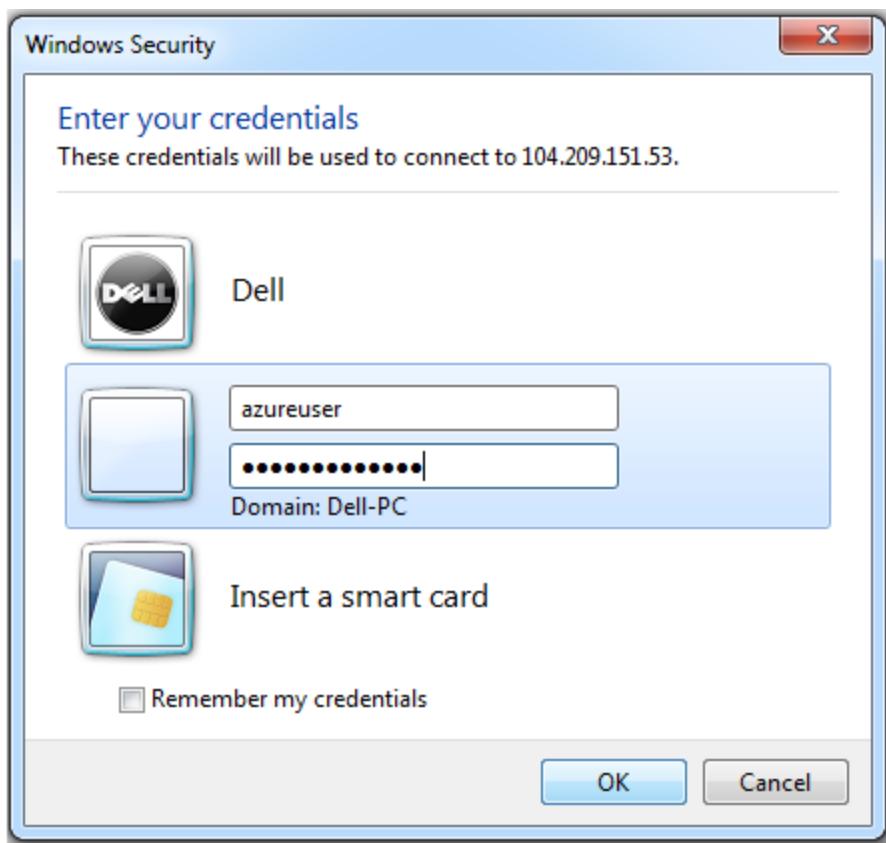
The screenshot shows the Azure portal interface for managing a virtual machine. On the left, there's a sidebar with various navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Networking, Connect, Disks, and Size. Under the Connect section, there's a file named "newwindowsvm.rdp". The main pane is titled "Connect with RDP" and displays fields for "IP address" (Public IP address: 104.209.151.53) and "Port number" (3389). A prominent blue button labeled "Download RDP File" is highlighted with a green border.

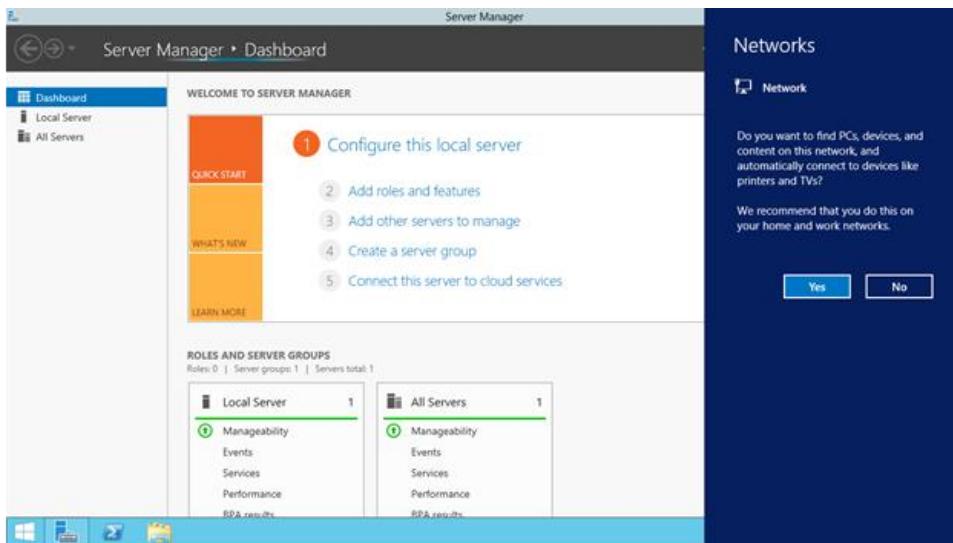


**Click on Use another account**



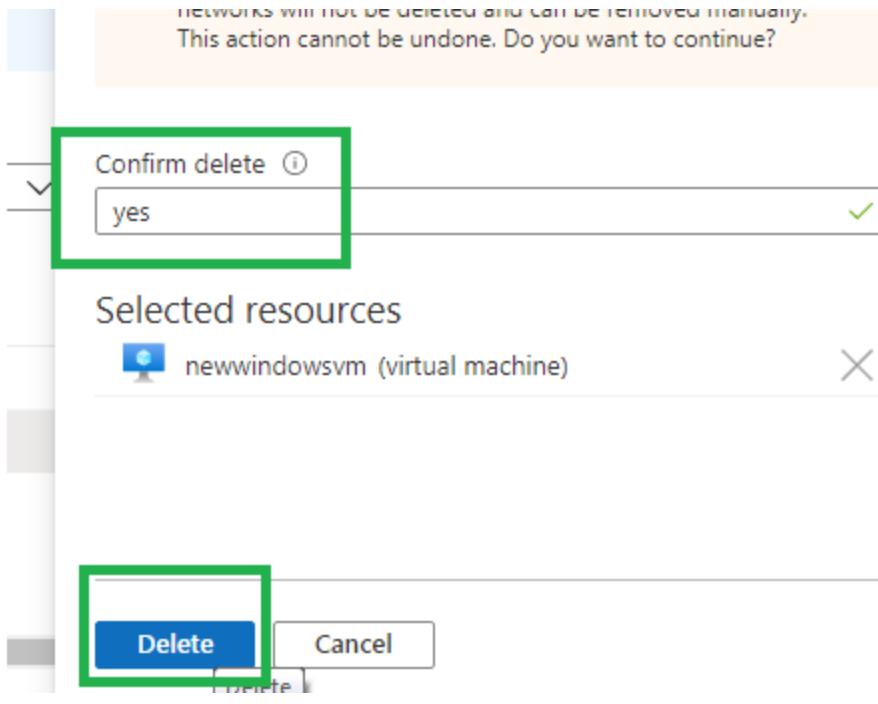
**Enter the Credentials given in the Password tab in the Basics tab.**





The screenshot shows the 'Virtual machines' blade in the Azure portal. At the top, there's a toolbar with 'Add', 'Reservations', 'Edit columns', 'Refresh', 'Try preview', 'Assign tags', 'Start', 'Restart', 'Stop', 'Delete', and 'Services'. A note below the toolbar says 'Try the new virtual machine resource browser! This experience is faster and has improved sorting and filtering capabilities. Please note that the new experience will not show classic virtual machines and does not support for some columns such as maintenance status.' Below the toolbar is a search bar and filter options for 'Subscriptions', 'Filter by name...', 'All subscriptions', 'All resource groups', 'All types', 'All locations', 'All tags', and 'No group'. The main table lists two items:

Name	Type	Status	Resource group	Location	Source	Maintenance status
myfirstvm	Virtual machine	Running	newgroup2	East US 2	Marketplace	-
newwindowsvm	Virtual machine	Running	newgroup2	East US 2	Marketplace	-



## Connecting to an Apache server using Ubuntu 18.04

Step 1: Launch the Ubuntu Virtual Machine and execute the commands

```
sudo apt-get update
```

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

```
sudo service apache2 status      => active (Running)
```

```
curl localhost
```

## **Step 2: Open Azure portal and networking tab and add an inbound rule**

The screenshot shows the Azure portal's Networking settings for a resource group. The "Inbound port rules" tab is selected, displaying a table of rules:

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	
65500	DenyAllInbound	Any	Any	Any	Any	

An "Add inbound port rule" button is located in the top right corner of the table area, also highlighted with a green box.

 Add inbound security rule

webappvm1-nsg

 Basic

Source \* ⓘ  
Any

Source port ranges \* ⓘ  
\*

Destination \* ⓘ  
Any

Destination port ranges \* ⓘ  
80

Protocol \*  
 Any  TCP  UDP  ICMP

**Add**

### Step 3: Open the browser and type the public address



The screenshot shows a web browser window with the following details:

- Address bar: Not secure | 52.251.39.205
- Toolbar icons: Apps, WhatsApp, WebMail, State Bank of India, Tiruvachagam.
- Page Content:
  - Apache2 Ubuntu Default Page**
  - It works!**
  - This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.
  - If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
  - Configuration Overview**
  - Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.
  - The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

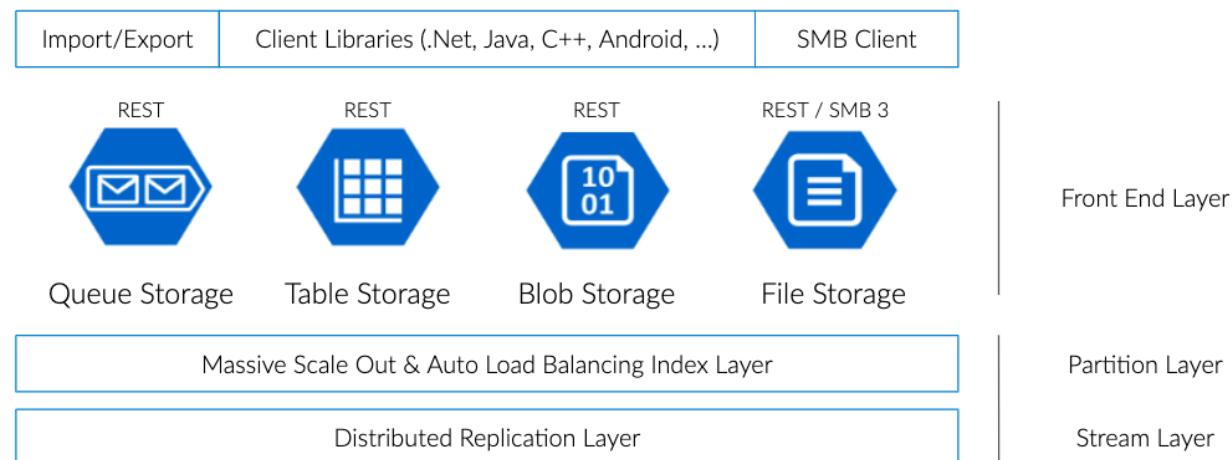
### Step 4: Exit the browser

## Storage Account

An Azure storage account contains all of your Azure Storage data objects: blobs, files, queues, tables, and disks. The storage account provides a unique namespace for your Azure Storage data that is accessible from anywhere in the world over HTTP or HTTPS. Data in your Azure storage account is durable and highly available, secure, and massively scalable.

### Types of Storage Accounts

## Azure Storage Architecture



### Storage types and their properties

Storage account type	Supported services	Redundancy options	Deployment model <sup>1</sup>
General-purpose V2	Blob, File, Queue, Table, Disk, and Data Lake Gen2 <sup>2</sup>	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS <sup>3</sup>	Resource Manager
General-purpose V1	Blob, File, Queue, Table, and Disk	LRS, GRS, RA-GRS	Resource Manager, Classic
Block Blob Storage	Blob (block blobs and append blobs only)	LRS, ZRS <sup>3</sup>	Resource Manager
File Storage	File only	LRS, ZRS <sup>3</sup>	Resource Manager
Blob Storage	Blob	LRS, GRS, RA-GRS	Resource Manager

## Creating a storage account in Azure

### Step 1: Search for storage accounts

The screenshot shows the Microsoft Azure portal interface. At the top, there is a search bar with the text "storage" typed into it. Below the search bar, the "Services" section is visible, listing several options: Storage accounts, Storage explorer, Storage accounts (classic), Storage Sync Services, Data Lake Storage Gen1, HPC caches, Disks (classic), and OS images (classic). A green box highlights the search bar and the "Storage accounts" option.

### Add a new Storage Account

The screenshot shows the "Storage accounts" list page. At the top, there is a header with the title "Storage accounts" and a "ValueMomentum Inc." logo. Below the header, there is a toolbar with buttons for "Add" (highlighted with a green box), "Manage view", "Refresh", "Export to CSV", and "Open query". There are also filters for "Subscription == all" and "Resource group == all". A search bar with the placeholder "Filter by name..." is located below the toolbar. The main area displays the message "Showing 0 to 0 of 0 records." Below this, there are two columns: "Name" and "Type", each with a "↑↓" arrow indicating sorting.

### Enter the details in the Basics tab

## Create storage account

Subscription \* Visual Studio Enterprise Subscription – MPN

Resource group \* newgroup2 Create new

Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name \* saistorageaccount123

Location \* (US) East US 2

Performance Standard Premium

Account kind StorageV2 (general purpose v2)

Replication Read-access geo-redundant storage (RA-GRS)

**Review + create**

< Previous

|| teams.microsoft.com is sharing your screen.

**Stop sharing**

Hide

**Validation passed**

Basics

Networking

Data protection

Advanced

Tags

**Review + create**

### Basics

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	newgroup2
Location	East US 2
Storage account name	saistorageaccount123
Deployment model	Resource manager
Account kind	StorageV2 (general purpose v2)
Replication	Read-access geo-redundant storage (RA-GRS)
Performance	Standard

**Create**

< Previous

Next

|| teams.microsoft.com is sharing your

 Delete  Cancel  Redeploy  Refresh

 We'd love your feedback! →

## Your deployment is complete



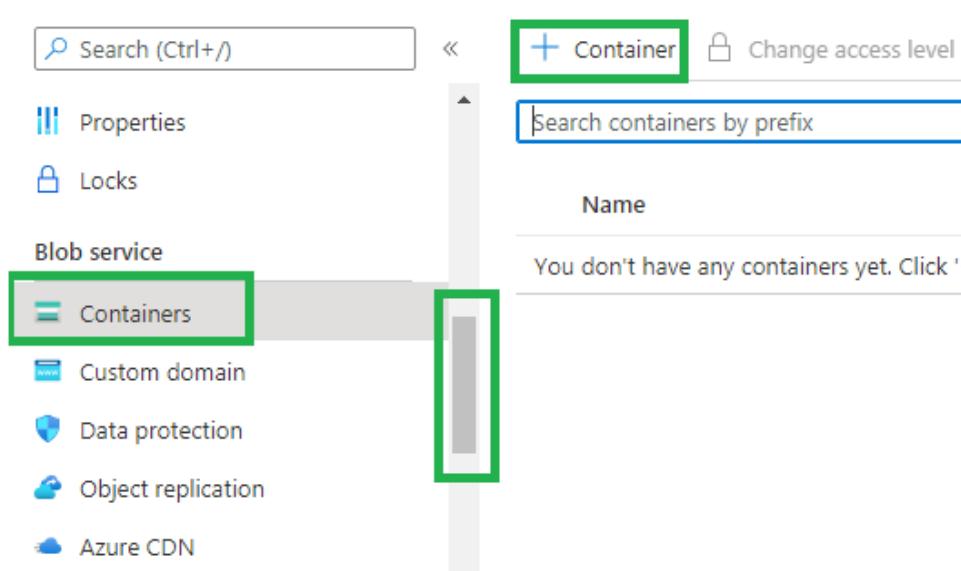
Deployment name: Microsoft.StorageAccount-20201230132242  
Subscription: Visual Studio Enterprise Subscription – MPN  
Resource group: newgroup2

▼ Deployment details [\(Download\)](#)

^ Next steps

[Go to resource](#)

**Click the Container in the left under Blob service**



The screenshot shows the Azure Storage Account blade for a specific storage account. On the left, there's a sidebar with the following options:

- Properties
- Locks
- Blob service** (selected)
- Containers (highlighted with a green box)
- Custom domain
- Data protection
- Object replication
- Azure CDN

The main content area has a search bar at the top. Below it, there's a "Container" button with a plus sign and a "Change access level" button. A "Search containers by prefix" input field is also present. A message below the search field says "Name" and "You don't have any containers yet. Click +".

Name \*

 ✓Public access level (i)

⚠ Blobs within the container can be read by anonymous request, but container data is not available. Anonymous clients cannot enumerate the blobs within the container.

▽ AdvancedCreate

Discard

+ Container 🔒 Change access level ↳ Restore cor

Search containers by prefix

Name
<input type="checkbox"/> mycontainer

### After creating the container upload a file into the container

Upload 🔒 Change access level ⟳ Refresh | ⓧ Delete | ⤓ Change tier | ⌚ Acquire lease | 🕒

Authentication method: Access key ([Switch to Azure AD User Account](#))  
Location: mycontainer

Search blobs by prefix (case-sensitive)

Files (i)  
 Upload X

Overwrite if files already exist

▽ Advanced

The screenshot shows the Azure Storage Blob upload and download interface. At the top, there's a file selection box containing "BrouchureContent.txt". Below it is a checkbox for "Overwrite if files already exist". A green box highlights the "Upload" button. In the bottom right corner of the upload area, there's a smaller "Upload" button. The main content area shows a table of blobs with one entry: "BrouchureContent.txt". The "Download" link in the table's context menu is highlighted with a green box. To the right of the table is a vertical context menu with options like "View/edit", "Download", "Properties", "Generate SAS", etc., with the "..." option also highlighted with a green box.

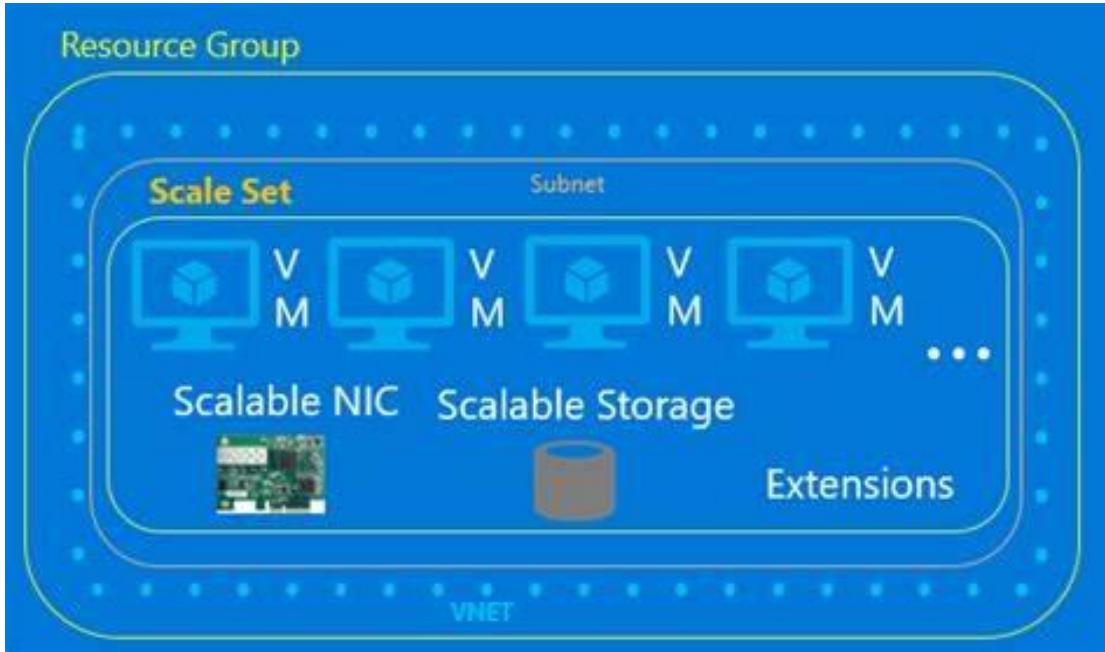
## Scale Sets

Azure virtual machine scale sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update a large number of VMs. With virtual machine scale sets, you can build large-scale services for areas such as compute, big data, and container workloads.

### Why we use Scale Sets in Azure?

- Easy to create and manage multiple VMs
- Provides high availability and application resiliency

- Allows your application to automatically scale as resource demand changes
  - Works at large-scale



## Creating a Scale Set in Azure

## **Step 1: Open the azure portal and search Scale sets**

The screenshot shows the Microsoft Azure search interface. The search bar at the top contains the query "virtual machine scale". Below the search bar, the results are displayed under the heading "Services". The first result, "Virtual machine scale sets", is highlighted with a yellow box. Other service options listed include "Virtual machines", "Virtual networks", "SQL virtual machines", "CloudSimple Virtual Machines", "Machine Learning", "Virtual clusters", "Virtual WANs", and "Virtual network gateways". To the right of the search results, there are sections for "Marketplace", "Documentation", and "Resource Groups". The "Documentation" section lists several links related to virtual machine scale sets, such as "Azure virtual machine scale sets overview - Azure Virtual ...", "Virtual Machine Scale Sets documentation - Azure Virtual ...", and "Quickstart - Create a virtual machine scale set in the ...". The "Resource Groups" section indicates "No results were found."

## Step 2: Add a new Scale set

Home >

## Virtual machine scale sets

ValueMomentum Inc.

+ Add Edit columns Refresh | Assign tags Start Restart Stop Delete

Subscriptions: All 2 selected – Don't see a subscription? Open Directory + Subscription settings

Filter by name... All subscriptions All resource groups All locations

0 items

Name ↑↓	Status	Instances	Azure Spot eviction policy
---------	--------	-----------	----------------------------

### Step 3: Enter the details in the Basics tab

#### Create a virtual machine scale set

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource group \*

Visual Studio Enterprise Subscription – MPN

newgroup2

Create new

Scale set details

Virtual machine scale set name \*

scalingset1

Region \*

(US) East US 2

Availability zone ⓘ

None

Instance details

Image \* ⓘ

Ubuntu Server 18.04 LTS - Gen1

See all images

Azure Spot instance ⓘ

Size \* ⓘ

Standard\_B1s - 1 vcpu, 1 GiB memory (€501.80/month)

See all sizes

#### Create a new key pair or Use an existing key pair in Azure

Administrator account

Authentication type  Password  SSH public key

Username \*

SSH public key source

Stored Keys

---

[Review + create](#) [< Previous](#) [Next : Disks >](#)

## Step 4: Select the Standard HDD in the Disks tab

Basics Disks Networking Scaling Management Health Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

Disk options

OS disk type \*  The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.

Encryption type \*

Enable Ultra Disk compatibility  Ultra Disk compatibility is not available for this VMSS size and location.

---

[Review + create](#) [< Previous](#) [Next : Networking >](#)

## Step 5: Scroll down and select “load balancer” and enter the details

## Create a virtual machine scale set

### Load balancing

You can place this virtual machine scale set in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Use a load balancer



### Load balancing settings

- **Application Gateway** is an HTTP/HTTPS web traffic load balancer with URL-based routing, SSL termination, session persistence, and web application firewall. [Learn more about Application Gateway](#)
- **Azure Load Balancer** supports all TCP/UDP network traffic, port-forwarding, and outbound flows. [Learn more about Azure Load Balancer](#)

Load balancing options \* ⓘ



Select a load balancer \* ⓘ



**Create new**

## Step 6: Create a load balancer

### Create a load balancer

X

Azure Load Balancer enables you to scale your applications and create high availability for your services. Load Balancer supports inbound and outbound scenarios, provides low latency and high throughput, and scales up to millions of flows for all TCP and UDP applications. [Learn more about Azure Load Balancer](#)

Your load balancer will be placed in the same subscription, resource group, and region as your virtual machine scale set. Azure will configure basic settings for the frontend IP, backend address pools, NAT rules, and NAT pools for this load balancer automatically.

Name \* ⓘ

scalingset1-lb

Public IP address name \* ⓘ

scalingset1-ip

Domain name label \* ⓘ

scalesetsai12

.eastus2.cloudapp.azure.com

SKU

Public

Type

Standard

Availability zone \* ⓘ

Zone-redundant

**Create**

**Discard**

## Step 7: Select Scaling and change the properties and Select Review & create

- **Initial instance count =2**

- **Scaling policy = Custom**
- **Number of instances increased by = 1**
- **CPU threshold = 25**

### **Step 8: After the completion of deployment go to instances in the left**

The screenshot shows the 'Instances' blade for a scale set named 'scalingset1'. The left sidebar has 'Instances' selected. The main area shows a table with columns: Name, Computer name, Status, Health state, Provisioning state, Protection policy, and Latest model. Two rows are present: 'scalingset1\_1' (status: Updating (Running), provisioning state: Updating) and 'scalingset1\_2' (status: Running, provisioning state: Succeeded).

### **Step 9: Select networking tab and check for the load balancer**

The screenshot shows the 'Networking' blade for the same scale set. The left sidebar has 'Networking' selected. The main area shows a network interface named 'newgroup2-vnet-nic01'. Under 'IP configuration', it shows 'newgroup2-vnet-nic01-defaultipConf...'. The 'Network Interface' section shows 'Virtual network/subnet: newgroup2-vnet/default' and 'Topology: Accelerated networking: Disabled'. Below this, there are tabs for 'Inbound port rules', 'Outbound port rules', and 'Load balancing'. The 'Load balancing' tab is selected, showing a table with columns: Type, Frontend IP address, Frontend DNS address, and Backend pool. One entry is listed: 'set1-lb' (Type: Load balancer, Frontend IP address: 52.247.82.85, Frontend DNS address: scalesetsai12.eastus2.cloudapp.azure.com, Backend pool: bepool).

Delete the Scale set if it is a free trial version.

---

## **Microsoft Azure Virtual Machine Deep Dive**

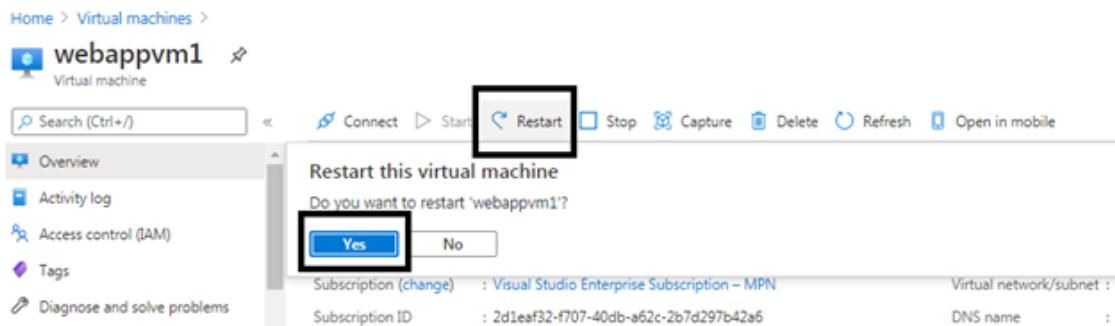
Azure virtual machine deep dive are the in-depth services provided and performed in the virtual machine.

Generally, the VM Deep dive consists of:

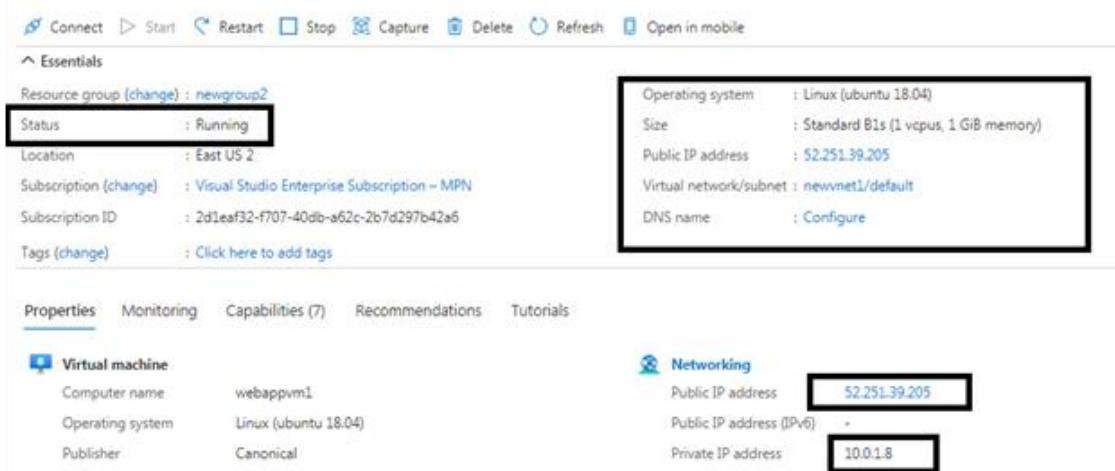
- Restarting a VM
- Starting and stopping a VM
- Creating a disk in the VM
- Resizing a VM

## 1) Restarting a VM

Open the VM resource and select restart at the top

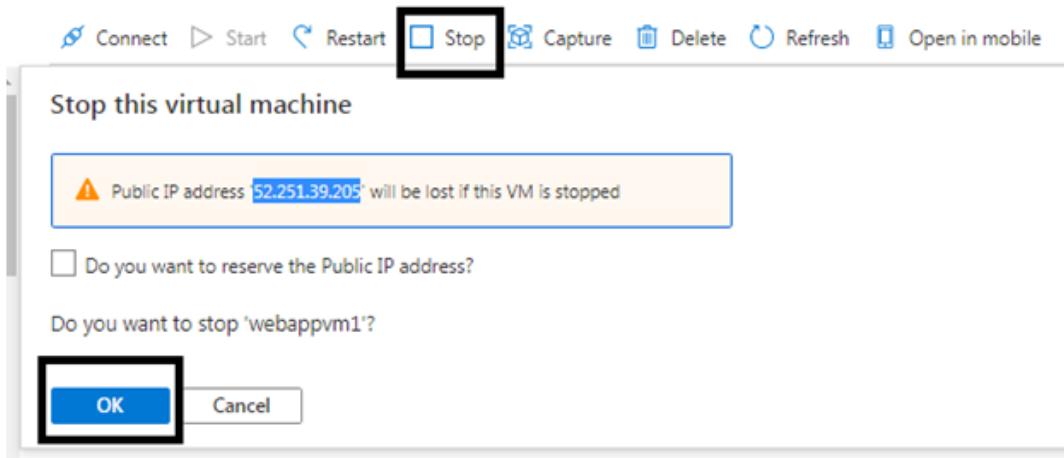


Note that the public IP address of the VM has changed



## 2) Stopping a VM

Select stop at the top in the resource



Connect ▶ Start ⏪ Restart ☰ Stop Capture Delete Refresh Open in mobile

Essentials

Resource group (change)	: newgroup2	Operating system	: Linux
Status	: Stopped (deallocated)	Size	: Standard B1s (1 vcpus, 1 GiB memory)
Location	: East US 2	Public IP address	: webappvm1-ip
Subscription (change)	: Visual Studio Enterprise Subscription – MPN	Virtual network/subnet	: newvnet1/default
Subscription ID	: 2d1leaf32-f707-40db-a62c-2b7d297b42a6	DNS name	: Configure
Tags (change)	: Click here to add tags		

After the VM has stopped then select start

Connect ▶ Start ⏪ Restart ☰ Stop Capture Delete Refresh Open in mobile

Essentials

Resource group (change)	: newgroup2	Operating system	: Linux
Status	: Stopped (deallocated)	Size	: Standard B1s (1 vcpus, 1 GiB memory)
Location	: East US 2	Public IP address	: webappvm1-ip
Subscription (change)	: Visual Studio Enterprise Subscription – MPN	Virtual network/subnet	: newvnet1/default
Subscription ID	: 2d1leaf32-f707-40db-a62c-2b7d297b42a6	DNS name	: Configure
Tags (change)	: Click here to add tags		

Note that the Public IP address has changed after starting the VM

Connect ▶ Start ⏪ Restart ☰ Stop Capture Delete Refresh Open in mobile

Essentials

Resource group (change)	: newgroup2	Operating system	: Linux
Status	: Starting	Size	: Standard B1s (1 vcpus, 1 GiB memory)
Location	: East US 2	Public IP address	: 52.254.53.96
Subscription (change)	: Visual Studio Enterprise Subscription – MPN	Virtual network/subnet	: newvnet1/default
Subscription ID	: 2d1leaf32-f707-40db-a62c-2b7d297b42a6	DNS name	: Configure
Tags (change)	: Click here to add tags		

### 3) Creating a disk in the VM

In the resource select the disk in the Settings tab

The screenshot shows the Azure portal's 'Disks' section under the 'Settings' tab for a virtual machine. On the left, there's a sidebar with various navigation items. The 'Disks' item is selected and highlighted with a black box. In the main content area, there's a table for the OS disk and a section for data disks. A 'Create and attach a new disk' button is visible. On the right, a 'Host caching' dropdown menu is open, showing three options: 'None', 'Read-only', and 'Read/write'. 'Read/write' is currently selected and highlighted with a blue box.

Select create and attach a new disk

Create a new disk with following

- **Disk name**
- **Storage type = Standard HDD**
- **Size = 32**
- **Host caching = Read only**

The screenshot shows the 'Data disks' section in the 'Settings' tab for a virtual machine. It displays a table for existing data disks and a form for creating a new one. The 'Disk name' field contains 'webappvm1Disc2', the 'Storage type' dropdown is set to 'Standard HDD', and the 'Size (GB)' input field has '32' entered. The 'Host caching' dropdown is set to 'Read/write'. The 'Encryption' dropdown is set to 'Platform-managed key'.

2

Save Discard Refresh | Additional settings

OS disk

Swap OS disk

Disk name	Storage type	Size (GiB)	Max IOPS	Max throughput (MB/s)	Encryption	Host caching
webappvm1_OsDisk_1_3cb06f	Standard HDD	30	500	60	SSE with PMK	Read/write

Data disks

Filter by name

Showing 1 of 1 attached data disks

+ Create and attach a new disk Attach existing disks

Storage type	Size (GiB)	Max IOPS	Max throughput (MB/s)	Encryption	Host caching
Standard HDD	32	500	60	Platform-managed key	Read-only

You can see that the disk has been successfully created

Save Discard Refresh | Additional settings

OS disk

Swap OS disk

Disk name	Storage type	Size (GiB)	Max IOPS	Max throughput (MB/s)	Encryption	Host caching
webappvm1_OsDisk_1_3cb06f	Standard HDD	30	500	60	SSE with PMK	Read/write

Data disks

Filter by name

Showing 1 of 1 attached data disks

+ Create and attach a new disk Attach existing disks

ID	Disk name	Storage type	Size (GiB)	Max IOPS	Max throughput (MB/s)	Encryption	Host caching
0	webappvm1Disc2	Standard HDD	32	500	60	SSE with PMK	Read-only

#### 4) Resizing a VM

Select size in the Settings tab in the resource

Change the size from B1s to B2s

**webappvm1 | Size**

Showing 371 VM sizes. Subscription: Visual Studio Enterprise Subscription = MPN Region: East US 2 Current size: Standard\_B1s Learn more about VM sizes Group by series

VM Size ↑	Family ↑	vCPUs ↑	RAM (GB) ↑	Data disks ↑	Max IOPS ↑	Temp storage (GB) ↑
D51_v2	General purpose	1	3.5	4	3200	7
D2v2	General purpose	2	8	4	3200	16
<b>B2s</b>	General purpose	<b>2</b>	<b>4</b>	<b>4</b>	<b>1280</b>	<b>8</b>
B1s	General purpose	1	1	2	320	4
B2ms	General purpose	2	8	4	1920	16
B1ms	General purpose	1	2	2	640	4
B1ls	General purpose	1	0.5	2	160	4

Prices present in any applicable region. teams.microsoft.com is sharing your screen. Stop sharing Hide

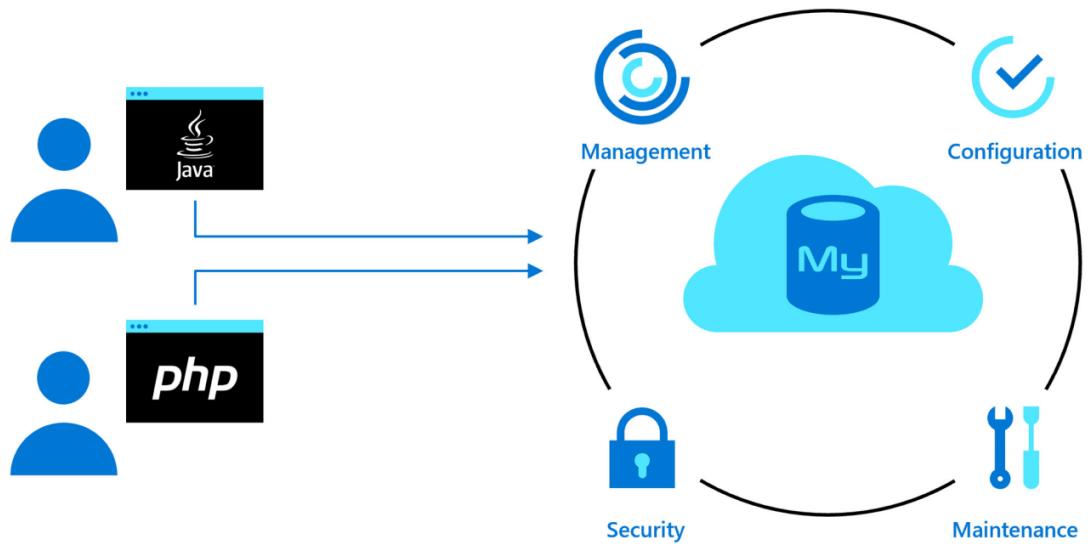
**Microsoft Azure** Home > webappvm1 | Size

Successfully resized virtual machine 'webappvm1' to 'Standard\_B2s'.

## MYSQL Database in Azure

Azure Database for MySQL is easy to set up, operate, and scale. Enjoy advanced security and high availability with up to 99.99 percent SLA and a choice of single zone or zone redundant high availability.

It automates the management and maintenance of your infrastructure and database server, including routine updates, backups, and security. Enjoy maximum control of database management with custom maintenance windows and multiple configuration parameters for fine grained tuning with Flexible Server



## Creating SQL Database in Azure

### Step 1: Open portal and search SQL database in azure

The screenshot shows the Microsoft Azure portal interface. The search bar at the top contains the text "database". The search results are displayed under the "Services" section, with "Azure Database for MySQL servers" highlighted. Other listed services include "Azure Database for MariaDB servers", "Azure Database for MySQL flexible servers", "Azure Database for PostgreSQL flexible servers", and "Azure Database for PostgreSQL servers". Below the search bar, there is a "Create a resource" button and a "Recent resources" sidebar with items like "webappservice" and "newgroup2". On the right side, there are sections for "Marketplace", "Documentation", and "Resource Groups".

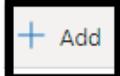
### Step 2: Add a new database

Microsoft Azure  Search resources, services, and docs (G+/-)

Home >

## SQL databases

ValueMomentum Inc.



Reservations



Edit columns



Refresh



Assign tags



Delete

New SQL database



Try our new Azure SQL resource browser! This experience offers a unified view of all your SQL Server resources

**Subscriptions:** All 2 selected – Don't see a subscription? Open Directory + Subscription settings

Filter by name...

All subscriptions

All resource groups

0 items

### Step 3: Enter the details such as resource group, database name

#### Create SQL Database

Microsoft

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Visual Studio Enterprise Subscription – MPN

Resource group \*

newgroup2

[Create new](#)

#### Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \*

newdb1

Server \*

Select a server

[Create new](#)

### Step 4: Create a new server for the database

## New server

X

Microsoft

Server name \*

 ✓

✓ Server name should not contain reserved words.  
✓ The specified server name is available.

Password \*

Confirm password \*

Location \*

 ▼

### Step 5: Create a username and a password for the database

Server name \*

newdb123 ✓ .database.windows.net

Server admin login \*

azureuser ✓

Password \*

\*\*\*\*\* ✓

Confirm password \*

\*\*\*\*\* ✓

Location \*

(US) East US

OK

## Step 6: Configure the Computer + Storage to Basic with 2GB

Database name \*

newdb1 ✓

Server \* ⓘ

(new) newdb123 (East US) ✓ Create new

Want to use SQL elastic pool? \* ⓘ  Yes  No

Compute + storage \* ⓘ

**General Purpose**  
Gen5, 2 vCores, 32 GB storage, zone redundant disabled  
[Configure database](#)

## Configure

Feedback

Looking for basic, standard, premium?

**General Purpose**  
Scalable compute and storage options  
500 - 20,000 IOPS  
2-10 ms latency

**Hyperscale**  
On-demand scalable storage  
500 - 204,800 IOPS  
1-10 ms latency

**Compute tier**

**Provisioned**   
Compute resources are pre-allocated  
Billed per hour based on vCores configured

**Serverless**   
Compute resources are auto-scaled  
Billed per second based on vCores used

## Configure

Feedback

**Basic**  
For less demanding workloads

**Standard**  
For workloads with typical performance requirements

**Premium**  
For IO-intensive workloads.

DTUs [What is a DTU?](#)

**5 (Basic)**

Data max size

100 MB  2 GB

**Apply**

## Step 6: In the networking select Connectivity method to No access

Basics **Networking** Additional settings Tags Review + create

Configure network access and connectivity for your server. The configuration selected below will apply to the selected server 'newdb123' and all databases it manages. [Learn more](#)

**Network connectivity**

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can config [Connectivity method](#) after server creation. [Learn more](#)

Connectivity method \*  **No access**  Public endpoint  Private endpoint

**Step 7: In the Additional settings select Use existing data to "Sample"**

## Create SQL Database

Microsoft

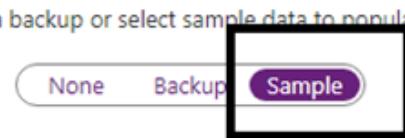
Basics Networking Additional settings Tags Review + create

Customize additional configuration parameters including collation & sample data.

### Data source

Start with a blank database, restore from a backup or select sample data to populate your new database.

Use existing data \*



AdventureWorksLT will be created as the sample database.

**Step 8: Select Review & Create**

## Create SQL Database

Microsoft

Basics Networking Additional settings Tags Review + create

### Product details

SQL database  
by Microsoft  
[Terms of use](#) | [Privacy policy](#)

#### Estimated cost per month

329.89 INR

[View pricing details](#)

### Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offer associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft provides support, billing and other transactional activities. Microsoft does not provide rights for third-party

[Create](#)

< Previous

[Download a template for automation](#)

**Step 8: After the deployment is successful go to resource**

Delete Cancel Redeploy Refresh

We'd love your feedback! →

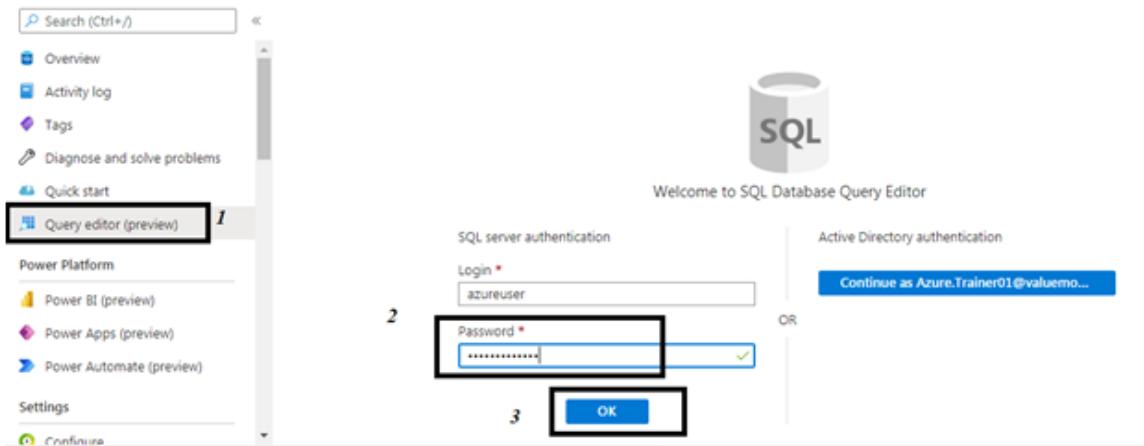
Your deployment is complete

Deployment name: Microsoft.SQLDatabase.newDatabaseNewServ...  
Subscription: Visual Studio Enterprise Subscription – MPN  
Resource group: newgroup2

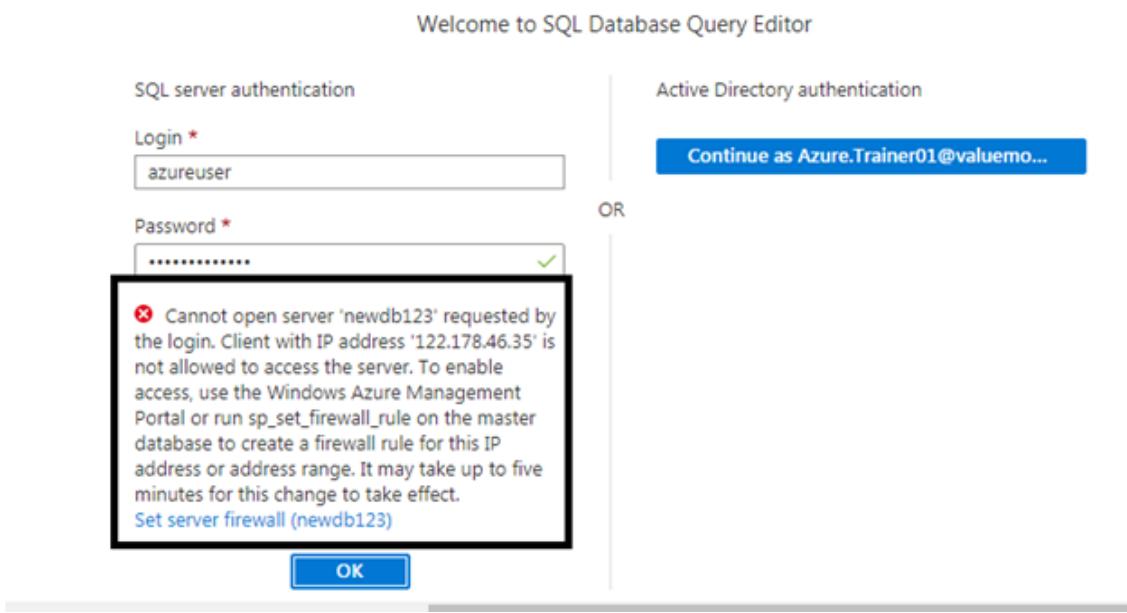
✓ Deployment details (Download)

^ Next steps

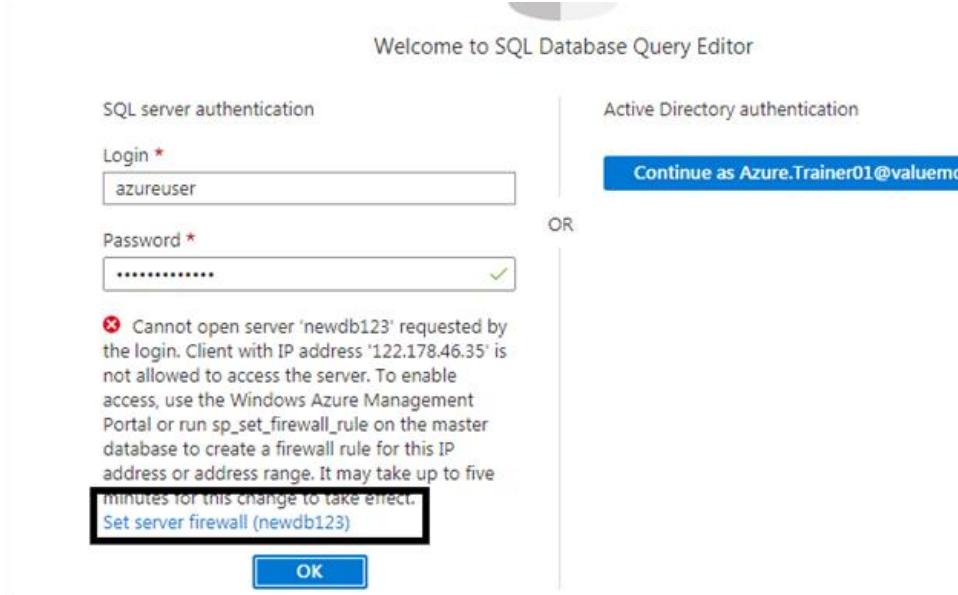
### Step 9: Go to the Query editor in the left, give the credentials as mentioned in the Basics

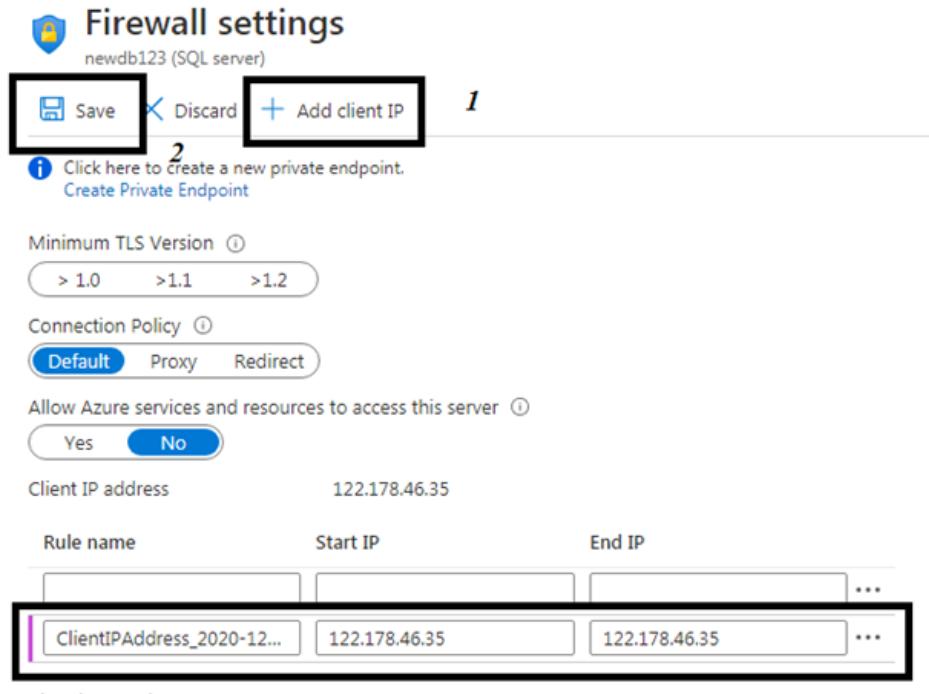


Note: Here we will be getting an error due to the firewall was not created

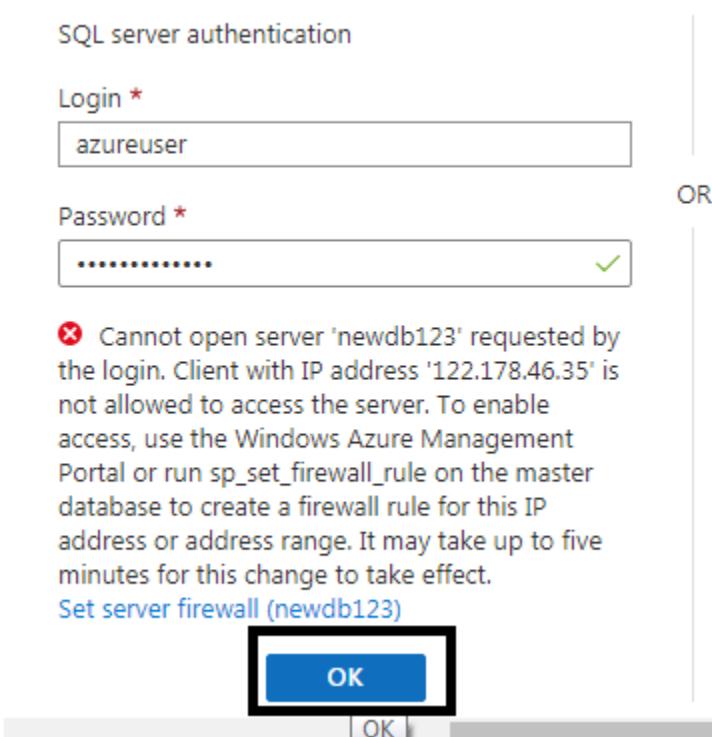


## Step 10: Setup a server firewall





### After creating the firewall try logging in the query editor



### Step 11: Create a query in the query editor and run the query

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. At the top, there's a toolbar with buttons for 'Run' (highlighted with a red box), 'Cancel query', 'Save query', 'Export data as', and 'Show only Editor'. Below the toolbar is a query editor window titled 'Query 1'. It contains a single line of T-SQL code: 'select \* from [SalesLT].[Customer];'. To the right of the code, there's a small number '1'. Below the query editor is a results grid. The grid has four columns: 'CustomerID', 'NameStyle', 'Title', and 'FirstName'. There are two rows of data: the first row has CustomerID 1, NameStyle False, Title Mr., and FirstName Orlando; the second row has CustomerID 2, NameStyle False, Title Mr., and FirstName Keith. A search bar labeled 'Search to filter items...' is located above the results grid. At the bottom of the results grid, a yellow bar indicates 'Query succeeded | 3s'.

Delete the database if it is a trial version

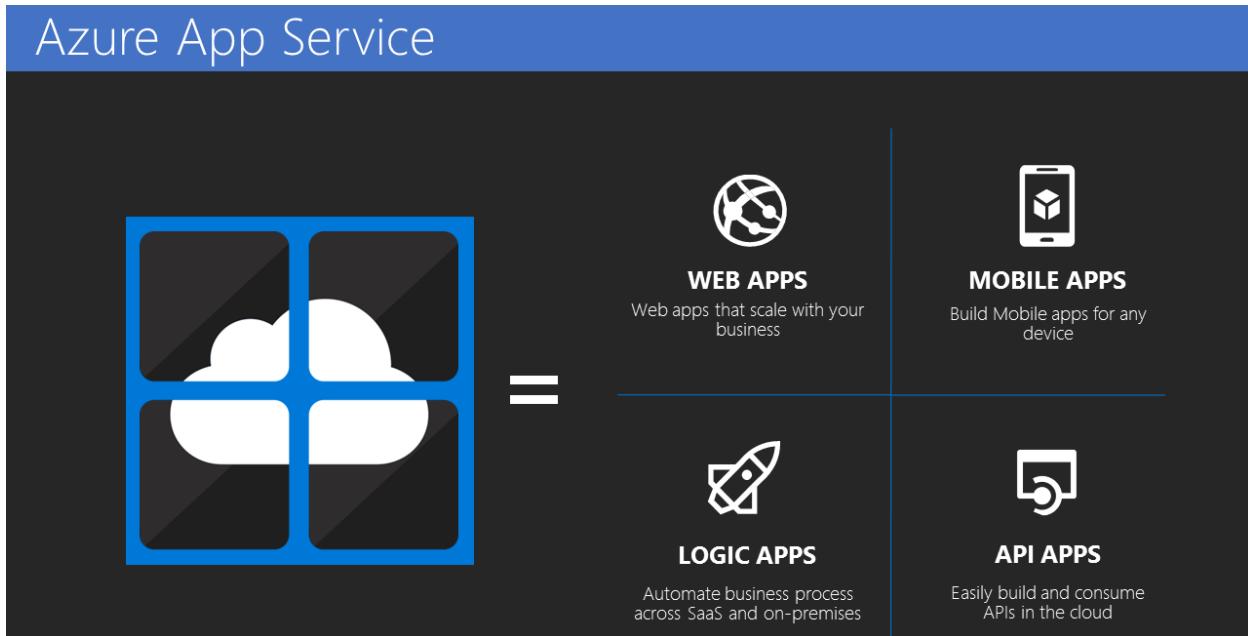
---

## **Azure Web Service Deployment**

*Azure App Service* is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on both Windows and [Linux](#)-based environments.

App Service not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. You can also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and TLS/SSL certificates.

With App Service, you pay for the Azure compute resources you use. The compute resources you use are determined by the *App Service plan* that you run your apps on.



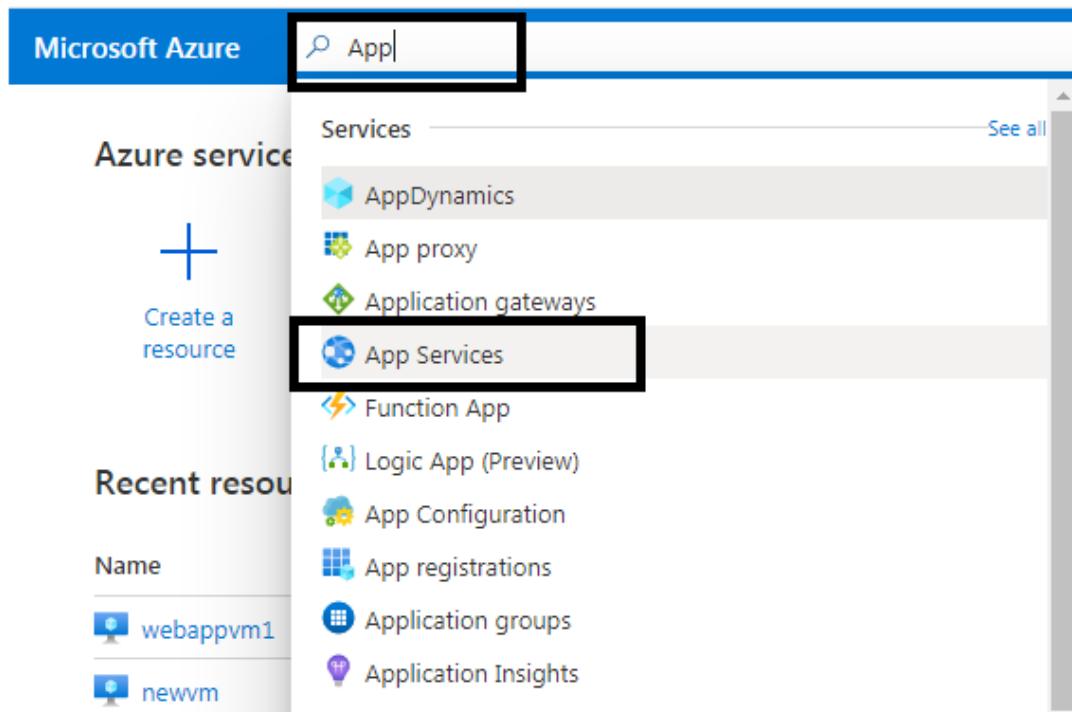
## Advantages

- **Multiple languages and frameworks** - App Service has first-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can also run PowerShell and other scripts or executables as background services.
- **Managed production environment** - App Service automatically patches and maintains the OS and language frameworks for you. Spend time writing great apps and let Azure worry about the platform.
- **Containerization and Docker** - Dockize your app and host a custom Windows or Linux container in App Service. Run multi-container apps with Docker Compose. Migrate your Docker skills directly to App Service.
- **DevOps optimization** - Set up continuous integration and deployment with Azure DevOps, GitHub, Bitbucket, Docker Hub, or Azure Container Registry. Promote updates through test and staging environments. Manage your apps in App Service by using Azure PowerShell or the cross-platform command-line interface (CLI).
- **Global scale with high availability** - Scale up or out manually or automatically. Host your apps anywhere in Microsoft's global datacenter infrastructure, and the App Service SLA promises high availability.
- **Connections to SaaS platforms and on-premises data** - Choose from more than 50 connectors for enterprise systems (such as SAP), SaaS services (such as Salesforce), and internet services (such as Facebook). Access on-premises data using Hybrid Connections and Azure Virtual Networks.

- **Security and compliance** - App Service is ISO, SOC, and PCI compliant. Authenticate users with [Azure Active Directory](#), [Google](#), [Facebook](#), [Twitter](#), or [Microsoft account](#). Create [IP address restrictions](#) and [manage service identities](#).
- **Application templates** - Choose from an extensive list of application templates in the [Azure Marketplace](#), such as WordPress, Joomla, and Drupal.
- **Visual Studio and Visual Studio Code integration** - Dedicated tools in Visual Studio and Visual Studio Code streamline the work of creating, deploying, and debugging.
- **API and mobile features** - App Service provides turn-key CORS support for RESTful API scenarios, and simplifies mobile app scenarios by enabling authentication, offline data sync, push notifications, and more.
- **Serverless code** - Run a code snippet or script on-demand without having to explicitly provision or manage infrastructure, and pay only for the compute time your code actually uses (see [Azure Functions](#))

## Deploying a Webapp Service

### Step 1: Go to portal and search for App services



### Step 2: Click Add and create an app service

Home >

## App Services

ValueMomentum Inc.

+ Add Manage view Refresh Export to CSV Open query Assign tags Start Restart

Filter Add... Subscription == all Resource group == all Location == all Add filter

Showing 0 to 0 of 0 records.

Name ↑↓	Status ↑↓	Location ↑↓	Pricing Tier ↑↓	App Service Plan ↑↓
---------	-----------	-------------	-----------------	---------------------

### Step 3: Enter the details in the Basics tab

#### Create Web App

Subscription \*

Resource Group \*  [Create new](#)

Instance Details

Name \*  .azurewebsites.net

Publish \*  Code  Docker Container

Runtime stack \*

Operating System \*  Linux  Windows

Region \*  [Not finding your App Service Plan? Try a different region.](#)

### Step 4: Select region then than East US2 as it doesn't consist any App Service Plan

Operating System \*  Linux  Windows

Region \*  [Not finding your App Service Plan? Try a different region.](#)

### Step 5: Create a new app service plan

## App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app.

Learn more [↗](#)

The screenshot shows the 'App Service Plan' creation interface. At the top, there's a list item 'Linux Plan (Central US) \*'. To its right, a modal window titled 'New App Service Plan' is open, with the 'Name' field containing 'newplanforapp'. Below the name field are 'OK' and 'Cancel' buttons. At the bottom of the main screen, there are 'Review + create' and '< Previous' buttons. A status message 'teams.microsoft.com is sharing y' is visible on the right.

## Step 6: Select the Sku and size

To Dev/Test and select 1GB

The screenshot shows the 'Sku and size' selection step. On the left, it says 'Sku and size \*'. On the right, a purple box highlights the 'Premium V2 P1v2' option, which includes '210 total ACU, 3.5 GB memory'. Below this is a 'Change size' button.

The screenshot shows the 'Spec Picker' dialog. It has a header 'Spec Picker' and a close button 'X'. Below the header are three options: 'Dev / Test' (selected), 'Production', and 'Isolated'. A note below says 'The first Basic (B1) core for Linux is free for the first 30 days!'. Under 'Recommended pricing tiers', there are two options: 'F1' (orange box) and 'B1' (green box). The F1 box contains '1 GB memory', '60 minutes/day compute', and 'Free'. The B1 box contains '100 total ACU', '1.75 GB memory', 'A-Series compute equivalent', and '868.50 INR/Month (Estimated)'. There is also a link 'See additional options'.

## Step 7: Click Review & Create and check for the deployment

# Create Web App

Publish  
Runtime stack

Code  
Python 3.7

## App Service Plan (New)

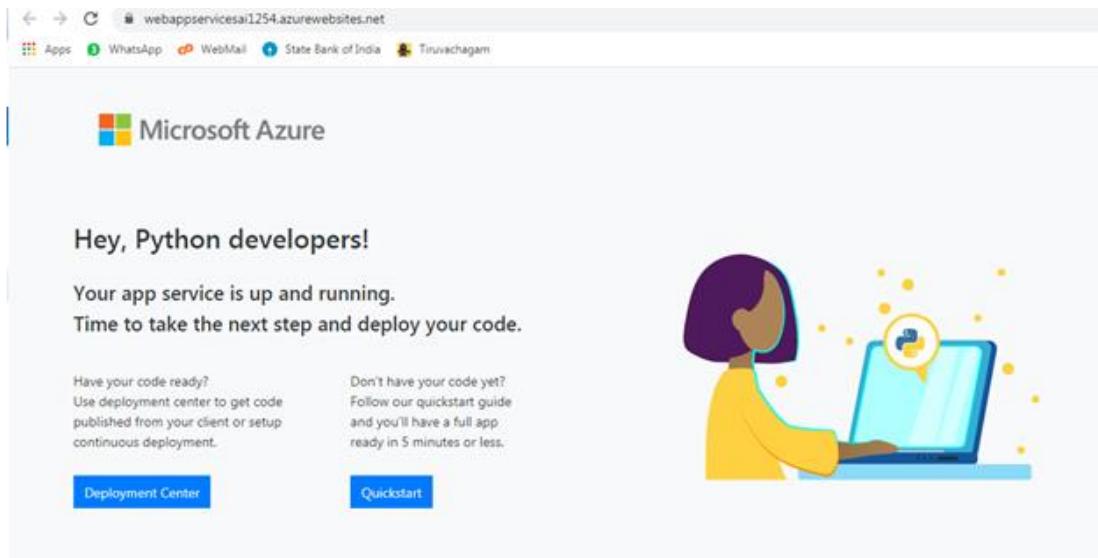
Name	newplanforapp
Operating System	Linux
Region	East US 2
SKU	Free
ACU	Shared infrastructure
Memory	1 GB memory

## Monitoring

Application Insights Not enabled

The screenshot shows the Azure portal interface for creating a new App Service plan. At the top, there is a navigation bar with a 'Create' button highlighted with a black border. Below the navigation bar, the page title is 'webappservicesai1254' with an 'X' icon. The main content area has a header 'App Service'. On the left, there is a sidebar with links: 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', and 'Diagnose and solve problems'. The 'Overview' link is selected and highlighted with a blue border. In the center, under the 'Essentials' section, the 'Resource group (change)' dropdown is set to 'newgroup2'. The 'Status' field shows 'Running'. To the right, a summary box displays the 'URL' as 'https://webappservicesai1254.azurewebsites.net', the 'App Service Plan' as 'newplanforapp (F1: Free)', and the 'Location' as 'East US 2'. There are also links for 'Get publish profile', 'Reset publish profile', and 'Send us'.

**Step 8: Copy the URL in the resource and paste it in the Browser**



Delete the app service if it is a trial version

## Deploying Custom Website in Azure

### **Step 1: Download the HTML code from the git website**

<https://github.com/rampatra/wedding-website>

A screenshot of a GitHub repository page for 'rampatra/wedding-website'. The repository has 2 branches and 1 tag. The commit history shows several initial commits and a fix for a broken symlink. On the right side, there is a 'Code' dropdown menu with options like 'Clone' and 'Download ZIP'. The 'Download ZIP' button is highlighted with a red box and labeled '3'. The URL 'https://github.com/rampatra/wedding-website' is also visible in the dropdown.

### **Step 1: Create a Virtual machine**

## Create a virtual machine

Region \* ⓘ (US) East US 2

Availability options ⓘ No infrastructure redundancy required

Image \* ⓘ Ubuntu Server 18.04 LTS - Gen1  
See all images

Azure Spot instance ⓘ

Size \* ⓘ Standard\_B1s - 1 vcpu, 1 GiB memory (₹501.80/month)  
See all sizes

Administrator account  SSH public key

[Review + create](#) [< Previous](#) [Next : Disks >](#)

## Step 2: Download WINSCP from the external website

[WinSCP :: Official Site :: Free SFTP and FTP client for Windows](#)

Download latest WinSCP 5.17 2020-11-20

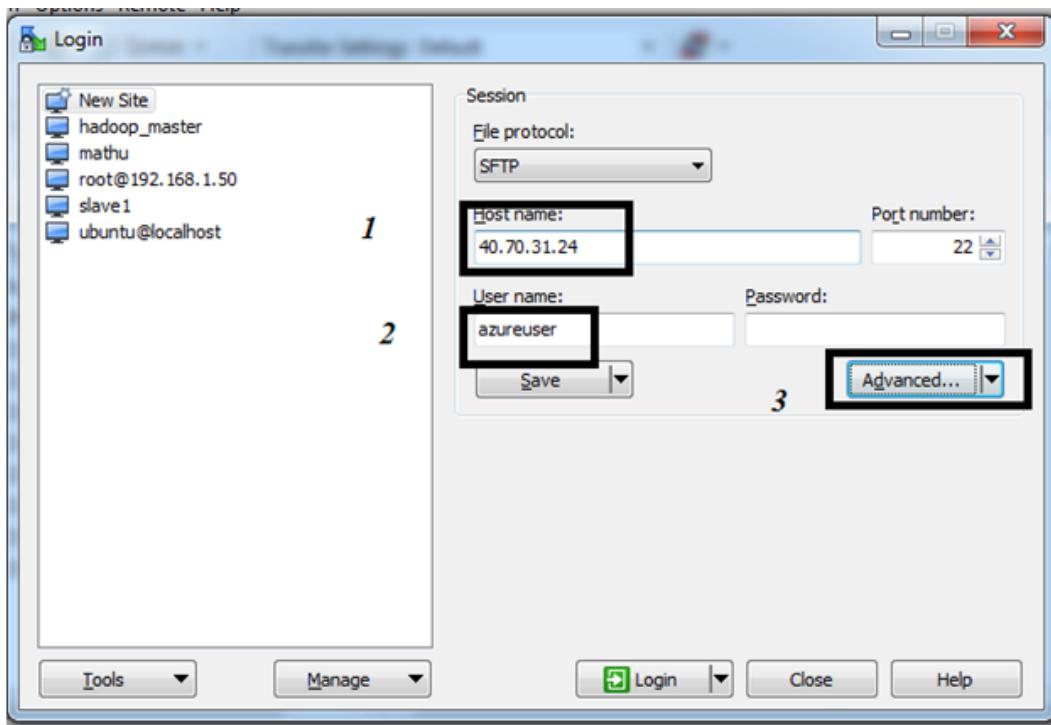
WinSCP 5.17 is a major application update. New features and enhancements include:

- Improvements to sessions and workspace management, so that WinSCP can now easily restore tabs that were open when it was last closed.
- Hardware-accelerated AES.
- Extension *Archive and Download* to archive remote files and download the archive.
- Improvements to Synchronization checklist window.
- Allowed sorting of find results.
- SSH core upgraded to PuTTY 0.73.
- The binaries are signed with new EV certificate valid until February 2023.

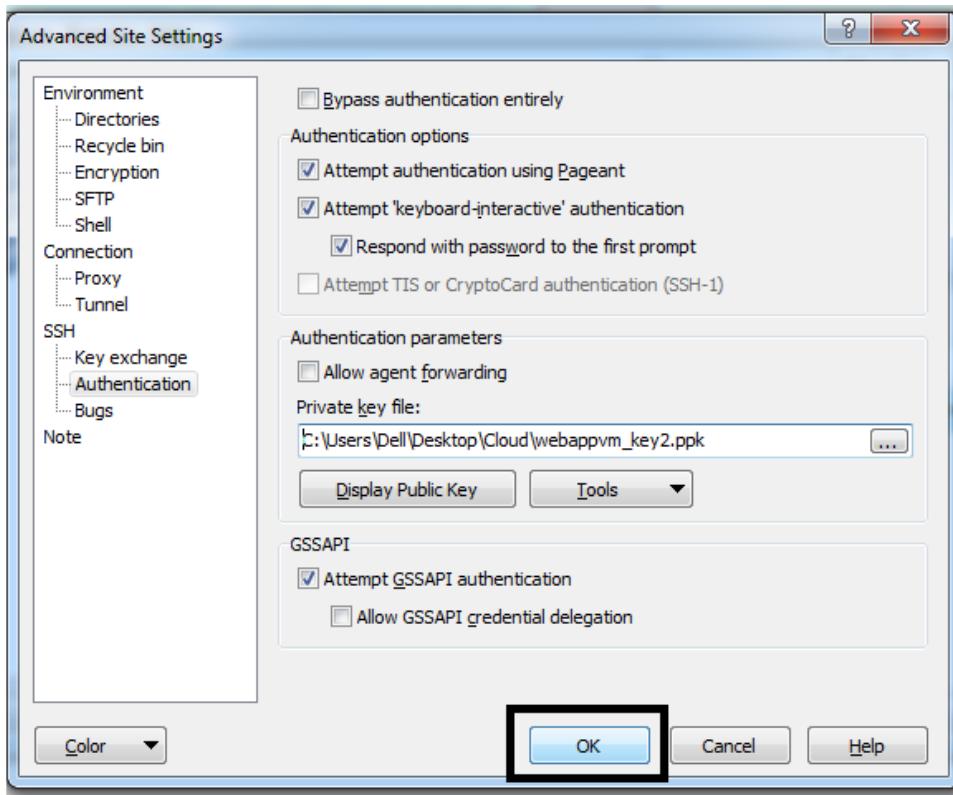
[DOWNLOAD WINSSCP 5.17.9 \(10.6 MB\)](#) Get it from Microsoft [LIST OF CHANGES](#)

1,958,657 downloads since release [What is this?](#)

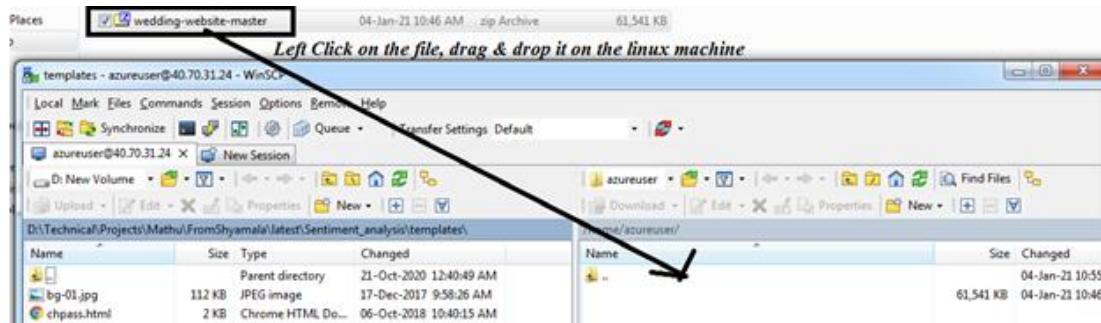
## Step 3: Open the WINSCP and paste the Public IP address



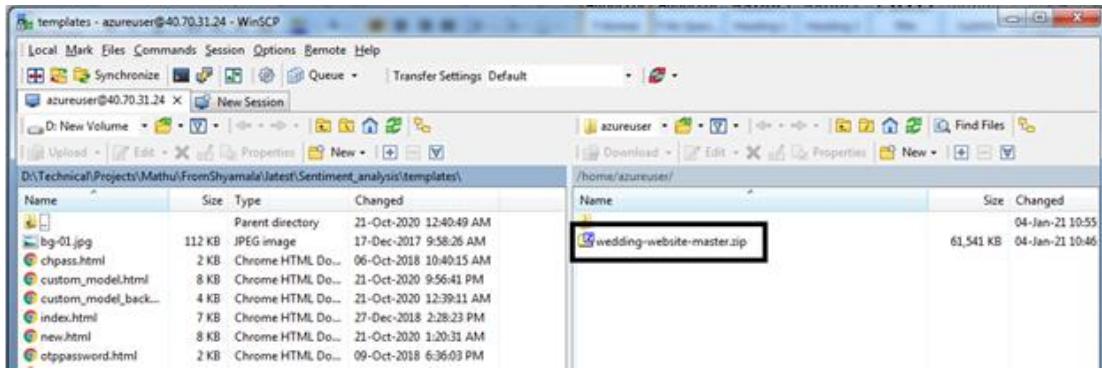
**Step 4: Open Authentication and browse the .ppk file already present in the system**



### Step 5: Login in the WINSCP and drag & drop the wedding-website.zip file in WINSCP



### Step 6: After uploading the file we'll see that the file is present in the Linux machine



### Step 7: Open putty and copy the Public IP to open Linux machine

```
System information as of Mon Jan  4 05:43:30 UTC 2021

System load:  0.0          Processes:           111
Usage of /:   4.7% of 28.90GB  Users logged in:    0
Memory usage: 21%
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@ubuntuwebserver:~$ ls
wedding-website-master.zip
azureuser@ubuntuwebserver:~$
```

Execute the following commands in the Linux

```
sudo apt-get update
```

```
sudo apt-get install -y unzip
```

```
unzip wedding-website-master.zip
```

### Step 8: Execute ls and you'll see that the wedding-website folder is available

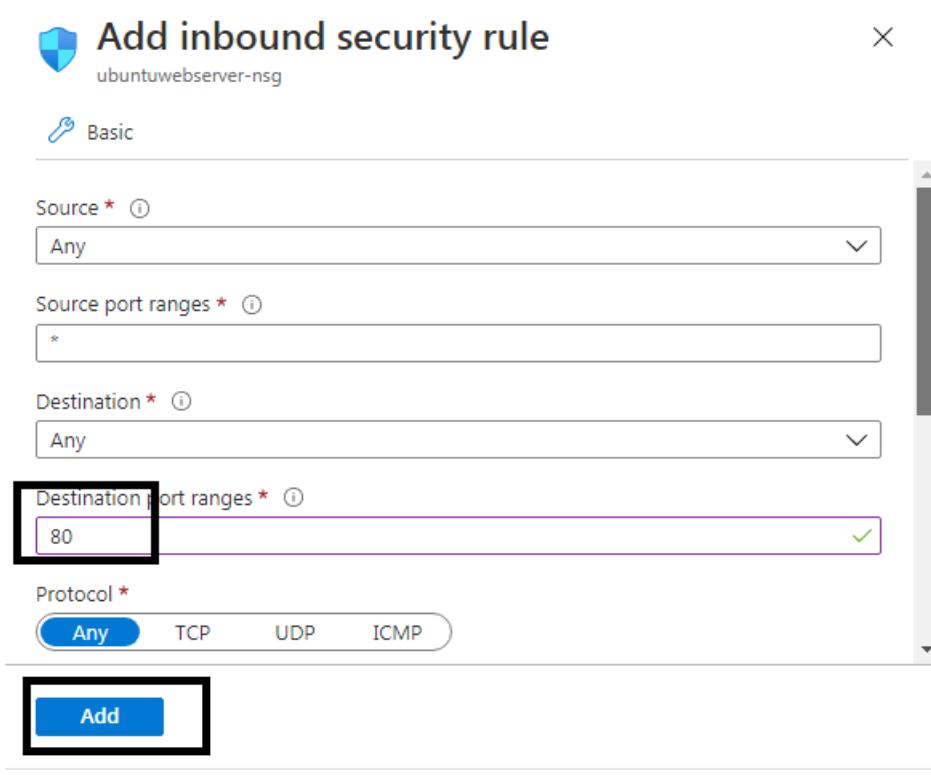


```
> ../../.. /mkdirp/bin/cmd.js
azureuser@ubuntuwebserver:~$ ls
wedding-website-master  wedding-website-master.zip
azureuser@ubuntuwebserver:~$
```

### Step 9: Execute the command to open the apache web server

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

Create an inbound rule in the networking tab in the VM

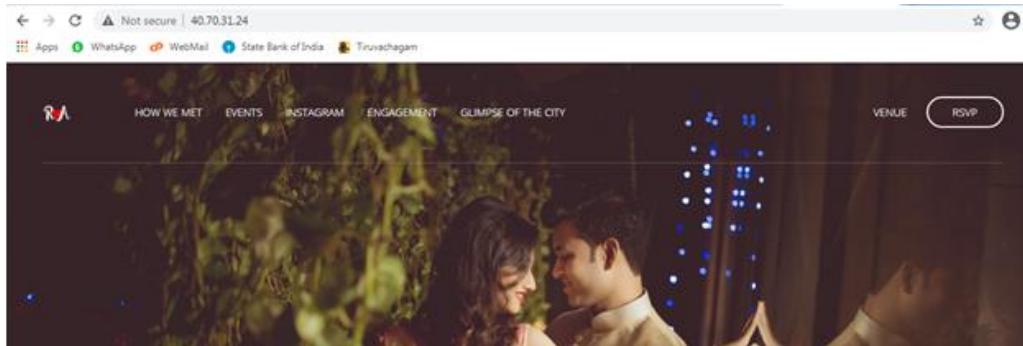


Paste the IP address in the browser and open the apache web



**Now, execute the following command**

```
sudo cp -r wedding-website-master/* /var/www/html/
```



## **Deploying a custom Website in Azure App Service**

**Step 1:** Open the portal and create an App Service

A screenshot of the Microsoft Azure portal. At the top, there's a blue header bar with the Microsoft Azure logo and a search bar. Below the header, the main dashboard is titled "Azure services". It features several icons: a plus sign for "Create a resource", a monitor for "Virtual machines", a box with a globe for "App Services" (which is highlighted with a black rectangular selection box), a cloud icon for "App Service plans", and another cloud icon for "CDN profiles". Below the dashboard, there's a section titled "Recent resources".

## Create Web App

Create new

Instance Details

Name \*  .azurewebsites.net

Publish \*  Code  Docker Container

Runtime stack \*

Operating System \*  Linux  Windows

Region \* Central US

Not finding your App Service Plan? Try a different region.

**Review + create** < Previous Next : Monitoring >

NOTE: During the selection of Sku and Storage check that the size should be "Standard S1 (100 total ACU, 1.75GB memory)

## Create Web App

Region \* Central US

Not finding your App Service Plan? Try a different region.

App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app.

[Learn more](#)

Linux Plan (Central US) \*  ASP-newgroup-b18e (S1)

Create new

Sku and size \* Standard S1

100 total ACU, 1.75 GB memory

**Review + create** < Previous Next : Monitoring >

**Step 2: Change the size of the App service and select Review + Create**

# Create Web App

Basics    Monitoring    Tags    Review + create

Summary

 **Web App**  
by Microsoft

Details

Subscription	2d1eaf32-f707-40db-a62c-2b7d297b42a6
Resource Group	newgroup2
Name	appsabservice124
Publish	Code
Runtime stack	DHD 7.2

**Create**    < Previous    Next >    Download a template for automation

## Step 3: After successful deployment of App service go to resource and copy the URL

Home > Microsoft.Web-WebApp-Portal-9b181bd7-b20a >

 **appsabservice124** 

App Service

Search (Ctrl+F)          Get publish profile  Send us your feedback

Overview    Essentials

Activity log    Access control (IAM)    Tags    Diagnose and solve problems    Security    Events (preview)

Resource group (change)  
newgroup2

Status  
Running

Location  
Central US

Subscription (change)  
Visual Studio Enterprise Subscription - MPN

Subscription ID  
2d1eaf32-f707-40db-a62c-2b7d297b42a6

Tags (change)

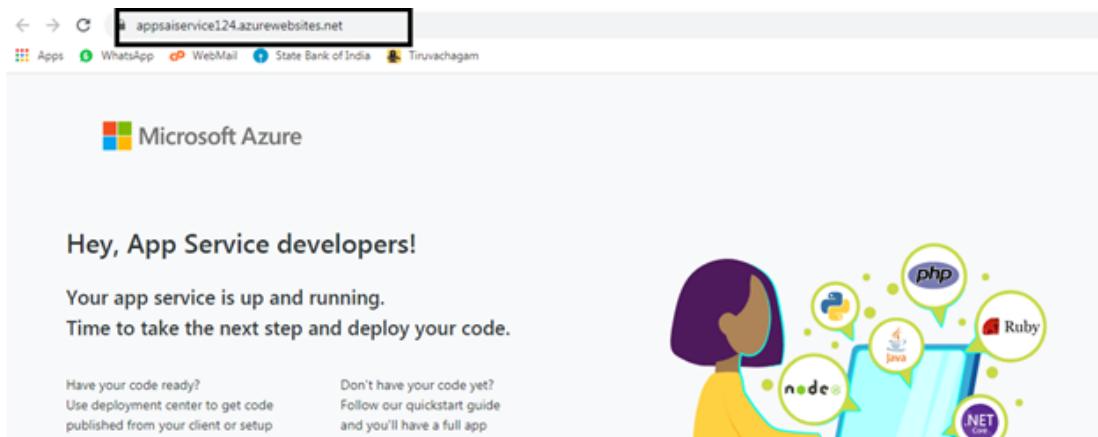
URL  
**https://appsabservice124.azurewebsites.net**

App Service Plan  
ASP-newgroup-b1f6 (S1: 1)

FTP deployment username  
No FTP deployment user set

FTP hostname  
ftp://waws-prod-dm1-181 ftp.azurewebsites.windows.net/site/wwwroot

FTPS hostname  
https://waws-prod-dm1-181.ftp.azurewebsites.windows.net/site/wwwroot



Step 4: Now got to resource and Select SSH

A screenshot of the Azure portal showing the 'appaiservice124 | SSH' blade. On the left, a sidebar lists 'Quotas', 'Change App Service plan', 'Development Tools' (which includes 'Clone App', 'SSH' [selected and highlighted with a black box], 'Advanced Tools', and 'Extensions'), and a search bar. On the right, the 'SSH' section is displayed with the sub-header 'SSH provides a Web SSH console experience for your Linux App code. Learn more'. A 'Go →' button is visible.

Step 5: Now a prompt will open or Linux machine will open

A screenshot of a terminal session titled 'APP SERVICE ON LINUX'. The title is composed of various ASCII characters. The session shows a root prompt: 'root@b3123172847a:/home#'. The terminal also displays documentation links and a note about data persistence.

Step 6: Execute “ls” command and “ls site”

 Apps  WhatsApp  WebMail  State Bank of India  Tiruvachagam

Step 7: Now we need to access the root folder

 Apps  WhatsApp  WebMail  State Bank of India  Tiruvachagam

```
/ [ ] \ \ [ ] / [ ] \ [ ] \ [ ] >
\ [ ] \ [ ] \ [ ] / [ ] \ [ ] \ [ ] \
A P P S E R V I C E O N L I N U X

Documentation: http://aka.ms/webapp-linux
PHP quickstart: https://aka.ms/php-qs
PHP version : 7.2.33
Note: Any data outside '/home' is not persisted
root@b3123172847a:/home# ls
ASP.NET LogFiles site
root@b3123172847a:/home# ls site/
deployments locks repository wwwroot
root@b3123172847a:/home# 
```

ls site/wwwroot/

 Apps  WhatsApp  WebMail  State Bank of India  Tiruvachagam

```
/ [ ] \ \ [ ] / [ ] \ [ ] \ [ ] >
\ [ ] \ [ ] \ [ ] / [ ] \ [ ] \ [ ] \
A P P S E R V I C E O N L I N U X

Documentation: http://aka.ms/webapp-linux
PHP quickstart: https://aka.ms/php-qs
PHP version : 7.2.33
Note: Any data outside '/home' is not persisted
root@b3123172847a:/home# ls
ASP.NET LogFiles site
root@b3123172847a:/home# ls site/
deployments locks repository wwwroot
root@b3123172847a:/home# ls site/wwwroot/
hostingstart.html
root@b3123172847a:/home# 
```

Step 8: We should access the git in the Linux machine

It shows a pop up saying that the git: command is not found

So, we need to install the git

```
root@b3123172847a:/home# git  
-bash: git: command not found
```

Step 9: Execute the commands

```
sudo apt-get update
```

```
Sudo apt-get install git -y
```

```
root@b3123172847a:/home# apt-get update  
Ign:1 http://deb.debian.org/debian stretch InRelease  
Get:2 http://deb.debian.org/debian stretch-updates InRelease [93.6 kB]  
Get:3 http://security.debian.org/debian-security stretch/updates InRelease [53.0 kB]  
Get:4 http://deb.debian.org/debian buster InRelease [121 kB]  
Get:5 http://deb.debian.org/debian buster-updates InRelease [51.9 kB]  
Get:6 http://security.debian.org/debian-security buster/updates InRelease [65.4 kB]  
Hit:7 http://deb.debian.org/debian stretch Release  
0% [2 InRelease gpgv 93.6 kB]
```

```
Fetched 9317 kB in 11s (806 kB/s)  
Reading package lists... Done  
root@b3123172847a:/home# apt-get install -y git  
Reading package lists... vs
```

Step 10: Now the git is installed in the Linux machine, we need to clone the website in the Linux machine using the command

```
git clone https://github.com/rampatra/wedding-website
```

```
root@b3123172847a:/home# git clone https://github.com/rampatra/wedding-website  
Cloning into 'wedding-website'...
```

Step 11: Now we should opt the wedding-website folder in the Linux machine

```
cd wedding-website
```

```
root@b3123172847a:/home# ls
ASP.NET LogFiles site wedding-website
root@b3123172847a:/home# cd wedding-website
root@b3123172847a:/home/wedding-website#
root@b3123172847a:/home/wedding-website# ls
CNAME           css          gulpfile.js
LICENSE         favicon-16x16.png img
README.md       favicon-256x256.png index.html
android-chrome-192x192.png favicon-32x32.png js
android-chrome-256x256.png favicon-64x64.png manifest.json
apple-touch-icon.png favicon.ico mstile-150x150.png
browserconfig.xml fonts        node_modules
root@b3123172847a:/home/wedding-website#
```

Step 12: Now we should access the wedding-website to the root so that it opens in the browser

```
cp -r ../../site/wwwroot
```

```
cd /home/site/wwwroot
```

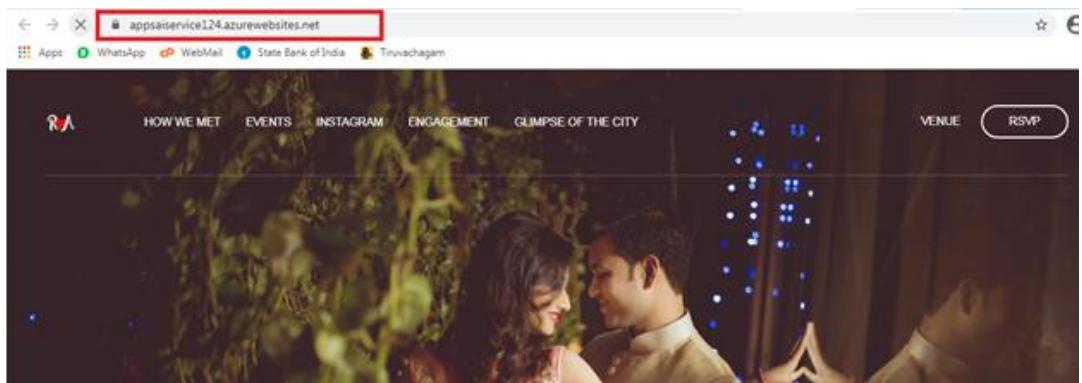
```
browserconfig.xml      fonts          node_modules
root@b3123172847a:/home/wedding-website# cp -r ../../site/wwwroot
[redacted]
```

```
root@b3123172847a:~# cd /home
root@b3123172847a:/home# cd /home/site/wwwroot
root@b3123172847a:/home/site/wwwroot# pwd
/home/site/wwwroot
root@b3123172847a:/home/site/wwwroot#
```

Step 13: Now if we execute ls we'll see that the files in the wedding-website folder are shown

```
root@b3123172847a:/home/site/wwwroot# ls
CNAME           favicon-16x16.png   img
LICENSE         favicon-256x256.png index.html
README.md       favicon-32x32.png   js
android-chrome-192x192.png favicon-64x64.png manifest.json
android-chrome-256x256.png favicon.ico   mstile-150x150.png
apple-touch-icon.png fonts        node_modules
browserconfig.xml gulpfile.js
css            hostingstart.html
root@b3123172847a:/home/site/wwwroot#
```

Step 14: Go to the browser and refresh the URL page



## Getting access to the logs:

Home > Microsoft.Web-WebApp-Portal-9b181bd7-b20a > appaiservice124

### 🔗 appaiservice124 | Diagnose and solve problems

App Service

Search (Ctrl+ /)

Home

Overview

Activity log

Access control (IAM)

Tags

**Diagnose and solve problems**

Security

Events (preview)

Deployment

Availability and Performance

Is your app experiencing downtime or slowness? Discover issues that may impact SLA, caused by your app itself or Azure.

Keywords: Downtime, 5xx Errors, 4xx Errors, CPU, Memory

Configuration and Management

Find out if your app service features are misconfigured.

Keywords: Backups, Slots, Swaps, Scaling

### 🔗 appaiservice124 | Diagnose and solve problems

App Service

Search (Ctrl+ /)

Home

Availability and Performance

Here are some issues related to Availability and Performance. Please select the tile that best describes your issue.

Application Logs

Container Crash

Container Issues

CPU Usage

Linux - Host Disk Space Usage

Memory Usage

Port Usage

Process Full List

Process List

SNAT Port Exhaustion

Swap Effects on Availability

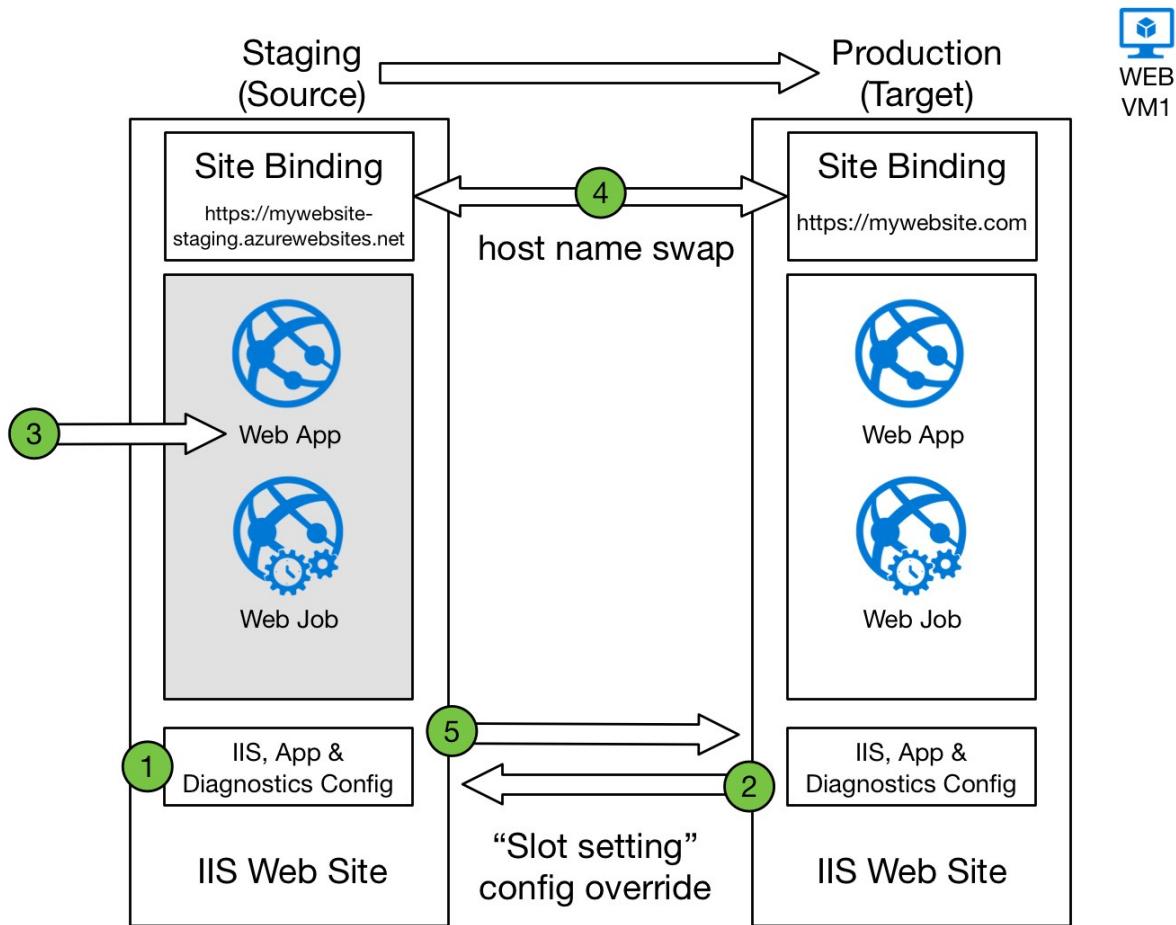
TCP Connections for Linux Apps

Testing in Production

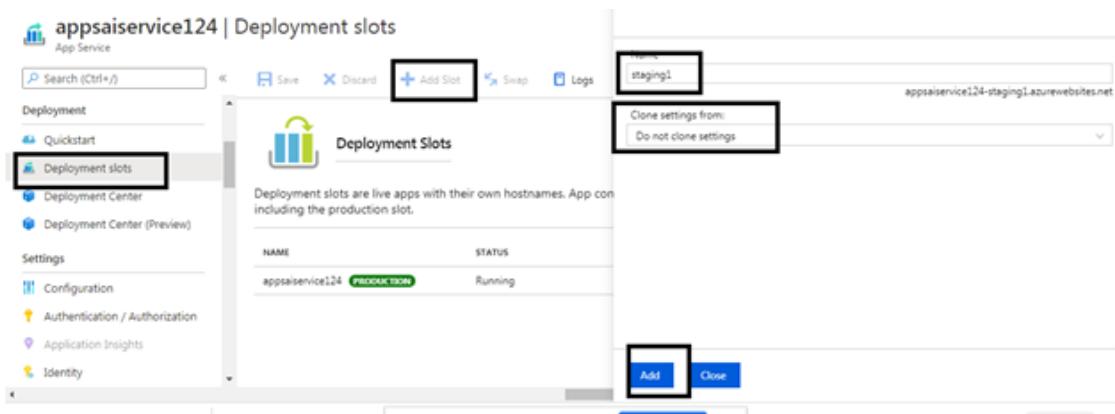
Web App Down

Web App Restarted

## Swap Process in App Service



## Creating Deployment slots in App Service



Save Discard Add Slot Swap Logs

### Deployment Slots

Deployment slots are live apps with their own hostnames. App content and configurations can be swapped between two deployment slots, including the production slot.

NAME	STATUS
appaiservice124 <b>PRODUCTION</b>	Running
appaiservice124-staging1	Running

Name: staging1  
Clone settings from: Do not clone settings

Successfully created slot 'staging1'

Add Close

Save Discard Add Slot Swap Logs Refresh

### Deployment Slots

Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the production slot.

NAME	STATUS	APP SERVICE PLAN	TRAFFIC %
appaiservice124 <b>PRODUCTION</b>	Running	ASP-newgroup-b18e	100
appaiservice124-staging1	Running	ASP-newgroup-b18e	0

Save Discard Add Slot Swap Logs Refresh

### Deployment Slots

Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the production slot.

NAME	STATUS	APP SERVICE PLAN	TRAFFIC %
appaiservice124 <b>PRODUCTION</b>	Running	ASP-newgroup-b18e	100
appaiservice124-staging1	Running	ASP-newgroup-b18e	0

staging1 (appaiservice124/staging1) X

App Service (Slot)

Search (Ctrl+ /) JSON View

Browse Stop Swap Restart Delete Refresh Get publish profile Reset publish profile Send us your feedback

Overview Essentials

Activity log Access control (IAM) Tags Diagnose and solve problems Security Events (preview)

Deployment

Resource group (change)  
newgroup2

Status  
Running

Location  
Central US

Subscription (change)  
Visual Studio Enterprise Subscription – MPN

Subscription ID  
2d1eaef32-f707-40db-a62c-2b7d297b42a6

Tags (change)

URL  
<https://appaiservice124-staging1.azurewebsites.net>

App Service Plan  
ASP-newgroup-8118e (S1: 1)

FTP/deployment username  
No FTP/deployment user set

FTP hostname  
ftp://waws-prod-dm1-181.ftp.azurewebsites.windows.net/site/wwwroot

FTPS hostname  
https://waws-prod-dm1-181.ftp.azurewebsites.windows.net/site/wwwroot

appsaiservice124-staging1.azurewebsites.net ★

Apps WhatsApp WebMail State Bank of India Tiruvechagam

Microsoft Azure

## Hey, App Service developers!

Your app service is up and running.  
Time to take the next step and deploy your code.

Have your code ready?  
Use deployment center to get code published from your client or setup continuous deployment.

Don't have your code yet?  
Follow our quickstart guide and you'll have a full app ready in 5 minutes or less.



Home > appaiservice124 I

## appaiservice124 | Deployment slots

App Service

Search (Ctrl+ /) Save Discard Add Slot Swap Logs Refresh

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Security Events (preview)

Deployment Quickstart Deployment slots

**Deployment Slots**

Deployment slots are live apps with their own hostnames. App content and config including the production slot.

NAME	STATUS
appaiservice124 <b>PRODUCTION</b>	Running
appaiservice124-staging1	Running

## Swap

Source: appaiservice124-staging1

Target: **PRODUCTION**: appaiservice124

Swap with preview can only be used with sites that have deployment slot settings enabled

Perform swap with preview

### Config Changes

This is a summary of the final set of configuration changes on the source and target deployment slots after the swap has completed.

Source Changes		Target Changes	
SETTING	TYPE	OLD VALUE	NEW VALUE

**Swap** **Close**



Previous Production Code which had the custom site is now changed

<https://appaiservice124.azurewebsites.net/> -> This url has now the code in Previous Staging

<https://appaiservice124-staging1.azurewebsites.net/> -> This has now the wedding template which was deployed on Production earlier

## **Setting Environment Values from App Service Configuration:**

The screenshot shows the 'Configuration' blade for an Azure App Service named 'appaiservice124'. The left sidebar lists 'Deployment Center (Preview)', 'Settings' (with 'Configuration' highlighted), and 'Authentication / Authorization'. The main area has a search bar, refresh, save, and discard buttons. It includes a 'New application setting' button (highlighted with a red box), 'Show values', 'Advanced edit', and a 'Filter application settings' dropdown. A table below lists application settings with columns for Name, Value, and Source.

Name	Value	Source

Add/Edit application setting

Name	Google.Url
Value	www.google.com
<input type="checkbox"/> Deployment slot setting	

**OK** **Cancel**

Name is Case Sensitive

appsaiservice124 | Configuration

App Service

Search (Ctrl+ /) Refresh Save Discard

Application settings \* General settings Path mappings

**Application settings**

Application settings are encrypted at rest and transmitted over an encrypted connection. Application Settings are exposed as environment variables.

+ New application setting Show values Advanced edit

Filter application settings

Save changes

Any changes to applications settings and connection strings will restart your application. Are you sure you want to continue?

**Continue** **Cancel**

The screenshot shows the Azure portal interface for an app named 'appaiservice124'. On the left, there's a sidebar with a search bar and several navigation items: 'App Service plan', 'Quotas', 'Change App Service plan', 'Development Tools' (which is expanded), 'Clone App', 'SSH' (which is highlighted with a red box), and 'Advanced Tools'. On the right, under the 'SSH' tab, there's a brief description: 'SSH provides a Web SSH console experience for your Linux App Service Plan'. Below the description is a 'Go →' button, also highlighted with a red box.

in SSH tab:

```
echo $Google_Url

Documentation: http://aka.ms/webapp-linux
PHP quickstart: https://aka.ms/php-qs
PHP version : 7.2.33
Note: Any data outside '/home' is not persisted
root@9186366befa8:/home# echo $Google_Url
www.google.com
root@9186366befa8:/home#
```

### Changing the Environment from Older Version of PHP to Newer Version of PHP:

**appsaiservice124 | Configuration**

App Service

Save (highlighted)

Deployment slots Deployment Center Deployment Center (Preview)

General settings \* Path mappings

Stack settings

Stack: PHP

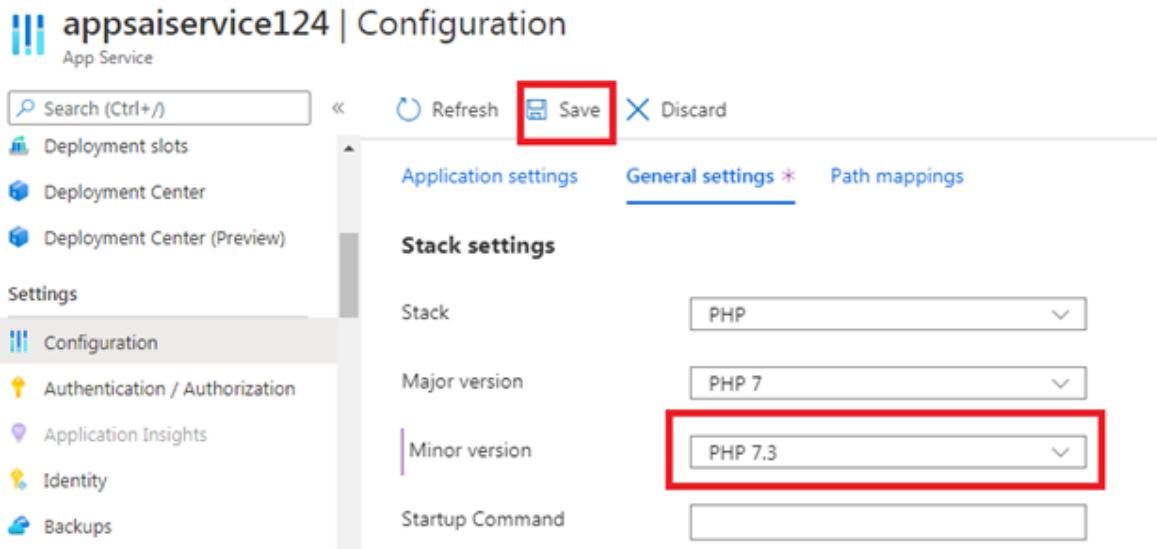
Major version: PHP 7

Minor version: PHP 7.3 (highlighted)

Startup Command:

Settings

- Configuration (highlighted)
- Authentication / Authorization
- Application Insights
- Identity
- Backups



### Platform Upgrade Confirmation, this will trigger an application restart

Home > appsaiservice124

**appsaiservice124 | Configuration**

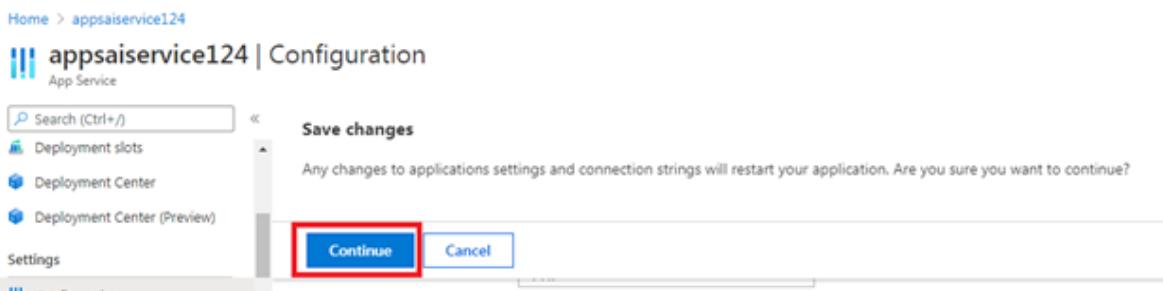
App Service

Save changes

Any changes to applications settings and connection strings will restart your application. Are you sure you want to continue?

Continue (highlighted) Cancel

Deployment slots Deployment Center Deployment Center (Preview)



**appsaiservice124 | TLS/SSL settings**

App Service

Search (Ctrl+/(highlighted)) Refresh Delete bindings Buy Certificate Troubleshoot FAQs

Bindings Private Key Certificates (.pfx) Public Key Certificates (.cer)

Protocol Settings

Protocol settings are global and apply to all bindings defined by your app.

HTTPS Only:  Off  On (highlighted)

Minimum TLS Version: 1.0 1.1 1.2

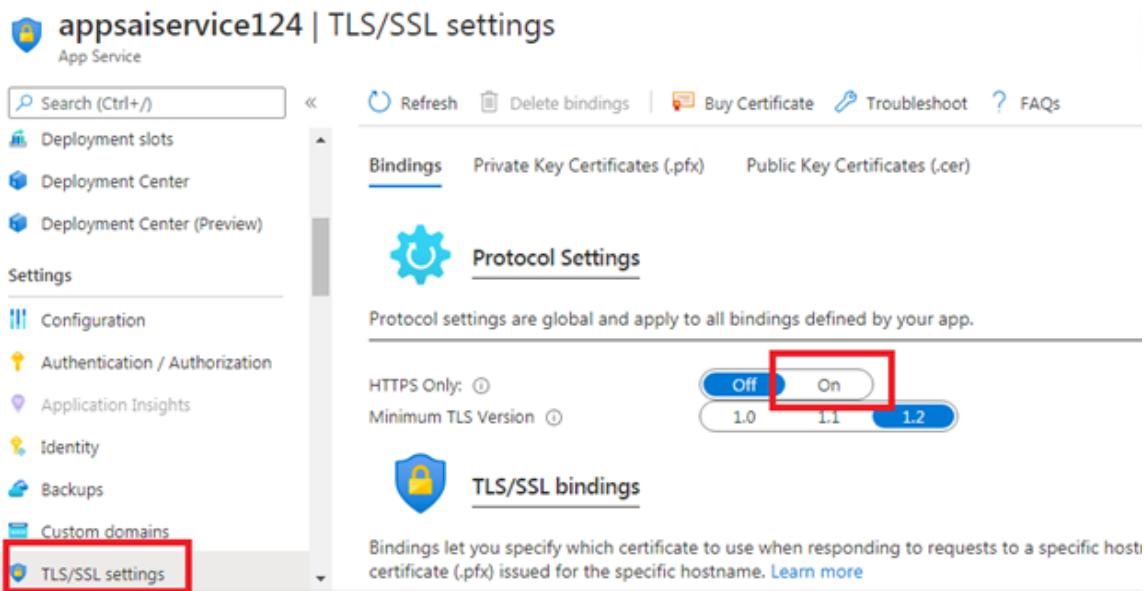
TLS/SSL bindings

Bindings let you specify which certificate to use when responding to requests to a specific host certificate (.pfx) issued for the specific hostname. [Learn more](#)

Deployment slots Deployment Center Deployment Center (Preview)

Settings

- Configuration
- Authentication / Authorization
- Application Insights
- Identity
- Backups
- Custom domains
- TLS/SSL settings (highlighted)



**appsaiservice124** App Service

Search (Ctrl+ /)

Overview (highlighted with red box)

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Security

Events (preview)

Deployment

Essentials

Resource group (change): newgroup2

Status: Running

Location: Central US

Subscription (change): Visual Studio Enterprise Subscription - MN

Subscription ID: 201aaef32-f707-40db-a62c-2b7d297b42a6

Tags (Inherit)

URL: <https://appsaiservice124.azurewebsites.net>

App Service URL: <https://appsaiservice124.azurewebsites.net>

ASP.NET: https://appsaiservice124.azurewebsites.net

FTP/deployment username: No FTP/deployment user set

FTP hostname: ftp://waws-prod-dm1-181.ftp.azurewebsites.windows.net/site/wwwroot

FTPS hostname: ftps://waws-prod-dm1-181.ftp.azurewebsites.windows.net/site/wwwroot

JSON View

**appsaiservice124 | Scale up (App Service plan)**

App Service

Search (Ctrl+ /)

TLS/SSL settings

Networking

Scale up (App Service plan) (highlighted with red box)

Scale out (App Service plan)

Webjobs

Push

MySQL In App

Properties

Locks

App Service plan

PIV3: 8 GB memory, 2 vCPU, 6176.03 INR/Month (Estimated)

P2V3: 16 GB memory, 4 vCPU, 12352.07 INR/Month (Estimated)

P3V3: 32 GB memory, 8 vCPU, 24704.13 INR/Month (Estimated)

Additional pricing tiers

S1: 100 total ACU, 1.75 GB memory, A-Series compute equivalent, 4583.77 INR/Month (Estimated)

S2: 200 total ACU, 3.5 GB memory, A-Series compute equivalent, 9167.55 INR/Month (Estimated)

S3: 400 total ACU, 7 GB memory, A-Series compute equivalent, 18335.10 INR/Month (Estimated)

Previously Selected Size

Included features: Every app hosted on this App Service plan will have access to these features:

Current Selection with higher Configuration

Included hardware: Every instance of your App Service plan will include the following:

Azure Compute Units (ACU)

Apply (highlighted with green box)

**appsaiservice124 | Scale out (App Service plan)**

App Service

Search (Ctrl+ /)

Save Discard Refresh Logs Provide feedback

Manual scale (highlighted with green box)

Custom autoscale

Override condition

Instance count: 1 (highlighted with green box)

[Save](#) [Discard](#) [Refresh](#) [Logs](#) [Provide feedback](#)

designed time windows. Autoscale enables your resource to be performant and cost effective by adding and removing instances based on demand. Learn more about Azure Autoscale or view the how-to video.

#### Choose how to scale your resource

Manual scale  
Maintain a fixed instance count

Custom autoscale  
Scale on any schedule, based on any metrics

Custom autoscale

Autoscale setting name \*

Resource group  [▼](#)

Delete warning (i) The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.

Scale mode  Scale based on a metric  Scale to a specific instance count

Rules (i) No metric rules defined; click Add a rule to scale out and scale in your instances based on rules. For example: 'Add a rule that increases instance count by 1 when CPU percentage is above 70%.'

[+ Add a rule](#)

Instance limits

Minimum <input type="radio"/>	Maximum <input type="radio"/>	Default <input type="radio"/>
<input type="text" value="1"/> ✓	<input type="text" value="2"/> ✓	<input type="text" value="1"/> ✓

This scale condition is executed when none of the other scale condition(s) match

## Scale rule

Enable metric divide by instance count (?)

**Operator \*** Metric threshold to trigger scale action \* (?)

edb Greater than  (%)

Duration (in minutes) \* (?)

10 minutes

Time grain (in mins) (?) Time grain statistic \* (?)

1 Average

**Action**

Operation \* Cool down (minutes) \* (?)

Increase count by

Instance count \*  (%)

**Add**

### appaiservice124 | Scale out (App Service plan)

App Service

Search (Ctrl+/) Save Discard Refresh Logs Provide feedback

The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.

TLS/SSL settings

Networking

Scale up (App Service plan)

Scale out (App Service plan) **Rules**

It is recommended to have at least one scale in a rule. To create new rules, click Add a rule.

### appaiservice124 | Change App Service plan

App Service

Search (Ctrl+/) Properties Locks

App Service plan App Service plan Quotas Change App Service plan Development Tools Clone App SSH Advanced Tools Extensions

DESTINATION PLAN DETAILS

App Service plan \*  Create new

Resource group newgroup

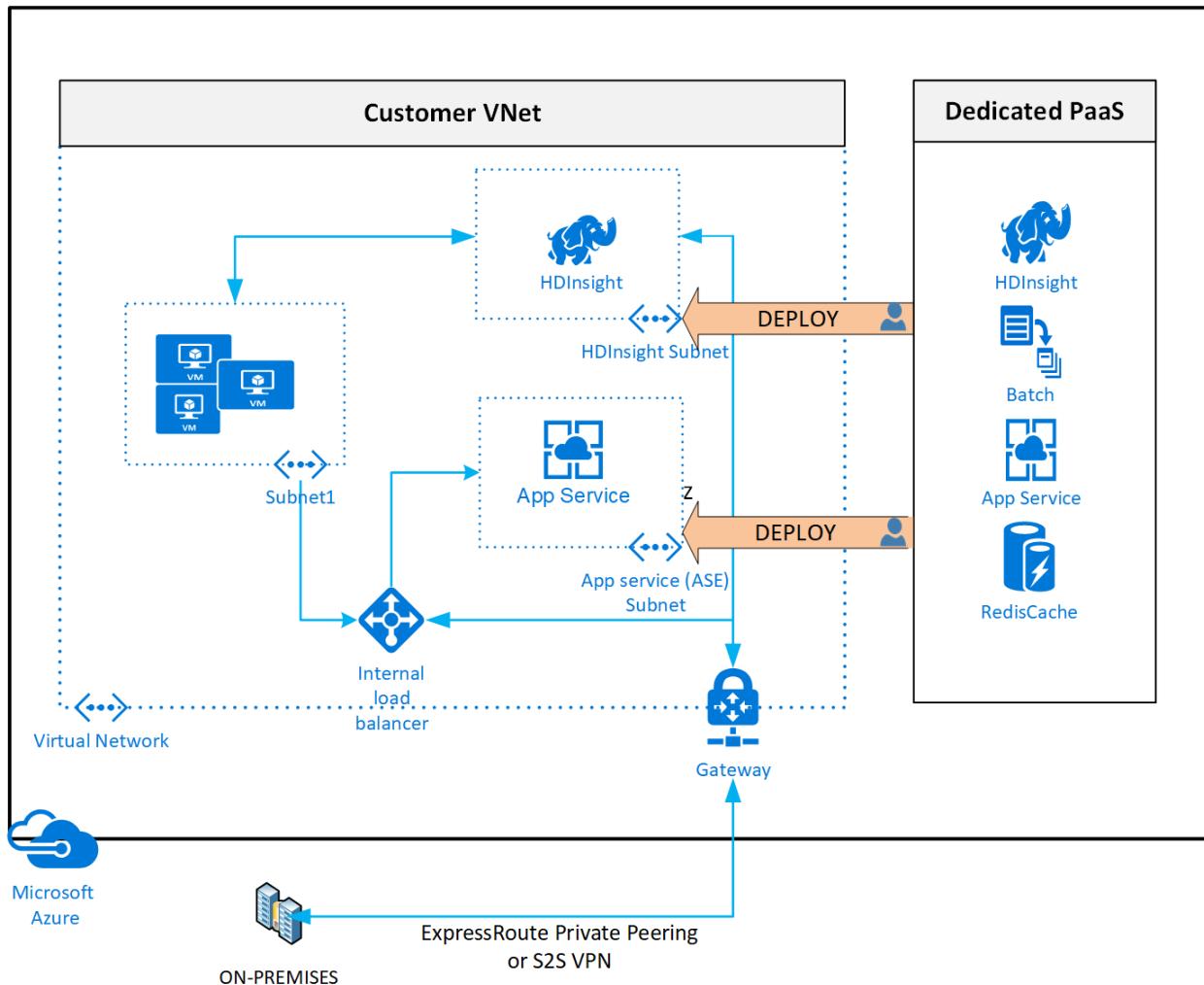
Region Central US

Pricing Tier Standard (\$2)

**OK**

## Virtual Networks

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.



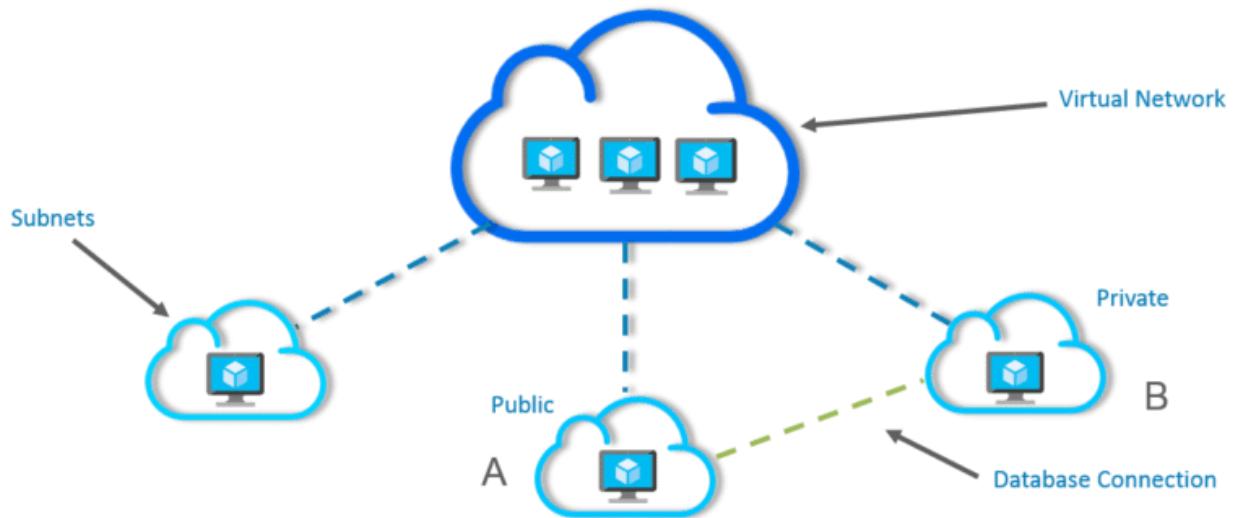
## Virtual Network in Azure

### Virtual Network Key Components

**Subnets:** Subnets enable segmenting a virtual network into one or more subnet networks and allocating a portion of the virtual network addresses space to each subnet. Azure resources are

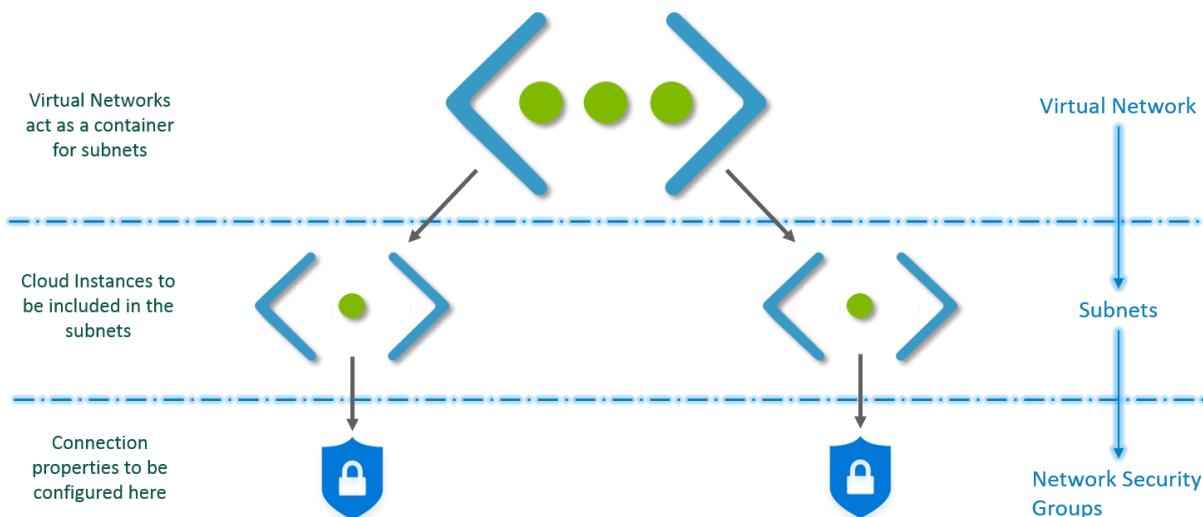
deployed to a specific subnet that is segmented using VNet address space. A subnet can further be divided into:

- **Private Subnet** – A network in which there is no internet access.
- **Public Subnet** – A network in which there is internet access.



**2. Network Security Groups (NSG):** Use to permit or deny traffic (inbound or outbound), via rules, to a subnet or network interface.

Any Azure virtual network can be placed into a security group where different inbound and outbound rules can be configured to allow or deny certain types of traffic. For each rule, you can specify source and destination, port, and protocol.



## Creating a Virtual Network in Azure

Step 1: Open the portal and search virtual networks

The screenshot shows the Microsoft Azure portal interface. The search bar at the top contains the text "virtual ne". Below the search bar, the "Azure services" section is visible, with a list of services including "Virtual networks" (which is highlighted with a red box), "Virtual network gateways", "Virtual networks (classic)", "Virtual machines", "Private DNS zones", "Virtual machine scale sets", "SQL virtual machines", and "Virtual clusters". To the right of the service list, there are links for "Marketplace", "See all", "Documentation", and "See all". Below the documentation links, there are three blue hyperlinks: "Azure Virtual Network | Microsoft Docs", "Create a virtual network - quickstart - Azure PowerShell ...", and "Connect virtual networks with VNet peering - tutorial ...".

Step 2: Add a new virtual network

The screenshot shows the "Virtual networks" blade in the Azure portal. At the top left, there is a breadcrumb navigation: "Home > Virtual networks". The main title is "Virtual networks" with a refresh icon. Below the title, it says "ValueMomentum Inc.". There is a toolbar with buttons for "Add" (highlighted with a red box), "Manage view", "Refresh", "Export to CSV", and "Open query". Below the toolbar, there are filter options: "Filter by name..." (with a red box around the "Add" button), "Subscription == all", and "Resource group == all". The text "Showing 1 to 1 of 1 records." is displayed. A table lists one record: "newgroup-vnet" (with a red box around the "new" part). The table has columns for "Name" and "Resource group".

Step 3: Enter the details of the virtual network

Project details

Subscription \* ⓘ Visual Studio Enterprise Subscription – MPN

Resource group \* ⓘ newgroup  
Create new

Instance details

Name \* newnetwork2

Region \* (US) East US 2

[Review + create](#) < Previous [Next : IP Addresses >](#) Download a template for automation

Step 4: Select IP Address and select the default Ip address and add new subnet

### Create virtual network

IPv4 address space

10.1.0.0/16 10.1.0.0 - 10.1.255.255 (65536 addresses)

Add IPv6 address space ⓘ

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

+ Add subnet Remove subnet

Subnet name  
 default Subnet address range 10.1.0.0/24

Step 5: Enter the subnet details Subnet name & Subnet address range

NOTE: Check the subnet address range should be in range 10.1.1.0 - 10.1.1.255

As the default subnet range is 10.1.0.0/24 it should be 10.1.1.0/24

## Add subnet

X

Subnet name \*

subnet1



Subnet address range \* ⓘ

10.1.1.0/24



10.1.1.0 - 10.1.1.255 (251 + 5 Azure reserved addresses)

### SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ

0 selected



Add

Cancel

Step 6: After creating the subnet Select Review + Create

+ Add subnet

Remove subnet

Subnet name

Subnet address range

default

10.1.0.0/24

subnet1

10.1.1.0/24

**Review + create**

< Previous

Next : Security >

D

 Validation passed

Basics IP Addresses Security Tags **Review + create**

**Basics**

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	newgroup
Name	newnetwork2
Region	East US 2

**IP addresses**

Address space	10.1.0.0/16
---------------	-------------

---

**Create** [< Previous](#) [Next >](#) [Download a template for automation](#)

## Connecting a Virtual Machine to a subnet

Step 1: Open Virtual Machines Section & Create a new VM

Username \* ⓘ

SSH public key source

Stored Keys

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ  None  Allow selected ports

**Review + create** [< Previous](#) **Next : Disks >**

Step 2: For the first Virtual Machine select the subnet to default

#### Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network \* ⓘ

Subnet \* ⓘ

Public IP ⓘ

NIC network security group ⓘ  Basic  None  Advanced

Public inbound ports \* ⓘ  None

## Create a virtual machine

 Validation passed

Basics Disks Networking Management Advanced Tags Review + create

#### PRODUCT DETAILS

Standard B1s  
by Microsoft  
[Terms of use](#) | [Privacy policy](#)

Subscription credits apply ⓘ  
**0.6874 INR/hr**  
[Pricing for other VM sizes](#)

#### TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same

Delete Cancel Redeploy Refresh

We'd love your feedback! →

## Your deployment is complete



Deployment name: CreateVm-Canonical.UbuntuServer-18.04-LTS-2... Start time: 2023-09-12T10:45:00Z  
Subscription: Visual Studio Enterprise Subscription – MPN Correlation ID: 00000000-0000-0000-0000-000000000000  
Resource group: newgroup

Deployment details [\(Download\)](#)

Next steps

[Setup auto-shutdown](#) Recommended

[Monitor VM health, performance and network dependencies](#) Recommended

[Run a script inside the virtual machine](#) Recommended

[Go to resource](#)

[Create another VM](#)

Connect Start Restart Stop Capture Delete Refresh Open in mobile

[Properties](#) [Monitoring](#) [Capabilities \(7\)](#) [Recommendations](#) [Tutorials](#)

### Virtual machine

Computer name: defaultsubnetvm  
Operating system: Linux (ubuntu 18.04)  
Publisher: Canonical  
Offer: UbuntuServer  
Plan: 18.04-LTS  
VM generation: V1  
Agent status: Ready

### Networking

Public IP address: 52.242.87.126  
Public IP address (IPv6): -  
Private IP address: 10.1.0.4  
Private IP address (IPv6): -  
Virtual network/subnet: newnetwork2/default  
DNS name: Configure

### Size

Step 2: Create Another VM, this time add it to the Subnet1

## Create a virtual machine

### Instance details

Virtual machine name \* ⓘ  ✓

Region \* ⓘ

Availability options ⓘ

Image \* ⓘ  [See all images](#)

Azure Spot instance ⓘ

Step 3: Now select the subnet as the new subnet created in the virtual network

## Create a virtual machine

When creating a virtual machine, a network interface will be created for you.

Virtual network \* ⓘ  [Create new](#)

Subnet \* ⓘ  [Manage subnet configuration](#)

Public IP ⓘ  [Create new](#)

NIC network security group ⓘ  Basic  None  Advanced

Public inbound ports \* ⓘ  Allow selected ports  None

**Review + create** < Previous Next : Management >

Validation passed

Basics Disks Networking Management Advanced Tags Review + create

**PRODUCT DETAILS**

Standard B1s by Microsoft

[Terms of use](#) | [Privacy policy](#)

**TERMS**

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same

[< Previous](#) [Next >](#) [Download a template for automation](#)

Delete Cancel Redeploy Refresh

We'd love your feedback! →

Deployment name: CreateVm-Canonical.UbuntuServer-18.04-LTS-2... Start time: 1/6/2021, 12:  
Subscription: Visual Studio Enterprise Subscription – MPN Correlation ID: a5027981  
Resource group: newgroup

✓ Deployment details [\(Download\)](#)

✗ Next steps

Setup auto-shutdown Recommended  
Monitor VM health, performance and network dependencies Recommended  
Run a script inside the virtual machine Recommended

Step 5: We can see that the Private IP address has changed in the resource.



Properties Monitoring Capabilities (7) Recommendations Tutorials

#### Virtual machine

Computer name	vmeinsubnet1
Operating system	Linux (ubuntu 18.04)
Publisher	Canonical
Offer	UbuntuServer
Plan	18.04-LTS
VM generation	V1
Agent status	Ready
Agent version	2.2.53
Host group	None

#### Networking

Public IP address	20.190.244.218
Public IP address (IPv6)	-
Private IP address	10.1.1.4
Private IP address (IPv6)	-
Virtual network/subnet	newnetwork2/subnet1
DNS name	Configure

#### Size

Size	Standard B1s
------	--------------

Step 6: Now go to the virtual networks and select the created virtual network

We can see that the Virtual machines are visible in the connected virtual network

Virtual networks

Showing 1 to 2 of 2 records.

Name	Resource group
newgroup-vnet	newgroup
newnetwork2	newgroup

newnetwork2

Virtual network

Overview

Subscription (change)  
Visual Studio Enterprise Subscription - MPN

Subscription ID  
2d1eaf32-f707-40db-a62c-2b7d297b42a6

Tags (change)  
Click here to add tags

Connected devices

Device	Type	IP Address	Subnet
defaultsubnetvm927	Network interface	10.1.0.4	default
vmeinsubnet1605	Network interface	10.1.1.4	subnet1

Search (Ctrl+ /) Refresh

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Address space Connected devices Subnets

Device	Type	IP Address	Subnet
defaultsubnetvm927	Network interface	10.1.0.4	default
mainsubnet1605	Network interface	10.1.1.4	subnet1

## Creating a Network Security Group

**Step 1:** Go to portal and search for network security groups

Microsoft Azure Search: network sec

Azure services Services See all

- + Create a resource
- Recent resources
- Name
  - newnetwork2
  - [newgroup]

Marketplace See all

- Network security group
- McAfee® Network Security Manager
- FireEye Network Security
- Stormshield Network Security for Cloud - XL

Documentation See all

- Azure network security groups overview | Microsoft Docs
- Network security concepts and requirements in Azure ...
- What is Azure Security Center? | Microsoft Docs
- Computer forensics Chain of Custody in Azure - Azure ...

Resource Groups

**Step 2:** Add a new network security group

ValueMomentum Inc.

Add Edit columns Refresh Try preview Assign tags

New Network security group

Subscriptions: All 2 selected – Don't see a subscription? Open Directory + Subscription settings

Filter by name... All subscriptions All resource groups All locations

4 items

Name	Resource group	Location
defaultsubnetvm-nsg	newgroup	East US 2
newvm1-nsg	newgroup	Central India

**Step 3:** Enter the details in the Basics tab

Project details

Subscription \* Visual Studio Enterprise Subscription – MPN

Resource group \* newgroup

Instance details

Name \* publicsecgroup

Region \* (US) East US 2

[Review + create](#) < Previous [Next : Tags >](#) Download a template for automation

Step 4: After entering the details click Review + Create

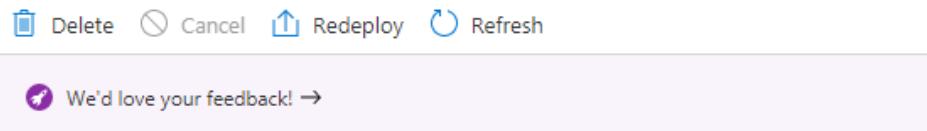
Validation passed

Basics Tags [Review + create](#)

**Basics**

Subscription	Visual Studio Enterprise Subscription – MPN
Resource group	newgroup
Region	East US 2
name	publicsecgroup

[Create](#) < Previous [Next >](#) Download a template for automation



Step 6: Go to the Inbound security rules in the Settings and Add an inbound rule

The screenshot shows the "publicsecgroup | Inbound security rules" page. It features a search bar, an "Add" button, and a refresh button. The main area displays a table of existing rules with columns for Priority, Name, Port, and Protocol. A new rule is being added with a priority of 65000, name "AllowVnetInBound", port Any, and protocol Any. The "Inbound security rules" link in the left sidebar is highlighted with a red box.

Step 7: Set the Destination port range to 80 and Add rule

 Add inbound security rule

publicsecgroup

 Basic

\*

Destination \* ⓘ

Any

Destination port ranges \* ⓘ

80

Protocol \*

Any    TCP    UDP    ICMP

Action \*

Allow    Deny

Priority \* ⓘ

100



Step 8: Now the inbound rule is created in the Settings

publicsecgroup | Inbound security rules

Network security group

Search (Ctrl+ /)    Add    Default rules    Refresh

Priority	Name	Port	Protocol	Source	Destination
100	Port_8080	80	Any	Any	Any
65000	AllowVmInBound	Any	Any	VirtualNetwork	VirtualNetwork
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any
65500	DenyAllInBound	Any	Any	Any	Any

## Creating a File Share and Service Mounting in Azure

Step 1: Create a Virtual machine and open the Linux Machine

## Create a Storage Account in the portal

The screenshot shows the Microsoft Azure portal interface for creating a new storage account. The top navigation bar includes the Microsoft Azure logo, a search bar, and various navigation icons.

In the "Azure services" section, the "Storage accounts" icon is highlighted with a black border. Other options like "Create a resource", "Network security groups", "Virtual networks", "Virtual machines", "Availability sets", and "Public IP addresses" are also listed.

The "Recent resources" section shows a dropdown menu for selecting a resource group, with "newgroup" selected and "Create new" option available.

The "Instance details" section contains the following fields:

- Storage account name \***: newsaistorage234
- Location \***: (US) East US 2
- Performance**: Standard (radio button selected)

At the bottom of this section are "Review + create" and "Next : Networking >" buttons. The "Networking" tab is currently selected in the breadcrumb navigation.

The "Networking" tab page includes the following sections:

- Network connectivity**: Describes connectivity options via public IP addresses or service endpoints, or privately via a private endpoint.
- Connectivity method \***: A radio button group where "Public endpoint (all networks)" is selected. Other options include "Public endpoint (selected networks)" and "Private endpoint". A note states: "All networks will be able to access this storage account." with a link to "Learn more about connectivity methods".

At the bottom of this page are "Review + create", "< Previous", and "Next : Data protection >" buttons.

## Create storage account

Basics Networking Data protection Advanced Tags Review + create

### Recovery

- Turn on point-in-time restore for containers

Use point-in-time restore to restore one or more containers to an earlier state. If point-in-time restore is enabled, then versioning, change feed, and blob soft delete must also be enabled. [Learn more](#)

- Turn on soft delete for blobs

Soft delete enables you to recover blobs that were previously marked for deletion, including blobs that were overwritten. [Learn more](#)

- Turn on soft delete for containers

Soft delete enables you to recover containers that were previously marked for deletion. [Learn more](#)

**Review + create**

< Previous

Next : Advanced >

## Create storage account

Validation passed

Basics Networking Data protection Advanced Tags **Review + create**

### Basics

Subscription Visual Studio Enterprise Subscription – MPN

Resource group newgroup

Location East US 2

Storage account name newsaistorage234

Deployment model Resource manager

Account kind StorageV2 (general purpose v2)

**Create**

< Previous

Next >

[Download a template for automation](#)

Delete Cancel Redeploy Refresh

We'd love your feedback! →

✓ Your deployment is complete

Deployment Deployment name: Microsoft.StorageAccount-20210106132828 Start time: 1/6/2021, 1:31:29 AM  
Subscription: Visual Studio Enterprise Subscription – MPN Correlation ID: fbee5e06-6c3d-43f3-833a-  
Resource group: newgroup

Deployment details [\(Download\)](#)

Next steps

[Go to resource](#)

Step 2: After successful deployment go to resource and select File shares

newsaistorage234 Storage account

Search (Ctrl+ /) Overview Activity log Tags (change) Click here to add tags

Containers Scalable, cost-effective storage for unstructured data Learn more

File shares Serverless SMB and NFS file shares Learn more Tables Tabular data storage Learn more

Queues Effectively scale apps according to traffic Learn more

Step 3: Create a new File share

File shares

+ File share Refresh

File share New file share

Active Directory: Not configured Soft delete: Disabled Share capacity: 5 TiB

Search file shares by prefix (case-sensitive)

Name	Modified	Tier	Quota
You don't have any file shares yet. Click '+ File share' to get started.			

Step 4: Enter the name and set Quota to 1025 and Tiers to Hot

### New file share

Name **\***  
sharedtovm

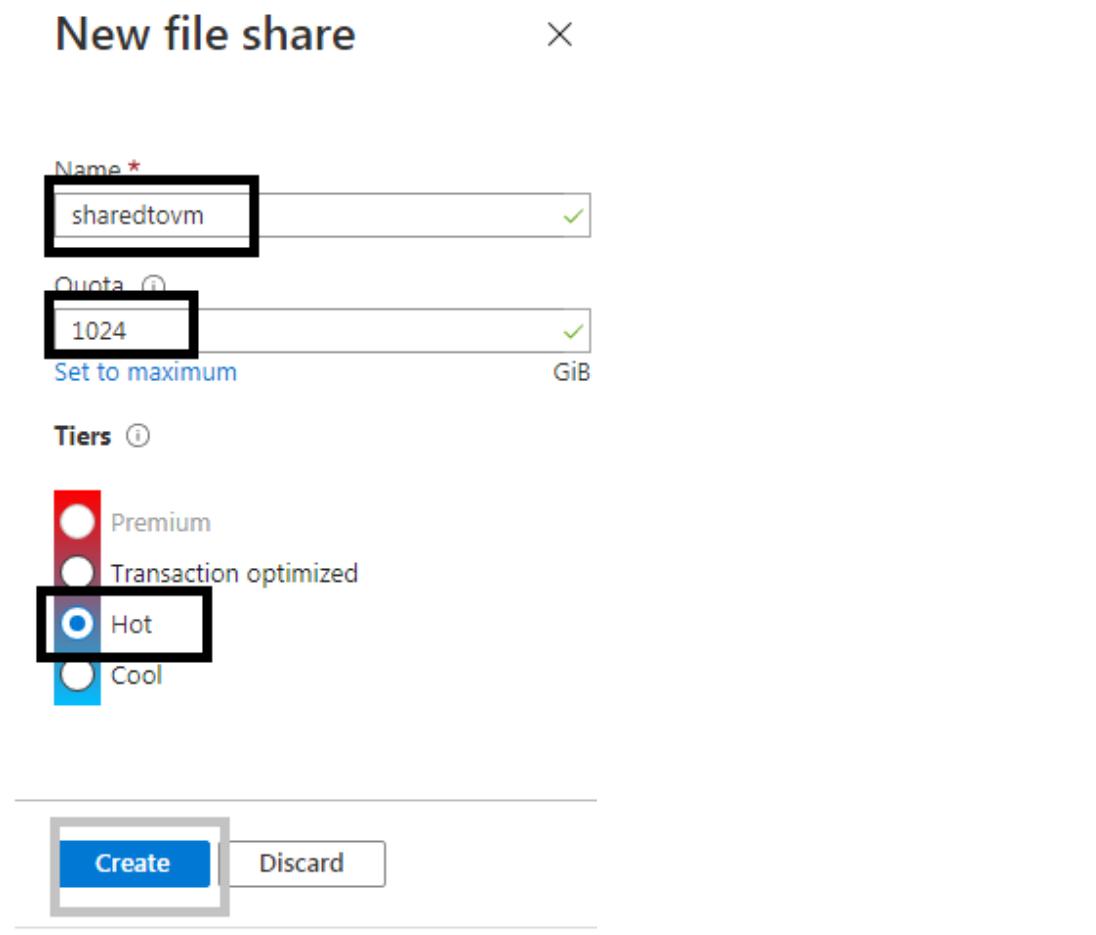
Quota **(i)**  
1024 GiB  
[Set to maximum](#)

Tiers **(i)**

- Premium
- Transaction optimized
- Hot**
- Cool

---

**Create** Discard



Step 5: Now connect the file share to the Linux machine

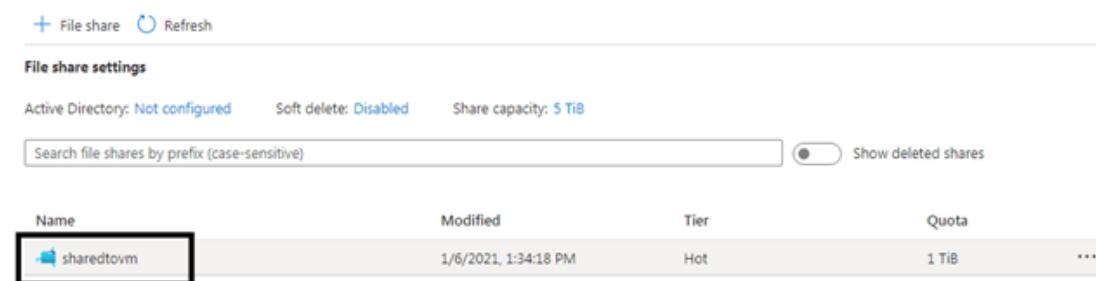
+ File share ⏪ Refresh

**File share settings**

Active Directory: Not configured   Soft delete: [Disabled](#)   Share capacity: **5 TiB**

Search file shares by prefix (case-sensitive)  Show deleted shares

Name	Modified	Tier	Quota	...
sharedtovm	1/6/2021, 1:34:18 PM	Hot	1 TiB	...



The screenshot shows the Azure Storage Explorer interface. At the top, there's a toolbar with icons for 'Connect', 'Upload', 'Add directory', 'Refresh', 'Delete share', 'Change tier', and 'Edit quota'. Below the toolbar is a search bar with the placeholder 'Search files by prefix'. A table below the search bar has columns for 'Name', 'Type', and 'Size'. A message 'No files found.' is displayed. The entire interface is framed by a black border.

Step 6: Copy the code and paste in the Linux machine already created

The screenshot shows a 'Connect' dialog box. At the top left is the title 'Connect' and at the top right is a close button 'X'. Below the title is the share name 'sharedtovm'. There's a yellow banner with the text 'Requirements: Click here to learn more...' and a number '1'. Below the banner are three tabs: 'Windows', 'Linux' (which is selected and highlighted with a blue underline), and 'macOS'. Underneath the tabs is a 'Mount point' input field containing 'sharedtovm'. To the right of the mount point field is a 'Copy' button with a 'Copy to clipboard' option and a '2' icon. Below the dialog is a note about mounting Azure file shares outside the region.

To connect to this file share from a Linux computer, run this command:

```
sudo mkdir /mnt/sharedtovm
if [ ! -d "/etc/smbcredentials" ]; then
    sudo mkdir /etc/smbcredentials
fi
if [ ! -f "/etc/smbcredentials/newsaistorage234.cred" ]; then
    sudo bash -c 'echo "username=newsaistorage234" >>
    /etc/smbcredentials/newsaistorage234.cred'
```

*Copy*

Copy to clipboard



2

In order to mount an Azure file share outside of the Azure region it is hosted in, such as on-premises or in a different Azure region, the OS must support the encryption functionality of SMB 3.0.

```
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
azureuser@vmeinsubnet1:~$
```

```
azureuser@vmeinsubnet1:~$ editvm -o vers=3.0,credentials=/etc/smbcredentials/newsaistorage234.cred,dir_mod=e=0777,file_mode=0777,serverinoazureuser@vmeinsubnet1:~$ if [ ! -d "/etc/smbcredentials" ]; then  
> sudo mkdir /etc/smbcredentials  
> fi  
azureuser@vmeinsubnet1:~$ if [ ! -f "/etc/smbcredentials/newsaistorage234.cred" ]; then  
>     sudo bash -c 'echo "username=newsaistorage234" >> /etc/smbcredentials/newsaistorage234.cred'  
>     sudo bash -c 'echo "password=5qAWPOUA6HA2vmUTvsA0ku99Pnovk91IV6Dta/HuJp4ck+imqtqr3vRgz6WBhgC0wDg5/FPTM1oS7X4NGjyI4w==" >> /etc/smbcredentials/newsaistorage234.cred'  
> fi  
azureuser@vmeinsubnet1:~$ sudo chmod 600 /etc/smbcredentials/newsaistorage234.cred  
azureuser@vmeinsubnet1:~$ sudo bash -c 'echo "//newsaistorage234.file.core.windows.net/sharedtovm /mnt/sharedtovm cifs nofail,vers=3.0,credentials=/etc/smbcredentials/newsaistorage234.cred,dir_mode=0777,file_mode=0777,serverino" >> /etc fstab'  
azureuser@vmeinsubnet1:~$ sudo mount -t cifs //newsaistorage234.file.core.windows.net/sharedtovm /mnt/sharedtovm -o vers=3.0,credentials=/etc/smbcredentials/newsaistorage234.cred,dir_mode=0777,file_mode=0777,serverino  
azureuser@vmeinsubnet1:~$
```

Step 7: Execute the `ls /mnt` command and we can see that a warning with user directory is shown

```
azureuser@vmeinsubnet1:~$ sudo mount -t cifs //newsaistorage234.file.core.windows.net/sharedtovm /mnt/sharedtovm -o vers=3.0,credentials=/etc/smbcredentials/newsaistorage234.cred,dir_mode=0777,file_mode=0777,serverino  
azureuser@vmeinsubnet1:~$ ls /mnt  
DATALOSS_WARNING_README.txt  lost+found  sharedtovm  
azureuser@vmeinsubnet1:~$
```

```
cp /mnt/DATALOSS_WARNING_README.txt /mnt/sharedtovm/
```

```
azureuser@vmeinsubnet1:~$ ls  
azureuser@vmeinsubnet1:~$ ls /mnt/sharedtovm/  
azureuser@vmeinsubnet1:~$ cp /mnt/DATALOSS_WARNING_README.txt /mnt/sharedtovm/  
azureuser@vmeinsubnet1:~$
```

Go to the File Share page, the file copied from VM will exist in the Storage

A screenshot of a web-based file sharing interface. At the top, there are navigation links: Connect, Upload, Add directory, Refresh, Delete share, Change tier, and Edit quota. Below this is a search bar labeled 'Search files by prefix'. A table lists a single file: 'DATALOSS\_WARNING\_README.txt', which is a File type with a size of 639 B.

A screenshot of the same file sharing interface, but with the 'Upload' button highlighted with a black box. The rest of the interface is identical to the first screenshot.

A screenshot of a Windows 'Open' file dialog. The file 'VirtualNetworks\_Subnets' is selected in the list. The dialog includes a 'File name:' field containing 'VirtualNetworks\_Subnets', a 'Type:' dropdown set to 'All Files', and two buttons: 'Open' and 'Cancel'. The 'Open' button is highlighted with a black box.



Go to the Linux VM in Putty session, run ls command there on the /mnt/sharedtovm folder

```
azureuser@vmeinsubnet1:~$ ls /mnt/sharedtovm/
DATALOSS_WARNING_README.txt VirtualNetworks_Subnets.docx
azureuser@vmeinsubnet1:~$ ls -l /mnt/sharedtovm/
total 849
-r-xr-xr-x 1 root root    639 Jan  6 08:10 DATALOSS_WARNING_README.txt
-rwxrwxrwx 1 root root 868286 Jan  6 08:14 VirtualNetworks_Subnets.docx
azureuser@vmeinsubnet1:~$
```

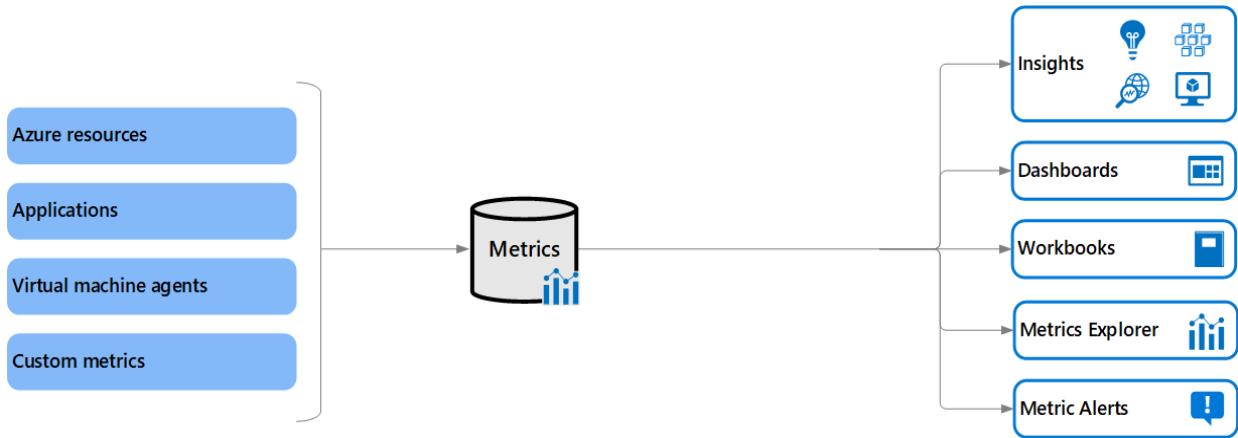
## AZURE Metrics

Azure Monitor Metrics is a feature of Azure Monitor that collects numeric data from monitored resources into a time series database. Metrics are numerical values that are collected at regular intervals and describe some aspect of a system at a particular time. Metrics in Azure Monitor are lightweight and capable of supporting near real-time scenarios making them particularly useful for alerting and fast detection of issues

The following table lists the different ways that you can use Metrics in Azure Monitor.

WHAT CAN YOU DO WITH AZURE MONITOR METRICS?

<b>Analyze</b>	Use <a href="#">metrics explorer</a> to analyze collected metrics on a chart and compare metrics from different resources.
<b>Alert</b>	Configure a <a href="#">metric alert rule</a> that sends a notification or takes <a href="#">automated action</a> when the metric value crosses a threshold.
<b>Visualize</b>	Pin a chart from metrics explorer to an <a href="#">Azure dashboard</a> . Create a <a href="#">workbook</a> to combine with multiple sets of data in an interactive report. Export the results of a query to <a href="#">Grafana</a> to leverage its dashboarding and combine with other data sources.
<b>Automate</b>	Use <a href="#">Autoscale</a> to increase or decrease resources based on a metric value crossing a threshold.
<b>Retrieve</b>	Access metric values from a command line using <a href="#">PowerShell cmdlets</a> Access metric values from custom application using <a href="#">REST API</a> . Access metric values from a command line using <a href="#">CLI</a> .
<b>Export</b>	<a href="#">Route Metrics to Logs</a> to analyze data in Azure Monitor Metrics together with data in Azure Monitor Logs and to store metric values for longer than 93 days. Stream Metrics to an <a href="#">Event Hub</a> to route them to external systems.
<b>Archive</b>	<a href="#">Archive</a> the performance or health history of your resource for compliance, auditing, or offline reporting purposes.



## Data collection

There are three fundamental sources of metrics collected by Azure Monitor. Once these metrics are collected in the Azure Monitor metric database, they can be evaluated together regardless of their source.

**Azure resources.** Platform metrics are created by Azure resources and give you visibility into their health and performance. Each type of resource creates a [distinct set of metrics](#) without any configuration required. Platform metrics are collected from Azure resources at one-minute frequency unless specified otherwise in the metric's definition.

**Applications.** Metrics are created by Application Insights for your monitored applications and help you detect performance issues and track trends in how your application is being used. This includes such values as *Server response time* and *Browser exceptions*.

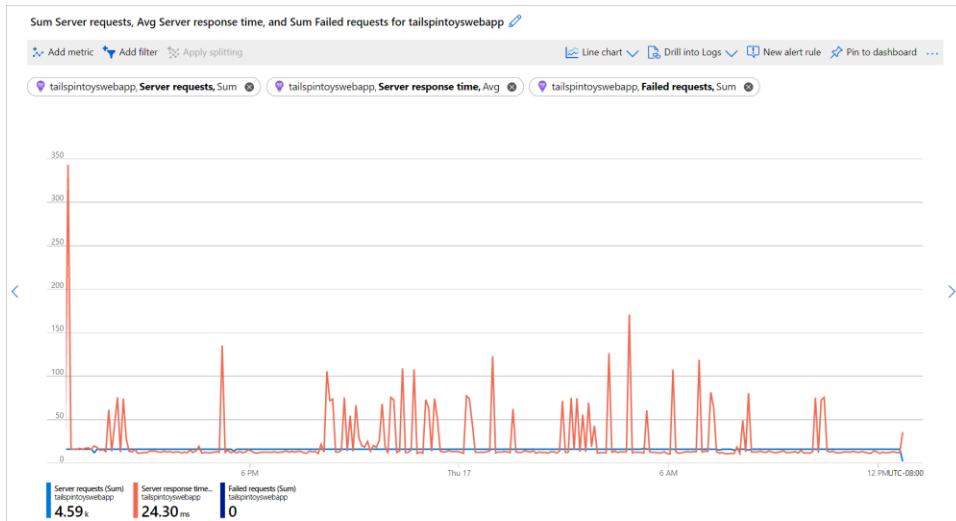
**Virtual machine agents.** Metrics are collected from the guest operating system of a virtual machine. Enable guest OS metrics for Windows virtual machines with [Windows Diagnostic Extension \(WAD\)](#) and for Linux virtual machines with [InfluxData Telegraf Agent](#).

**Custom metrics.** You can define metrics in addition to the standard metrics that are automatically available. You can [define custom metrics in your application](#) that's monitored by Application Insights or create custom metrics for an Azure service using the [custom metrics API](#).

- ★ See [What is monitored by Azure Monitor?](#) for a complete list of data sources that can send data to Azure Monitor Metrics.

## Metrics explorer

Use [Metrics Explorer](#) to interactively analyze the data in your metric database and chart the values of multiple metrics over time. You can pin the charts to a dashboard to view them with other visualizations. You can also retrieve metrics by using the [Azure monitoring REST API](#).



- ★ See [Getting started with Azure Monitor metrics explorer](#) to get started using metrics explorer.

## Data structure

Data collected by Azure Monitor Metrics is stored in a time-series database which is optimized for analyzing time-stamped data. Each set of metric values is a time series with the following properties:

- ★ The time the value was collected
- ★ The resource the value is associated with
- ★ A namespace that acts like a category for the metric
- ★ A metric name
- ★ The value itself
- ★ Some metrics may have multiple dimensions as described in [Multi-dimensional metrics](#). Custom metrics can have up to 10 dimensions.

## Multi-dimensional metrics

One of the challenges to metric data is that it often has limited information to provide context for collected values. Azure Monitor addresses this challenge with multi-dimensional metrics. Dimensions of a metric are name-value pairs that carry additional data to describe the metric value. For example, a metric *Available disk space* could have

a dimension called *Drive* with values *C*; *D*; which would allow viewing either available disk space across all drives or for each drive individually.

The example below illustrates two datasets for a hypothetical metric called *Network Throughput*. The first dataset has no dimensions. The second dataset shows the values with two dimensions, *IP Address* and *Direction*:

## Network Throughput

### NETWORK THROUGHPUT

Timestamp	Metric Value
8/9/2017 8:14	1,331.8 Kbps
8/9/2017 8:15	1,141.4 Kbps
8/9/2017 8:16	1,110.2 Kbps

This non-dimensional metric can only answer a basic question like "what was my network throughput at a given time?"

## Network Throughput + two dimensions ("IP" and "Direction")

### NETWORK THROUGHPUT + TWO DIMENSIONS ("IP" AND "DIRECTION")

Timestamp	Dimension "IP"	Dimension "Direction"	Metric Value
8/9/2017 8:14	IP="192.168.5.2"	Direction="Send"	646.5 Kbps
8/9/2017 8:14	IP="192.168.5.2"	Direction="Receive"	420.1 Kbps
8/9/2017 8:14	IP="10.24.2.15"	Direction="Send"	150.0 Kbps
8/9/2017 8:14	IP="10.24.2.15"	Direction="Receive"	115.2 Kbps
8/9/2017 8:15	IP="192.168.5.2"	Direction="Send"	515.2 Kbps
8/9/2017 8:15	IP="192.168.5.2"	Direction="Receive"	371.1 Kbps
8/9/2017 8:15	IP="10.24.2.15"	Direction="Send"	155.0 Kbps
8/9/2017 8:15	IP="10.24.2.15"	Direction="Receive"	100.1 Kbps

This metric can answer questions such as "what was the network throughput for each IP address?", and "how much data was sent versus received?" Multi-dimensional metrics carry additional analytical and diagnostic value compared to non-dimensional metrics.

## Retention of Metrics

For most resources in Azure, metrics are stored for 93 days. There are some exceptions:

## Guest OS metrics

- ★ **Classic guest OS metrics.** These are performance counters collected by the [Windows Diagnostic Extension \(WAD\)](#) or the [Linux Diagnostic Extension \(LAD\)](#) and routed to an Azure storage account. Retention for these metrics is 14 days.

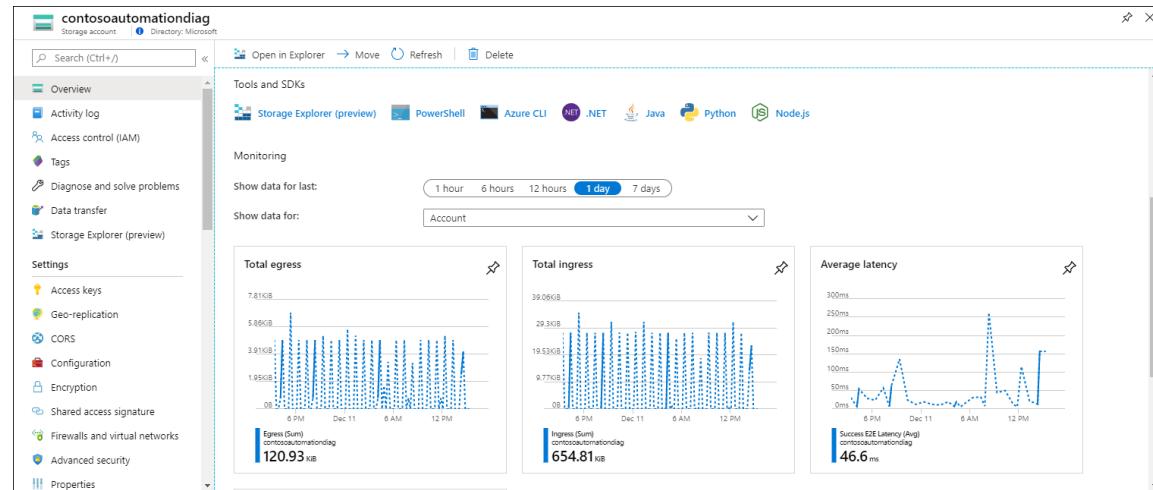
- ★ **Guest OS metrics sent to Azure Monitor Metrics.** These are performance counters collected by the [Windows Diagnostic Extension \(WAD\)](#) and sent to the [Azure Monitor data sink](#), or via the [InfluxData Telegraf Agent](#) on Linux machines. Retention for these metrics is 93 days.
- ★ **Guest OS metrics collected by Log Analytics agent.** These are performance counters collected by the Log Analytics agent and sent to a Log Analytics workspace. Retention for these metrics is 31 days, and can be extended up to 2 years.

## Application Insights log-based metrics.

- ★ Behind the scene, [log-based metrics](#) translate into log queries. Their retention matches the retention of events in underlying logs. For Application Insights resources, logs are stored for 90 days.

## Creating metrics in Azure

- ★ Locate an Azure resource in your subscription.
- ★ Go to the **Overview** page and note if there's any performance data displayed. This data will be provided by Azure Monitor. The example below is the **Overview** page for an Azure storage account, and you can see that there are multiple metrics displayed.



- ★ You can click on any of the graphs to open the data in metrics explorer which is described below.

## View the Activity log

The Activity log provides insight into the operations on each Azure resource in the subscription. This will include such information as when a resource is created or modified, when a job is started, or when a particular operation occurs.

- ★ At the top of the menu for your resource, select **Activity log**.
- ★ The current filter is set to events related to your resource. If you don't see any events, try changing the **Timespan** to increase the time scope.

Operation name	Status	Time	Time stamp	Subscription	Event initiated by	Resource group
> 'audit' Policy action.	Succeeded	9 h ago	Wed Dec 11...	Contoso IT - demo	Microsoft Azure Policy Ins...	ContosoAzureHQ
> 'deployIfNotExists' Policy action.	Succeeded	1 d ago	Tue Dec 10 ...	Contoso IT - demo	Microsoft Azure Policy Ins...	ContosoAzureHQ
> 'deployIfNotExists' Policy action.	Succeeded	2 d ago	Mon Dec 09 ...	Contoso IT - demo	Microsoft Azure Policy Ins...	ContosoAzureHQ
> 'audit' Policy action.	Succeeded	3 d ago	Sun Dec 08 ...	Contoso IT - demo	Microsoft Azure Policy Ins...	ContosoAzureHQ
> 'deployIfNotExists' Policy action.	Succeeded	4 d ago	Sat Dec 07 ...	Contoso IT - demo	Microsoft Azure Policy Ins...	ContosoAzureHQ
> Export an existing database.	Failed	5 d ago	Fri Dec 06 2...	Contoso IT - demo	Imolkova@microsoft.com	ContosoAzureHQ
> Export an existing database.	Failed	5 d ago	Fri Dec 06 2...	Contoso IT - demo	Imolkova@microsoft.com	ContosoAzureHQ
> Get Database Top Queries query text	Succeeded	5 d ago	Fri Dec 06 2...	Contoso IT - demo	fanyv@microsoft.com	contosoazurehq

- ★ If you want to see events from other resources in your subscription, either change criteria in the filter or even remove filter properties.

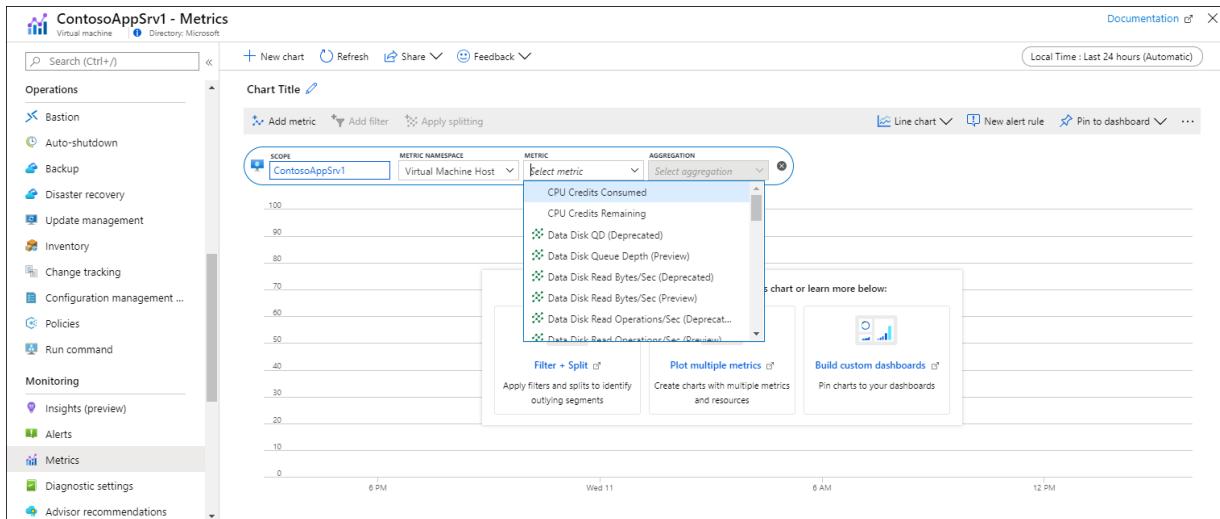
Operation name	Status	Time	Time stamp	Subscription	Event initiated by	Resource group	Resource type	Resource
> 'deployIfNotExists' Policy action.	Succeeded	30 min ago	Wed Dec 11...	Contoso IT - demo	Microsoft Azure Policy Ins...	contosoazurehq	Microsoft.Resources/...	virtualMachines/Con...
> Validate Deployment	Succeeded	31 min ago	Wed Dec 11...	Contoso IT - demo	d800f36368b4d93a5ba...	contosoazurehq	Microsoft.Resources/...	deployments/Policy...
> 'audit' Policy action.	Succeeded	38 min ago	Wed Dec 11...	Contoso IT - demo	Microsoft Azure Policy Ins...	contosoazurehq	Microsoft.Resources/...	virtualMachines/Con...
> Create or update Container Group	Failed	47 min ago	Wed Dec 11...	Contoso IT - demo	ContosoSH360KubCluster...	MC_ContosoSH360K...	Microsoft.Container...	containerGroups/def...
> Create or update Container Group	Failed	47 min ago	Wed Dec 11...	Contoso IT - demo	ContosoSH360KubCluster...	MC_ContosoSH360K...	Microsoft.Container...	containerGroups/def...
> Create or update Container Group	Failed	48 min ago	Wed Dec 11...	Contoso IT - demo	ContosoSH360KubCluster...	MC_ContosoSH360K...	Microsoft.Container...	containerGroups/def...
> Create or update Container Group	Failed	49 min ago	Wed Dec 11...	Contoso IT - demo	ContosoSH360KubCluster...	MC_ContosoSH360K...	Microsoft.Container...	containerGroups/def...
> List Storage Account Keys	Succeeded	49 min ago	Wed Dec 11...	Contoso IT - demo	Hyper-V Recovery Manager	contosorecoveryvaul...	Microsoft.Storage/st...	storageAccounts/qi...

## View metrics

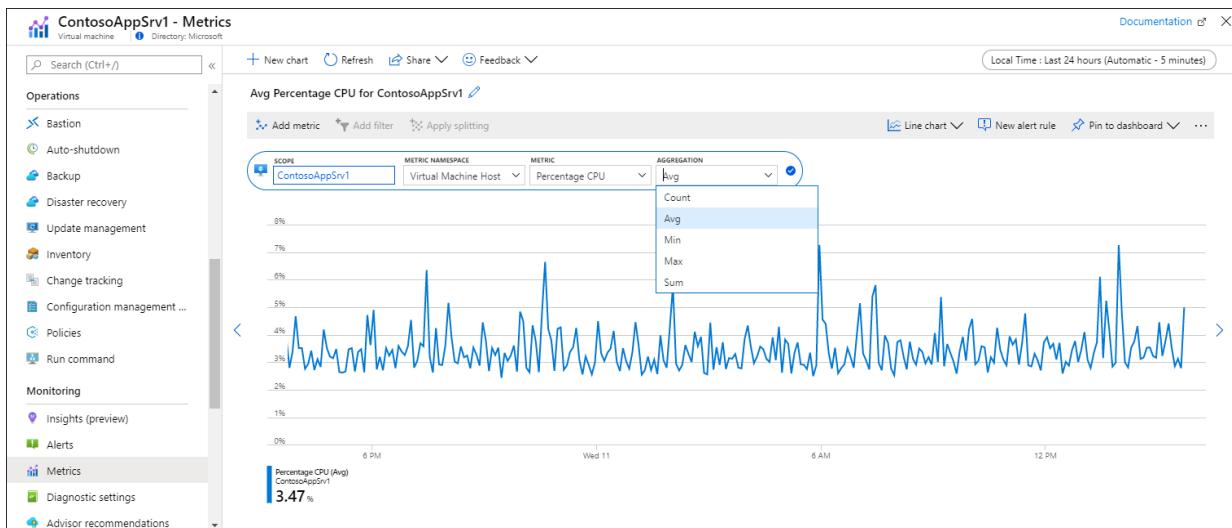
Metrics are numerical values that describe some aspect of your resource at a particular time. Azure Monitor automatically collects platform metrics at one minute intervals from all Azure resources. You can view these metrics using metrics explorer.

- ★ Under the **Monitoring** section of your resource's menu, select **Metrics**. This opens metrics explorer with the scope set to your resource.

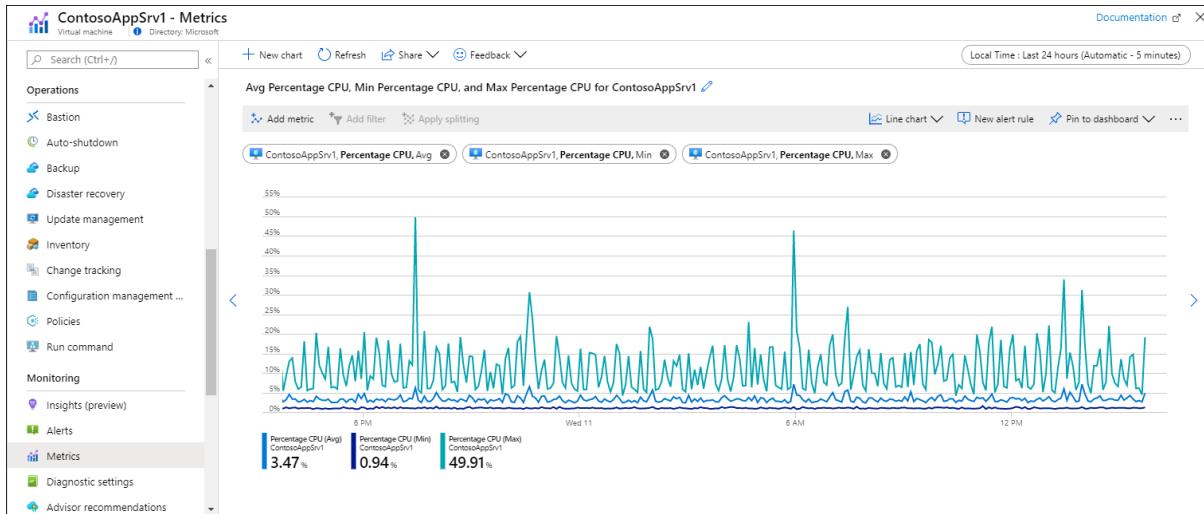
- ★ Click **Add metric** to add a metric to the chart.



- ★ Select a **Metric** from the dropdown list and then an **Aggregation**. This defines how the collected values will be sampled over each time interval.



- ★ Click **Add metric** to add additional metric and aggregation combinations to the chart.



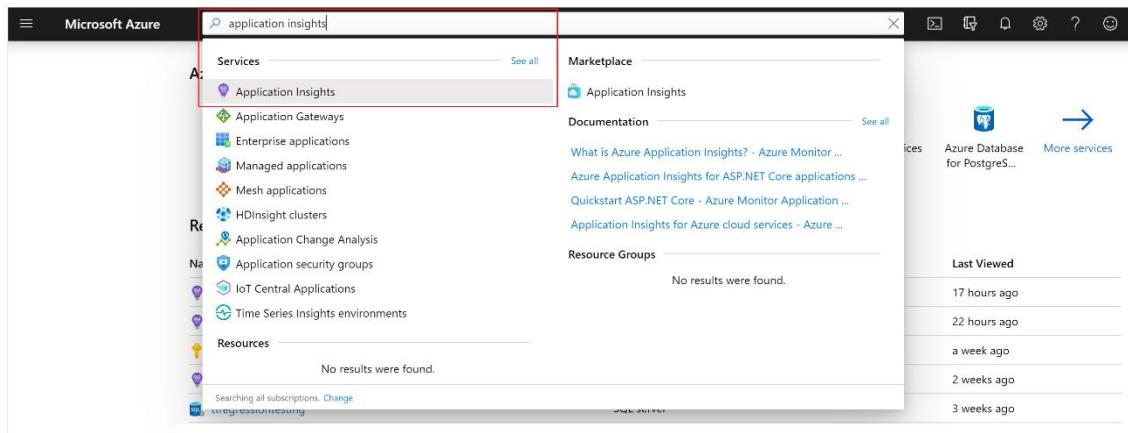
## App Insights in Azure

Application Insights includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app.

With your app connected to Applications Insights, you can collect information to help you drive better business decisions and improve the quality of your apps.

## Creating App Insights

- ★ Sign in to the [Azure portal](#).
- ★ Search for Application Insights:



- ★ Create an Application Insights resource:

**Subscriptions: Pay-As-You-Go**

Filter by name...	All resource groups	All locations	All tags	No
<input type="checkbox"/> Name ↑	Type ↑	Resource group ↑↓	Location ↑↓	Subscription ↑↓
<input type="checkbox"/> BudgetTracker	Application Insights	AppInsights	West US 2	Pay-As-You-Go

- ★ Enter the appropriate values and select **Review + create**. For more details, read [Create an Application Insights resource](#).

**Basics** Tags Review + create

Create an Application Insights resource to monitor your live web application. With Application Insights, you have full observability into your application across all components and dependencies of your complex distributed architecture. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and Java EE, hosted on-premises, hybrid, or any public cloud. [Learn More](#)

**PROJECT DETAILS**

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

**Subscription \*** ⓘ Pay-As-You-Go

**Resource Group \*** ⓘ Demo  
Create new

**INSTANCE DETAILS**

**Name \*** ⓘ Kudos-AI

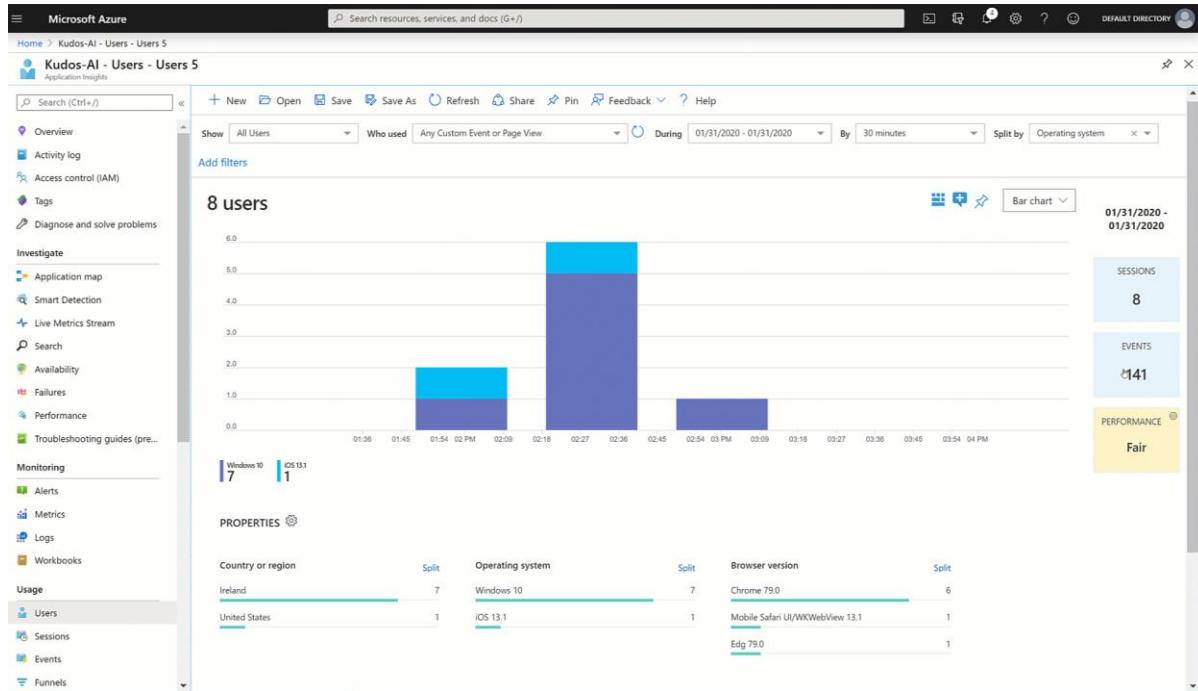
**Region \*** ⓘ (US) West US 2

**Review + create** << Previous Next : Tags >

- ★ After the Application Insights instance is created, you'll see the instance overview. Copy the **Instrumentation Key**. You'll need this key to configure your app.

- ★ Select one of the user sessions to drill into specific details. You can see information such as the session length and the screens visited:

- ★ Select the **Events** view in left navigation pane under the **Usage** section. You can see a summary of all the screens viewed across all app sessions:



## Analyze data in Application Insights

You can now begin to analyze the data you sent using the [Trace](#) function from your app in Application Insights.

- ★ Sign in to the [Azure portal](#) and open the Application Insights resource you created [earlier](#):

Name	Type	Last Viewed
Kudos-AI	Application Insights	14 minutes ago
Pay-As-You-Go	Subscription	a week ago

- ★ Select **Logs** under **Monitoring** from the left navigation pane:

★ Enter the following query and select **Run**. The feedback received from your app is returned:

★ KustoCopy

★ `traces| where message == "App Feedback" | order by timestamp`

timestamp [UTC]	message	severityLevel	itemType	customDimensions
2020-03-10T11:51:55.366Z	App Feedback	1	trace	{"UserName": "[REDACTED]", "ms-appSessionId": "512f6c64-3165-43b8-bcaa-2b04cd2febbb", "ms-appName": "Kudos"}
				FeedbackValue 1
				Screen Select Colleague Screen
				UserEmail [REDACTED]
				UserName [REDACTED]
				ms-appId 4aee54f2-f8fb-4a82-bbd3-aa34554380f0
				ms-appName Kudos
				ms-appSessionId 512f6c64-3165-43b8-bcaa-2b04cd2febbb

★ Select a row in the results and expand the *customDimensions* field.

- ★ The values for **Screen**, **UserName**, **UserEmail**, and **FeedbackValue** for the **OnSelect** event of the smile or frown icon in your component have been recorded.

There are also some additional values recorded for each event sent to Application Insights, such as the **appId**, **appName**, and **appSessionId**.

Drag a column header and drop it here to group by that column					
	timestamp [UTC]	message	severityLevel	itemType	customDimensions
▼	3/10/2020, 11:51:55.366 AM	App Feedback	1	trace	{"UserName": "████████", "ms-appSessionId": "512f6c64-3165-43b8-bcaa-2b04cd2feb8b", "ms-appName": "Kudos", "FeedbackValue": 1, "Screen": "Select Colleague Screen", "UserEmail": "████████", "UserName": "████████", "ms-appId": "4aee54f2-f8fb-4a82-bbd3-aa34554380f0", "ms-appName": "Kudos", "ms-appSessionId": "512f6c64-3165-43b8-bcaa-2b04cd2feb8b"}
	timestamp [UTC]	2020-03-10T11:51:55.366Z			
	message	App Feedback			
	severityLevel	1			
	itemType	trace			
▼	customDimensions	{"UserName": "████████", "ms-appSessionId": "512f6c64-3165-43b8-bcaa-2b04cd2feb8b", "ms-appName": "Kudos", "FeedbackValue": 1, "Screen": "Select Colleague Screen", "UserEmail": "████████", "UserName": "████████", "ms-appId": "4aee54f2-f8fb-4a82-bbd3-aa34554380f0", "ms-appName": "Kudos", "ms-appSessionId": "512f6c64-3165-43b8-bcaa-2b04cd2feb8b"}			
	FeedbackValue	1			
	Screen	Select Colleague Screen			
	UserEmail	████████			
	UserName	████████			
	ms-appId	4aee54f2-f8fb-4a82-bbd3-aa34554380f0			
	ms-appName	Kudos			
	ms-appSessionId	512f6c64-3165-43b8-bcaa-2b04cd2feb8b			

- ★ With the following example query, you can extend the properties of the JSON custom dimensions and project the columns in the results view.

#### ★ KustoCopy

```
★ traces | extend customdims = parse_json(customDimensions) | where message == "App Feedback" | project timestamp, message, AppName = customdims['ms-appName'], AppId = customdims['ms-appId'], FeedbackFrom = customdims.UserEmail, Screen = customdims.Screen, FeedbackValue = customdims.FeedbackValue | order by timestamp desc
```

Traces								
<pre>   extend customdims = parse_json(customDimensions)   where message == "App_Feedback"   project timestamp     , message     , AppName = customdims['ms-appName']     , AppId = customdims['ms-appId']     , FeedbackFrom = customdims.UserEmail     , Screen = customdims.Screen     , FeedbackValue = customdims.FeedbackValue   order by timestamp desc </pre>								
Completed. Showing results from the last 24 hours.								
<span style="float: right;">🕒 00:00:00.697</span> <span style="float: right;">31 records</span> <span style="float: right;">Display time (UTC+00:00) ▾</span> <span style="float: right;">Copy request ID</span>								
<span style="float: left;">Table</span> <span style="float: left;">Chart</span> <span style="float: left;">Columns ▾</span>								
Drag a column header and drop it here to group by that column								
timestamp [UTC]	message	AppName	AppId	FeedbackFrom	Screen	FeedbackValue		
3/10/2020, 11:51:33.284 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:33.110 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:52.942 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:52.766 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:52.590 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:52.380 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:51.646 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:50.640 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:42.932 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			
3/10/2020, 11:51:42.779 AM	App Feedback	Kudos	4aee54f2-f8fb-4a82-bbd3-aa34554380f0	Select Colleague Screen	-1			

## Shell Scripting in Azure

In few words, a shell script is nothing more and nothing less than a program that is executed step by step by a shell, which is another program that provides an interface layer between the Linux kernel and the end user

### Creating and Running a shell script and automating in azure VM

`sudo apt-get update`

`sudo apt-get install -y git apache2`

```
azureuser@universalForwarder:~  
To see these additional updates run: apt list --upgradable  
  
New release '20.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Mon Jan 18 07:28:31 2021 from 122.178.34.244  
azureuser@universalForwarder:~$ sudo apt-get update  
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease  
Hit:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease  
Reading package lists... Done  
azureuser@universalForwarder:~$ sudo apt-get install -y git apache2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
apache2 is already the newest version (2.4.29-1lubuntu4.14).  
git is already the newest version (1:2.17.1-1lubuntu0.7).  
git set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.  
azureuser@universalForwarder:~$
```

```
git clone https://github.com/adrianbautista/html-css-template.git
```

```
sudo cp -r html-css-template/* /var/www/html/
```

```
curl localhost
```

```
azureuser@universalForwarder:~$ ls html-css-template  
LICENSE.md  css      images    js  
README.md  favicon.ico index.html template.html  
azureuser@universalForwarder:~$ sudo cp -r html-css-template/* /var/www/html/  
azureuser@universalForwarder:~$ curl localhost  
<!DOCTYPE html>  
  <head>  
    <meta charset="utf-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
  
    <title></title>  
    <meta name="description" content="">  
  
    <link rel="shortcut icon" href="favicon.ico">  
  
    <link rel="stylesheet" href="css/normalize.css">  
    <link rel="stylesheet" href="css/html5-boilerplate.css">  
    <link rel="stylesheet" href="css/main.css">  
  </head>  
  
<body>
```

nano automate.sh

```
azureuser@universalForwarder: ~
GNU nano 2.9.3          automate.sh          Modified
sudo apt-get update
sudo apt-get install -y git apache2
git clone https://github.com/adrianbautista/html-css-template.git
sudo cp -r html-css-template/* /var/www/html/
File Name to Write: automate.sh
^G Get Help      M-D DOS Format     M-A Append      M-B Backup File
^C Cancel       M-M Mac Format     M-P Prepend     ^T To Files
```

sudo apt-get update

sudo apt-get install -y git apache2

git clone <https://github.com/adrianbautista/html-css-template.git>

sudo cp -r html-css-template/\* /var/www/html/

Ctrl+o + Enter

Ctrl+x

```
azureuser@universalForwarder: ~
.bash_logout
.bashrc
.cache/
.gnupg/
.profile
.rnd
.splunk/
.ssh/
.sudo_as_admin_successful
html-css-template/
splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz
azureuser@universalForwarder:~$ sudo rm -r html-css-template/
azureuser@universalForwarder:~$ sudo rm -r /var/www/
azureuser@universalForwarder:~$ ls /var
backups  crash  local  log  opt  snap  tmp
cache   lib    lock   mail  run  spool
azureuser@universalForwarder:~$ vi
azureuser@universalForwarder:~$ nano automate.sh
azureuser@universalForwarder:~$ 
azureuser@universalForwarder:~$ ls
automate.sh  splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz
azureuser@universalForwarder:~$
```

**sudo sh automate.sh**

```
azureuser@universalForwarder: ~
azureuser@universalForwarder:~$ nano automate.sh
azureuser@universalForwarder:~$ 
azureuser@universalForwarder:~$ ls
automate.sh  splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz
azureuser@universalForwarder:~$ sudo sh automate.sh
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1lubuntu0.7).
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 14 not upgraded.
Need to get 0 B/95.1 kB of archives.
After this operation, 536 kB of additional disk space will be used.
Selecting previously unselected package apache2.
```

**curl localhost**

```
azureuser@universalForwarder: ~
Resolving deltas: 100% (2749/2749), done.
azureuser@universalForwarder:~$ curl localhost
<!DOCTYPE html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title></title>
<meta name="description" content="">

<link rel="shortcut icon" href="favicon.ico">

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="css/html5-boilerplate.css">
<link rel="stylesheet" href="css/main.css">
</head>

<body>
<p>Hello Front Row Viewers!</p>
```

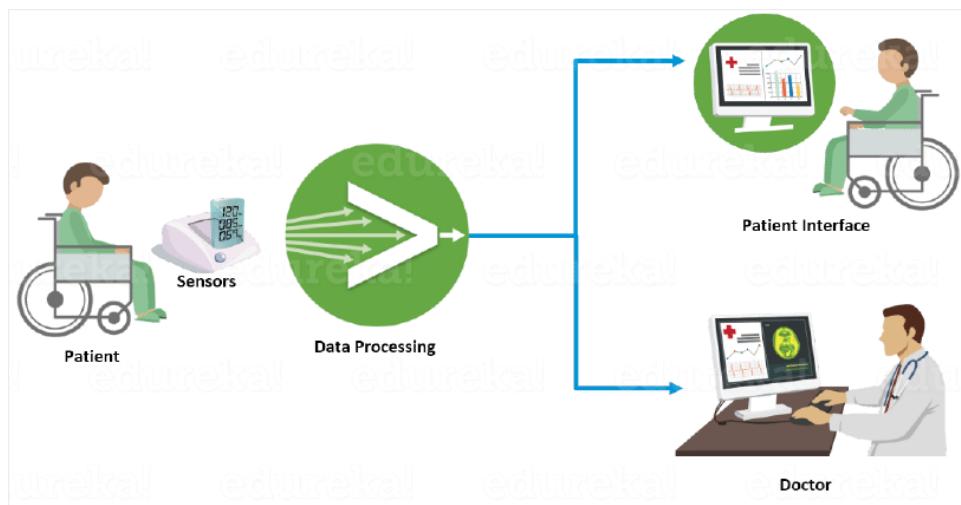
---

## **2. SPLUNK**

### **2.1 What is Splunk?**

Splunk is a software platform to search, analyze and visualize the machine-generated data gathered from the websites, applications, sensors, devices etc. which make up your IT infrastructure and business.

You must be aware of the exponential growth in machine data over the last decade. It was partly because of the growing number of machines in the IT infrastructure and partly because of the increased use of IoT devices. This machine data has a lot of valuable information that can drive efficiency, productivity and visibility for the business. Splunk was founded in 2003 for one purpose: To Make Sense of Machine Generated Log Data and since then the demand for Splunk skill is increasing.



## **2.2What is Splunk used for?**

If you were a Sys-admin trying to figure out what went wrong in your system's hardware and you stumble upon logs Would you be able to locate in which step your hardware failed you? There is a remote chance that you might be able to figure it out, but even that is only after spending hours in understanding what each word means. To tell you in a nutshell, machine data is:

- Complex to understand
- In an unstructured format
- Not suitable for making analysis / visualization

This is where a tool like Splunk comes in handy. You can feed the machine data to Splunk, which will do the dirty work (data processing) for you. Once it processes and extracts the relevant data, you will be able to easily locate where and what the problems were.



Splunk started off this way, but it became more prominent with the onset of Big Data. Since Splunk can store and process large amounts of data, data analysts like myself started feeding big data to Splunk for analysis. Dashboards meant for visualization was a revelation and within no time Splunk was extensively used in the big data domain for analytics.

### 2.3 SPLUNK INSTALLATION IN UBUNTU MACHINE:

This is a step-by-step process of installing the Splunk in a virtual machine in Microsoft Azure.

[splunk>](#) Products ▾ Solutions ▾ Why Splunk? ▾ Resources ▾ Support ▾ [Free Splunk](#)

license to continue using the expanded functionality designed for enterprise-scale deployments.

Choose Your Installation Package

Platform	Distribution	File Type	Size	Action
Windows	Windows	.msi	367.89 MB	<a href="#">Download Now</a>
Linux	2.6+, 3.x+, or 4.x+ kernel Linux distributions	.deb	367.89 MB	<a href="#">Download Now</a>
		.tgz	484.76 MB	<a href="#">Download Now</a>
		.rpm	485.41 MB	<a href="#">Download Now</a>

**Ctrl+C -> To copy the wget link after clicking "here" link**

GET STARTED

## You're Downloading Splunk Enterprise 8.1.1 for Linux

Your download should have started. No? [Try this URL](#).

Choose additional platforms [here](#).

**USEFUL TOOLS**

- Download via [Command Line \(wget\)](#)
- Download via [File Transfer Protocol \(FTP\)](#)
- Get direct download links [here](#)

We've got ampersands in the URL and they're all escaped and ready for wget. This URL won't work in your browser. Click [here](#) to select the entire command.

```
wget -O splunk-8.1.1-08187535c166-Linux-x86_64.tgz https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.1&product=splunk&filename=splunk-8.1.1-08187535c166-Linux-x86_64.tgz&wget=true
```

```
wget -O splunk-8.1.1-08187535c166-Linux-x86_64.tgz
'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86\_64&platform=linux&version=8.1.1&product=splunk&filename=splunk-8.1.1-08187535c166-Linux-x86\_64.tgz&wget=true'
```

Create a Linux Machine in Azure with following specifications:

OS: Ubuntu 18.04

Size: B2s or Higher with 2 CPU Cores & 4 GB RAM or Higher

simpleLinuxVM

Virtual machine

Search (Ctrl+ /)

Connect Start Restart Stop Capture Delete Refresh Open in mobile

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource group (change)  
NEWGROUP

Status  
Running

Location  
Central India

Subscription (change)  
Visual Studio Enterprise Subscription – MPN

Subscription ID  
2d1ef32-f707-40db-a62c-2b7d297b42a6

Operating system  
Linux (Ubuntu 18.04)

Size  
Standard B2s (2 vcpus, 4 GiB memory)

Public IP address  
52.140.100.148

Virtual network/subnet  
newgroupvnet9477/default

DNS name  
simplelinuxvm-mdo7fe7ncuwju.centralindia.cloudapp.azure.com

```
azureuser@splunkvm: ~
System information as of Thu Jan  7 06:48:19 UTC 2021
System load:  0.6          Processes:           130
Usage of /:   4.5% of 28.90GB  Users logged in:    0
Memory usage: 5%           IP address for eth0: 10.0.0.6
Swap usage:   0%
0 packages can be updated.
0 updates are security updates.

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

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

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@splunkvm:~$
```

**Paste the wget command copied from the Splunk download page in the ubuntu VM by right clicking on the Putty**

```
azureuser@splunkvm: ~$ wget -O splunk-8.1.1-08187535c166-Linux-x86_64.tgz 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.1&product=splunk&filename=splunk-8.1.1-08187535c166-Linux-x86_64.tgz&wget=true'
--2021-01-07 06:49:59-- https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.1&product=splunk&filename=splunk-8.1.1-08187535c166-Linux-x86_64.tgz&wget=true
Resolving www.splunk.com (www.splunk.com)... 96.17.150.163, 96.17.150.155
Connecting to www.splunk.com (www.splunk.com)|96.17.150.163|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://download.splunk.com/products/splunk/releases/8.1.1/linux/splunk-8.1.1-08187535c166-Linux-x86_64.tgz [following]
--2021-01-07 06:49:59-- https://download.splunk.com/products/splunk/releases/8.1.1/linux/splunk-8.1.1-08187535c166-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com)... 52.84.8.96, 52.84.8.12, 52.84.8.62, ...
Connecting to download.splunk.com (download.splunk.com)|52.84.8.96|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 508304047 (485M) [application/x-gzip]
Saving to: 'splunk-8.1.1-08187535c166-Linux-x86_64.tgz'

k-8.1.1-08187535c16 49%[=====>] 238.01M 79.2MB/s eta 3s
```

```
azureuser@splunkvm: ~$ ls
splunk-8.1.1-08187535c166-Linux-x86_64.tgz
azureuser@splunkvm: ~$
```

sudo cp splunk-8.1.1-08187535c166-Linux-x86\_64.tgz /opt/

ls /opt

```
azureuser@splunkvm:~$ sudo cp splunk-8.1.1-08187535c166-Linux-x86_64.tgz /opt/
azureuser@splunkvm:~$ ls /opt/
splunk-8.1.1-08187535c166-Linux-x86_64.tgz
azureuser@splunkvm:~$
```

```
cd /opt/
```

```
ls
```

```
sudo tar -xvf splunk-8.1.1-08187535c166-Linux-x86_64.tgz
```

```
ls
```

```
cd splunk/bin
```

```
ls
```

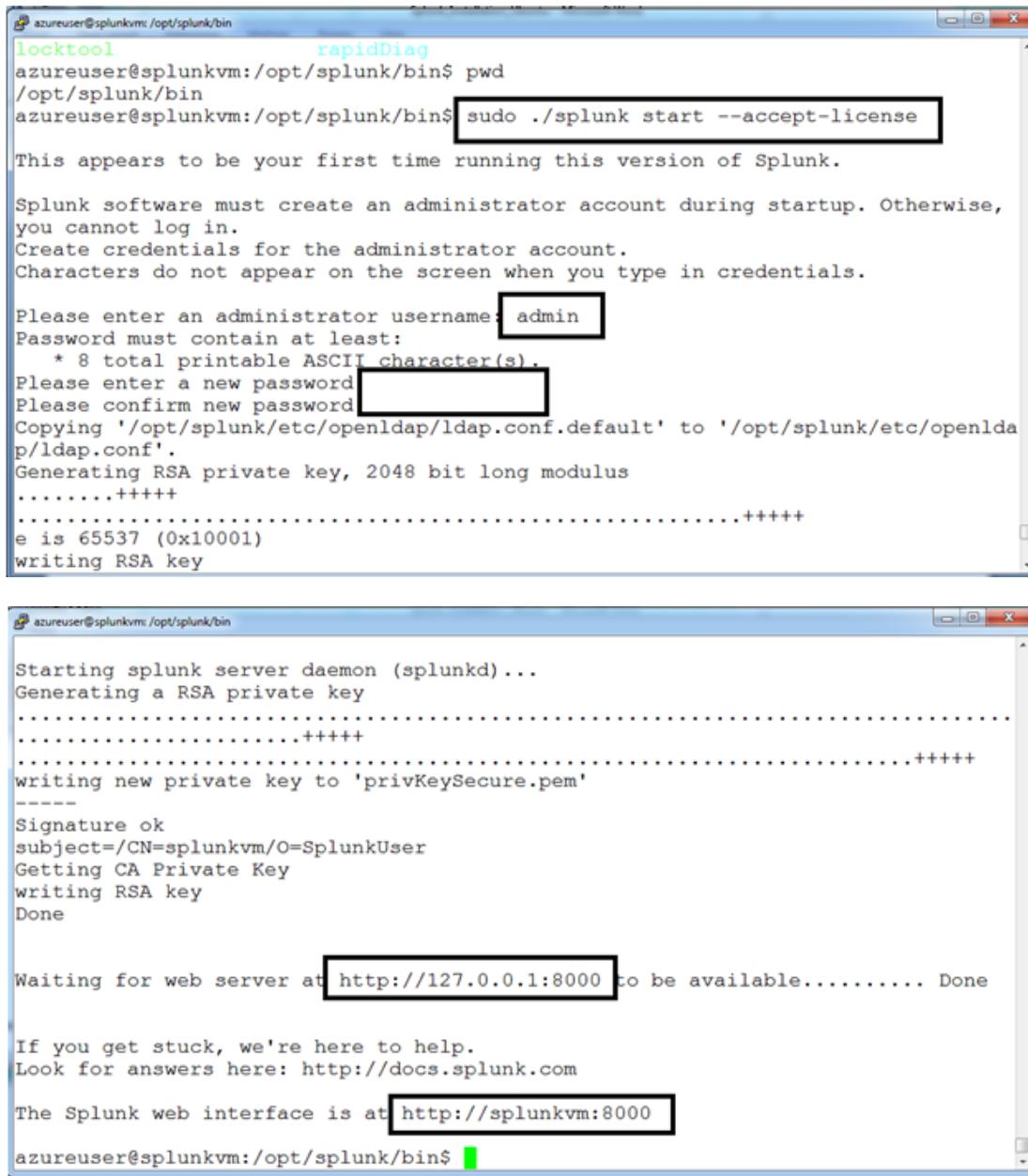


Enter the Username and Password of your own

Example: Username: Admin

Password: Admin@1234 and Confirm the password.

**NOTE:** *The password would not be visible to you, but the user should be typing the password and confirming it.*



```
azureuser@splunkvm:/opt/splunk/bin
locktool      rapidDiag
azureuser@splunkvm:/opt/splunk/bin$ pwd
/opt/splunk/bin
azureuser@splunkvm:/opt/splunk/bin$ sudo ./splunk start --accept-license
This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise,
you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: admin
Password must contain at least:
    * 8 total printable ASCII character(s).
Please enter a new password
Please confirm new password
Copying '/opt/splunk/etc/openldap/ldap.conf.default' to '/opt/splunk/etc/openldap/ldap.conf'.
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
writing RSA key

Starting splunk server daemon (splunkd)...
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'privKeySecure.pem'
-----
Signature ok
subject=/CN=splunkvm/O=SplunkUser
Getting CA Private Key
writing RSA key
Done

Waiting for web server at http://127.0.0.1:8000 to be available..... Done

If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://splunkvm:8000
azureuser@splunkvm:/opt/splunk/bin$
```

sudo ./splunk enable boot-start -user azureuser

- ❖ Open the Networking Tab in the Virtual Machine and Click on the "**Add Inbound Port Rule**" and Open Port 8000 to be accessible from Internet on this VM:

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Networking

Connect

Disk

Size

Security

Advisor recommendations

Attach network interface Detach network interface

splunkvm360

IP configuration ipconfig1 (Primary)

Network Interface: splunkvm360 Effective security rules Topology

Virtual network/subnet: newgroup-vnet/default NIC Public IP: 40.80.92.104 NIC Private IP: 10.0.0.6 Accelerated networking: Disabled

Inbound port rules Outbound port rules Application security groups Load balancing

Network security group splunkvm-nsg (attached to network interface: splunkvm360) Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowwanNatLoadBalancerInbound	Any	Any	Any	Any	Allow

## Add inbound security rule

splunkvm-nsg

Basic

Source \* ⓘ

Any

Source port ranges \* ⓘ

\*

Destination \* ⓘ

Any

Destination port ranges \* ⓘ

8000

Protocol \*

Any TCP UDP ICMP

Action \*

Allow Deny

Add

Copy the IP address of the VM

The screenshot shows the Azure portal interface for a virtual machine named 'splunkvm'. The 'Essentials' section displays various configuration details:

- Resource group: newgroup
- Status: Running
- Location: Central India
- Subscription: Visual Studio Enterprise Subscription - MPN
- Subscription ID: 2d1eaf32-f707-40db-a62c-2b7d297b42a6
- Tags: (change)

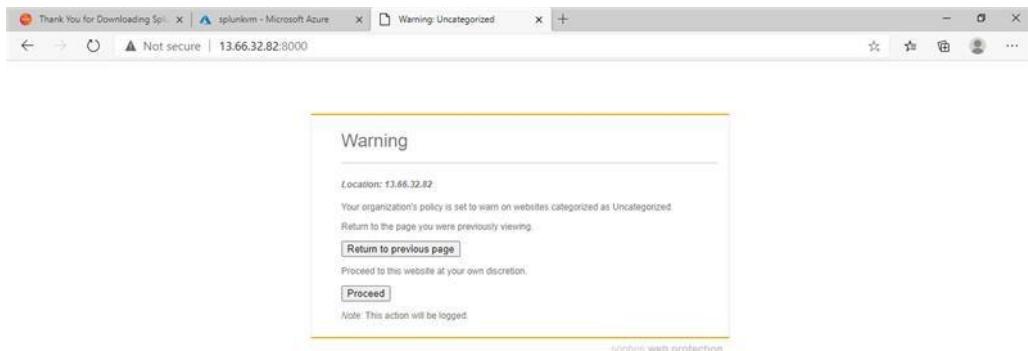
On the right side, specific details are listed:

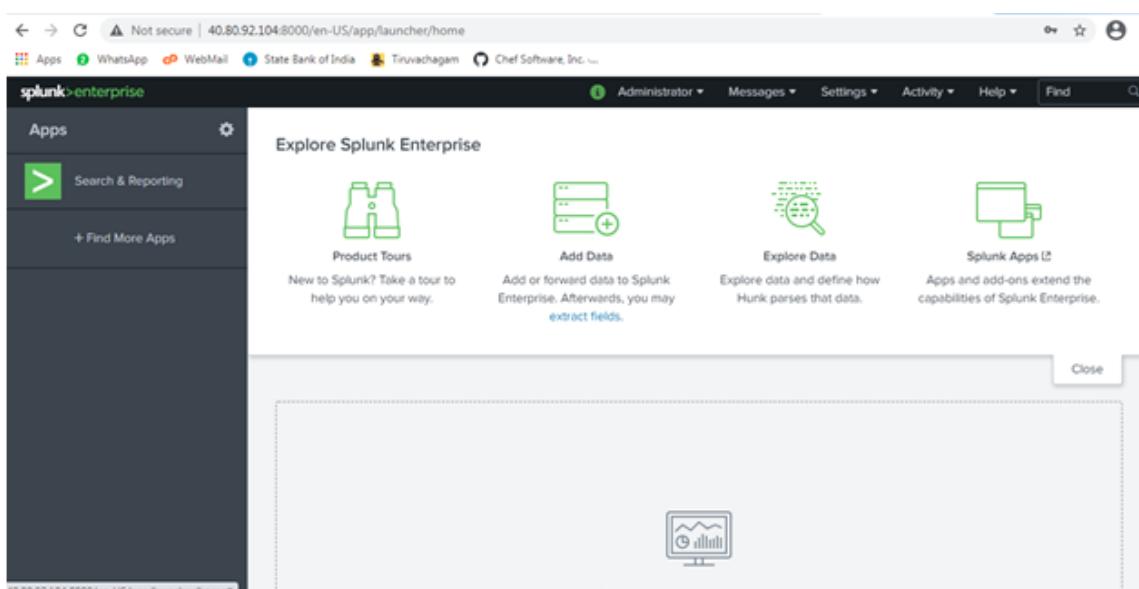
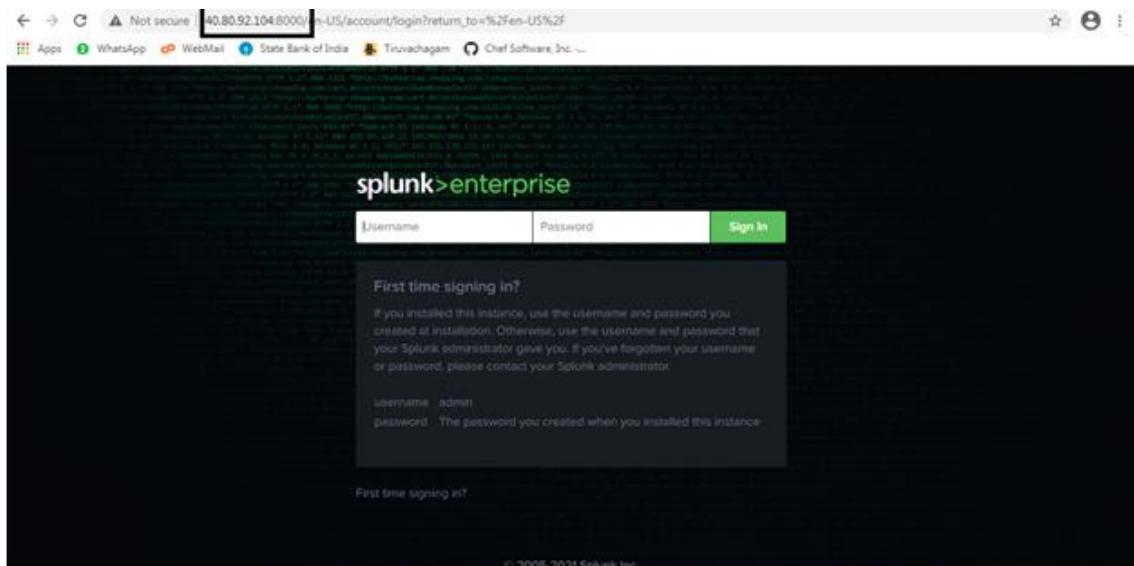
- Operating system: Linux (ubuntu 18.04)
- Size: Standard\_B1ms (1 vCPU, 1.5 GB memory)
- IP address: 40.80.92.104 (highlighted in a red box)
- Virtual network/subnet: newgroup-vnet/default
- DNS name: Configure

Open the browser and paste the IP Address with :8000

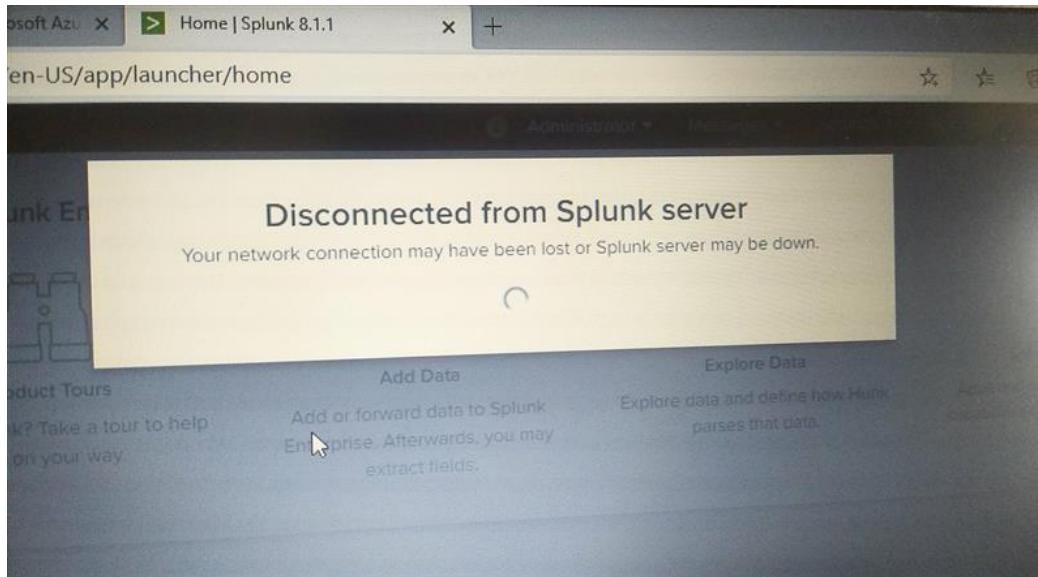
40.80.92.104:8000 (**Paste your IP Address with :8000**)

- ❖ Some of the users may be getting this kind of message on the entering the IP Address



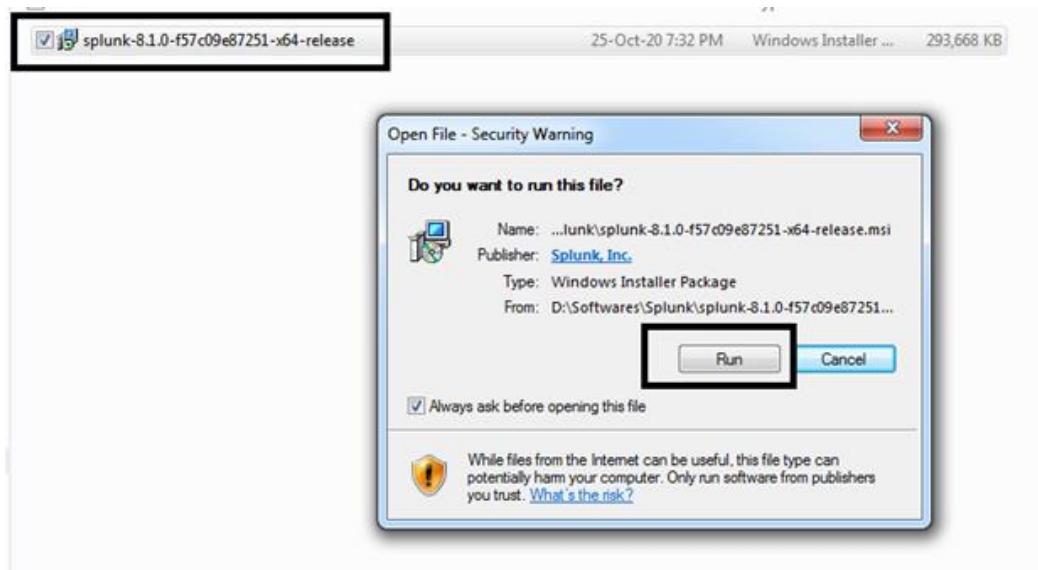


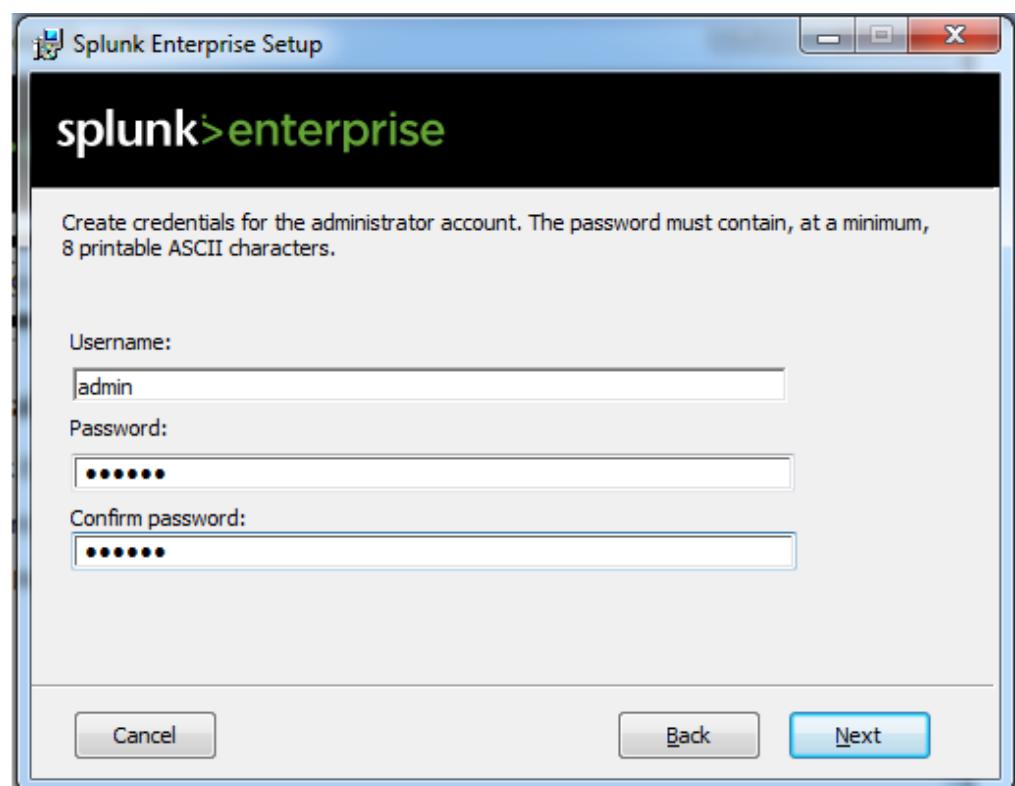
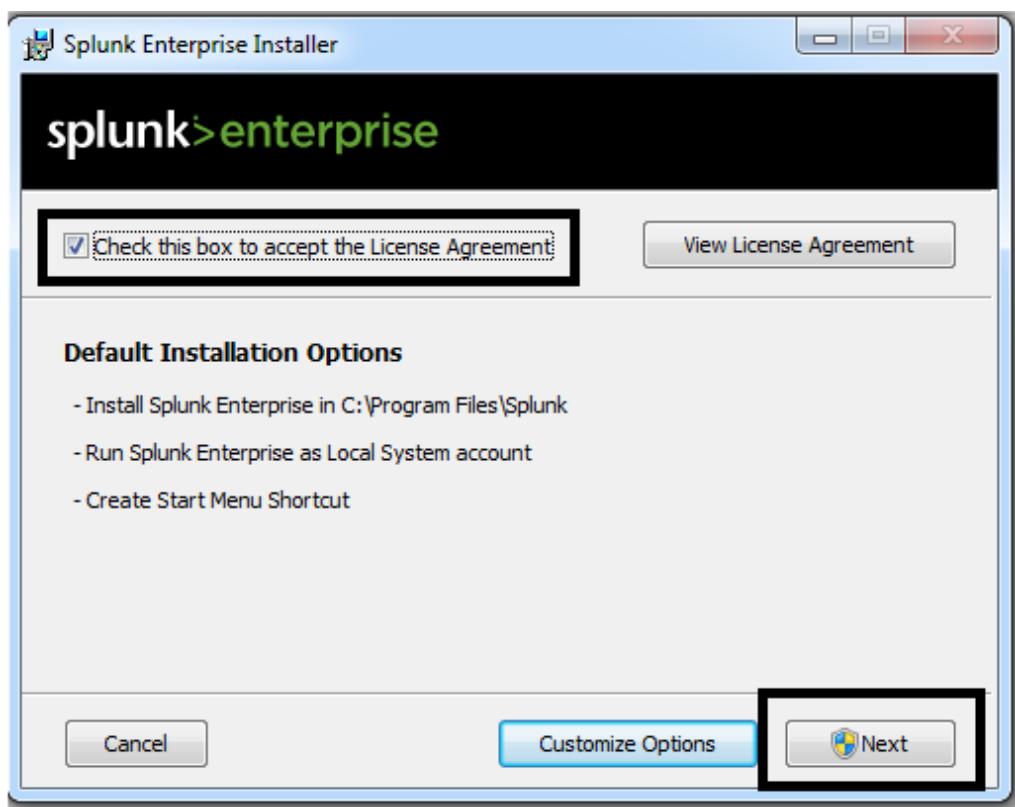
**REFRESH THE PAGE WHENEVER THIS MESSAGE APPEARS ON YOUR SCREEN.**

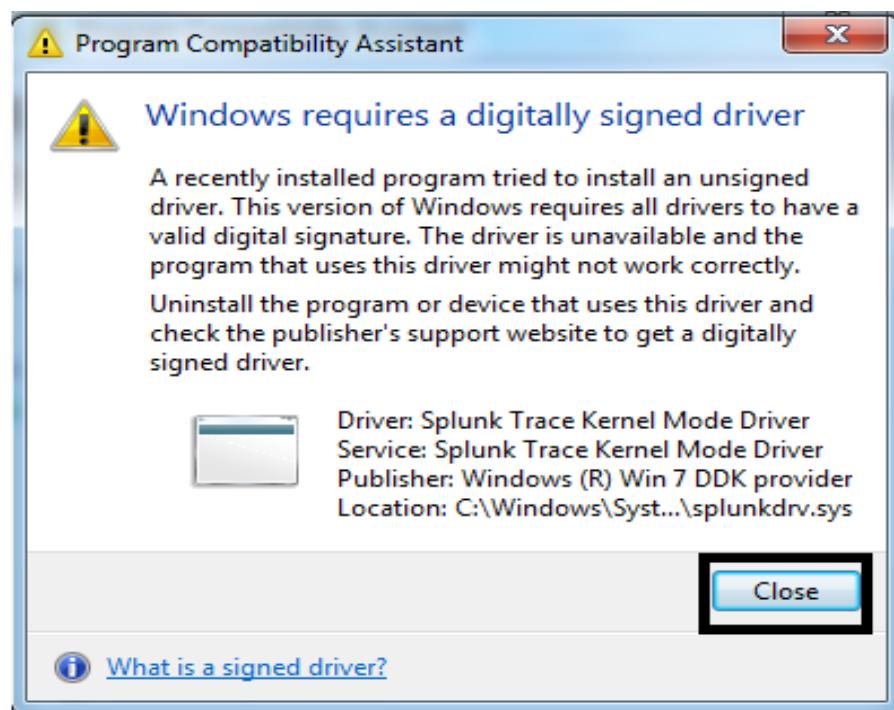
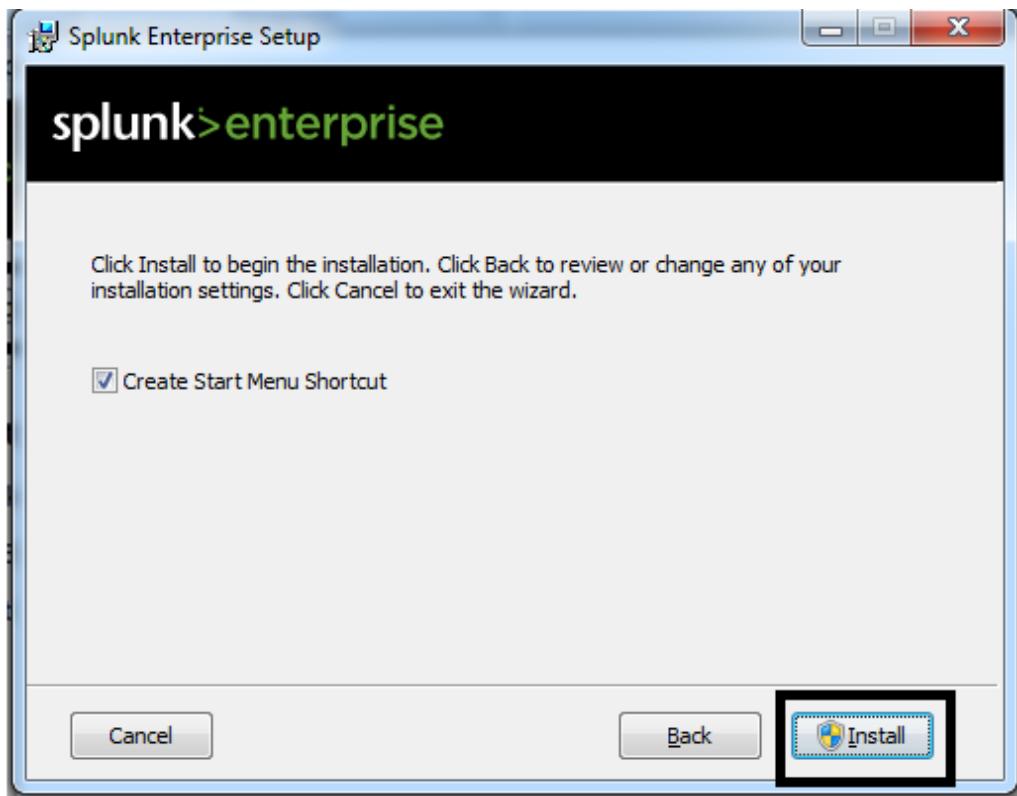


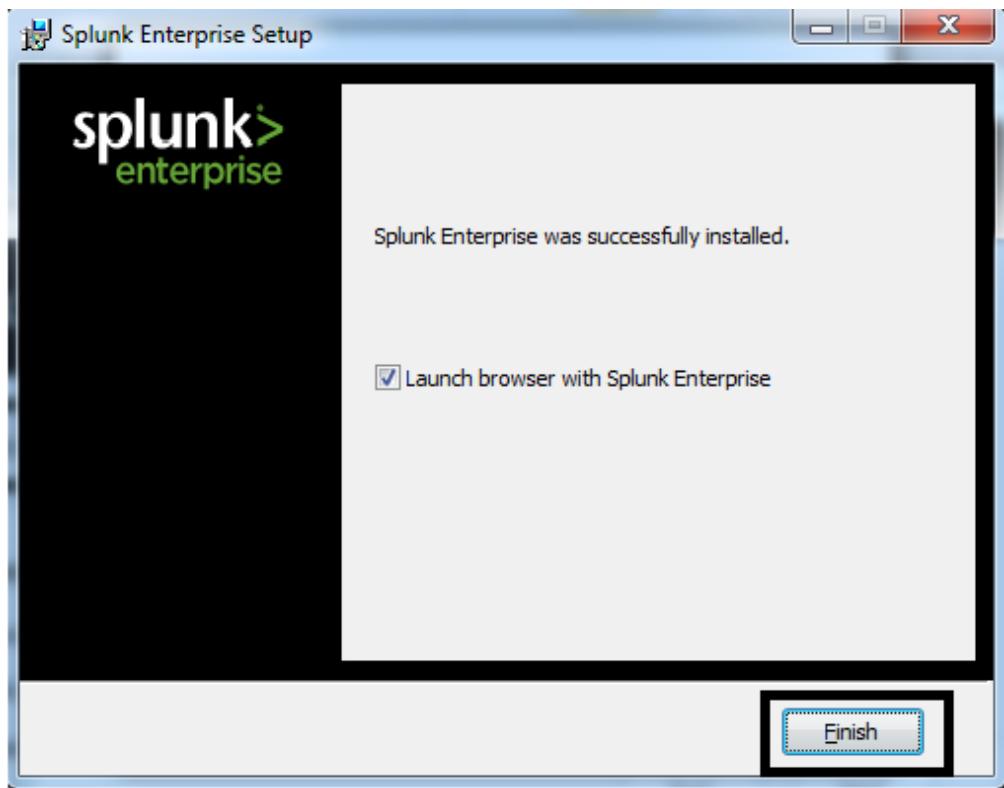
## **2.4 SPLUNK INSTALLATION IN WINDOWS**

Download the Splunk file from the Splunk Website according to the Windows System.







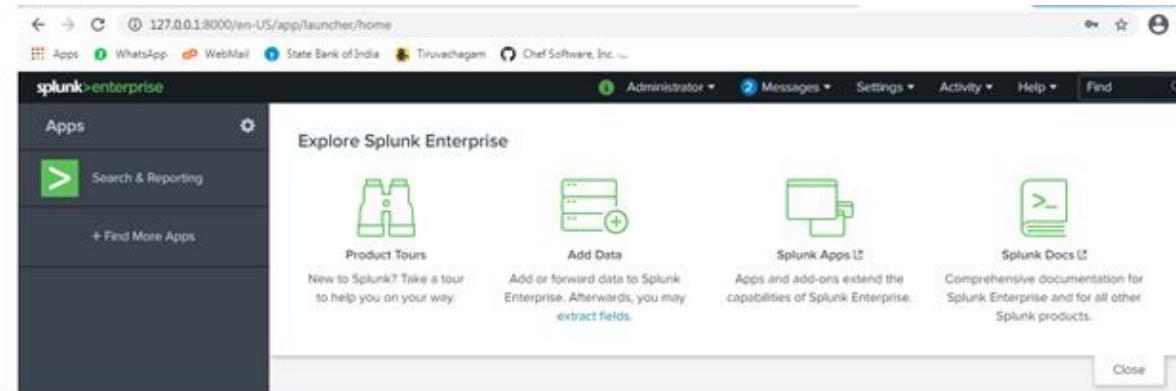


**Launch Browser & open the URL**

**localhost:8000**

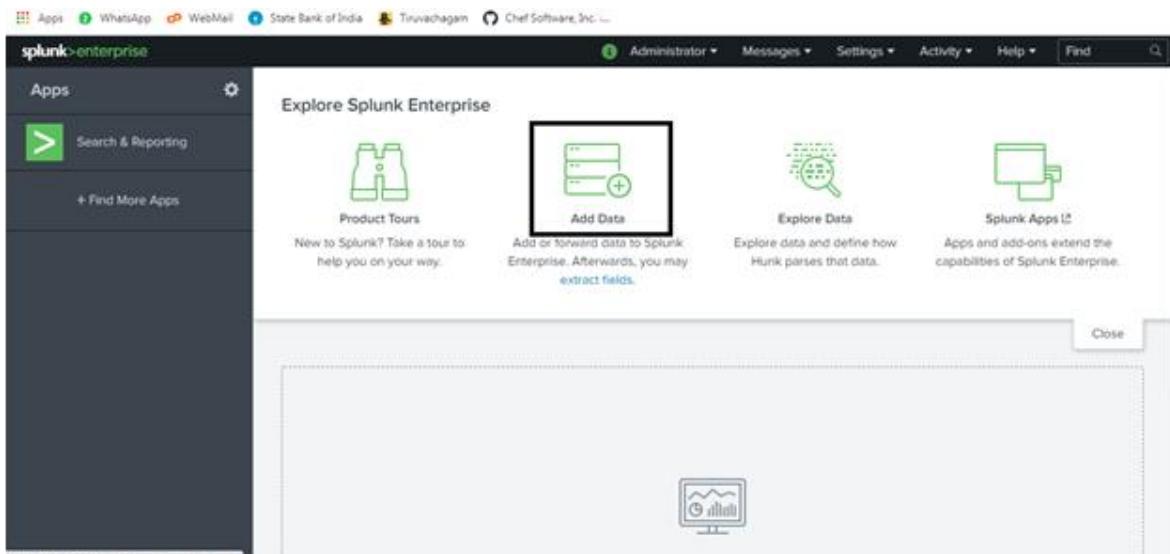


**Splunk Dashboard:**



## 2.5 LOADING THE DATA IN SPLUNK:

Load this File- Sampledata



Scroll down & select the Upload Button

The screenshot shows the Splunk mobile application interface. At the top, there is a navigation bar with icons for Apps, WhatsApp, WebMail, State Bank of India, Tiruvachagam, and Chef Software, Inc. Below the navigation bar, there is a section titled "Data Sources" with four cards:

- Cloud: Get your cloud computing data in to the Splunk platform. 10 data sources.
- Networking: Get your networking data in to the Splunk platform. 2 data sources.
- System: Get your operating system data in to the Splunk platform. 1 data source.
- Security: Get your security data in to the Splunk platform. 3 data sources.

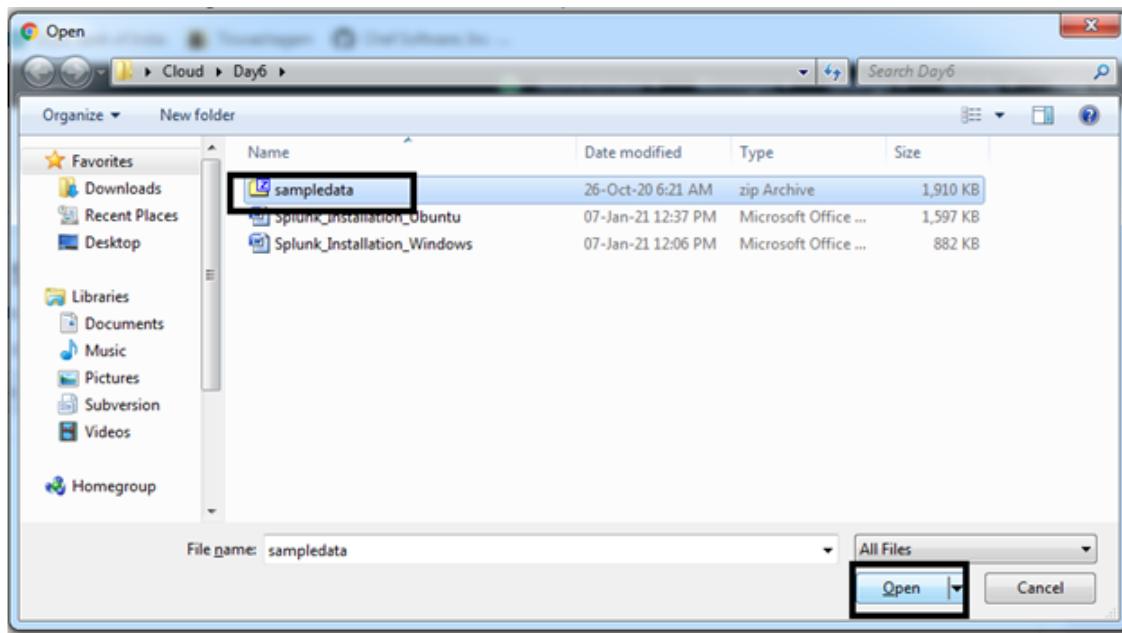
Below these cards, it says "4 data sources in total".

At the bottom, there is a section titled "Or get data in with the following methods" with three options:

- Upload**: Files from my computer. Sub-options: Local log files, Local structured files (e.g. CSV), Tutorial for adding data.
- Monitor**: Files and ports on this Splunk platform instance. Sub-options: Files, HTTP, VM, TCP/UDP, Scripts, Modular inputs for external data sources.
- Forward**: Data from a Splunk forwarder. Sub-options: Files, TCP/UDP, Scripts.

The screenshot shows the Splunk enterprise web interface. At the top, there is a header with the Splunk logo, "splunk>enterprise", and "Apps". On the right side of the header, there are links for "Administrator", "Messages", "Settings", "Activity", "Help", and "Find". Below the header, there is a breadcrumb trail: "Add Data" > "Select Source".

The main content area is titled "Select Source" and contains the following instructions: "Choose a file to upload to the Splunk platform, either by browsing your computer or by dropping a file into the target box below. Learn More". Below this, there is a "Selected File: No file selected" message and a "Select File" button.



splunk>enterprise Apps ▾

Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Add Data

Select Source Input Settings Review Done

< Back Next >

### Select Source

Choose a file to upload to the Splunk platform, either by browsing your computer or by dropping a file into the target box below. [Learn More](#)

**⚠ Preview is not supported for this archive file, but it can still be indexed.**

Selected File: **sampledata.zip**

Select File

Add Data

Select Source Input Settings Review Done

< Back Review > 2

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

Automatic Select New

Host

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Constant value  
 Regular expression on path  
 Segment in path

Host field value **sampledata**

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your

Index Default Create a new index

splunk>enterprise Apps ▾

Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

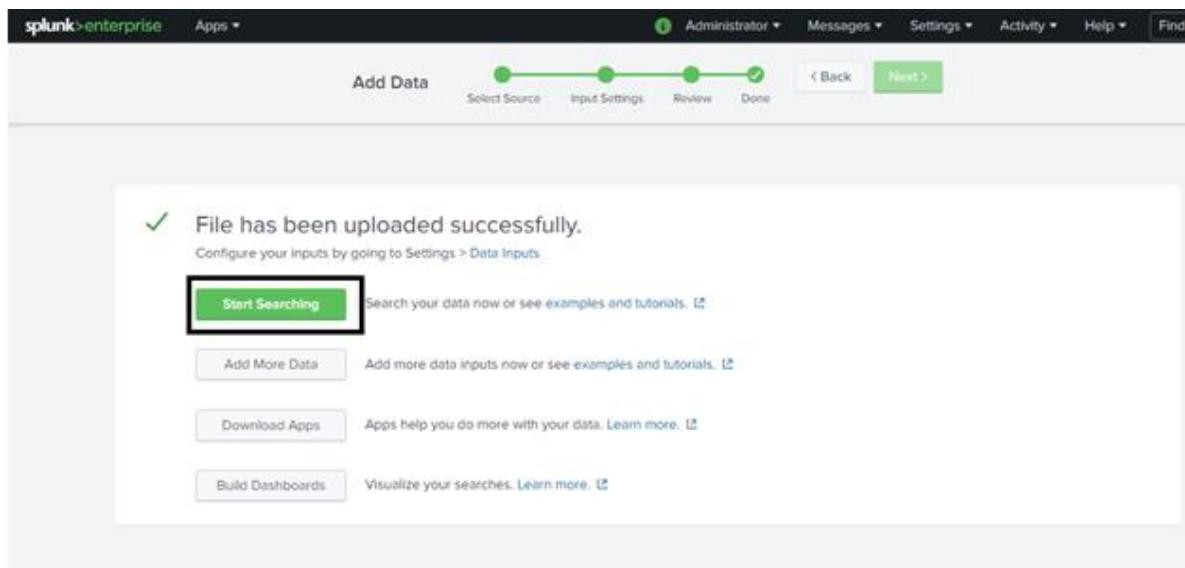
Add Data

Select Source Input Settings Review Done

< Back Submit >

### Review

Input Type	.....	Uploaded File
File Name	.....	sampledata.zip
Source Type	.....	Automatic
Host	.....	sampledata
Index	.....	Default



source="sampledata.zip:/\*" host="sampledata"

The screenshot shows the Splunk search interface with the search bar containing 'source="sampledata.zip:/\*" host="sampledata"'. The results section displays 109,864 events from 10/24/2020. The results table includes columns for Time and Event. The first few rows of the table are:

Time	Event
10/24/20 6:24:02.000 PM	[24/Oct/2020:18:24:02] VendorID=5036 Code=B AcctID=6024298300471575 host = sampledata   source = sampledata.zip:/vendor_sales/vendor_sales.log sourcetype = vendor_sales
10/24/20 6:23:46.000 PM	[24/Oct/2020:18:23:46] VendorID=7026 Code=C AcctID=8702194102896748 host = sampledata   source = sampledata.zip:/vendor_sales/vendor_sales.log sourcetype = vendor_sales
10/24/20 6:23:31.000 PM	[24/Oct/2020:18:23:31] VendorID=1043 Code=B AcctID=2063718909897951 host = sampledata   source = sampledata.zip:/vendor_sales/vendor_sales.log sourcetype = vendor_sales

## 2.6 QUERYING THE LINUX SYSTEM LOGS FROM SPLUNK:

The screenshot shows the Splunk Enterprise dashboard. On the left, there's a sidebar titled 'Apps' with a green arrow icon and the text 'Search & Reporting'. Below it is a link '+ Find More Apps'. The main content area is titled 'Explore Splunk Enterprise' and contains four cards: 'Product Tours' (binoculars icon), 'Add Data' (server icon), 'Explore Data' (magnifying glass icon), and 'Splunk Apps' (monitor icon). Each card has a brief description and a 'Close' button in the bottom right corner.

Or get data in with the following methods



**TYPE THIS IN THE PUTTY:** cd /var/log

```
azureuser@splunkvm:/opt/splunk$ cd /var/log
azureuser@splunkvm:/var/log$ ls
alternatives.log  cloud-init-output.log  kern.log    syslog
apt              cloud-init.log        landscape   tallylog
auth.log         dist-upgrade       lastlog     unattended-upgrades
azure            dpkg.log           lxd          waagent.log
btmp             journal           samba       wtmp
azureuser@splunkvm:/var/log$
```

The screenshot shows the 'Add Data' configuration wizard. Step 1: Select Source. The 'Select Source' tab is selected. A box highlights the 'Files & Directories' section, which contains the instruction 'Upload a file, index a local file, or monitor an entire directory.' Below it are sections for 'HTTP Event Collector', 'TCP / UDP', and 'Scripts'.

**Step 1: Select Source**

Files & Directories  
Upload a file, index a local file, or monitor an entire directory.

HTTP Event Collector  
Configure tokens that clients can use to send data over HTTP or HTTPS.

TCP / UDP  
Configure the Splunk platform to listen on a network port.

Scripts  
Get data from any API, service, or database with a script.

**Step 2: Set Source Type**

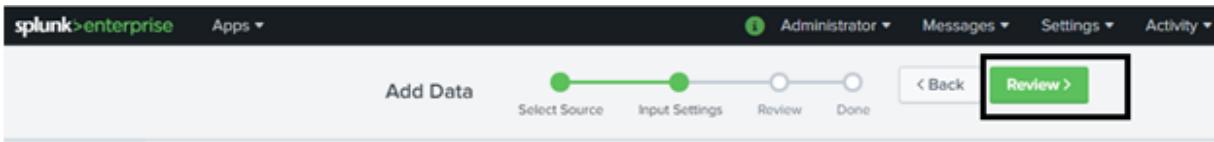
File or Directory: /var/log

Continuously Monitor      Index Once

**Step 3: Input Settings**

Review

Next >



#### Host

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

- Constant value
- Regular expression on path
- Segment in path

Host field value

splunkvm1

#### Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for

Index

Default

[Create a new index](#)

**Submit >**

#### Review

Input Type .....	Directory Monitor
Source Path .....	/var/log
Whitelist .....	N/A
Blacklist .....	N/A
Source Type .....	Automatic
App Context .....	search
Host .....	splunkvm1
Index .....	default

**Next >**

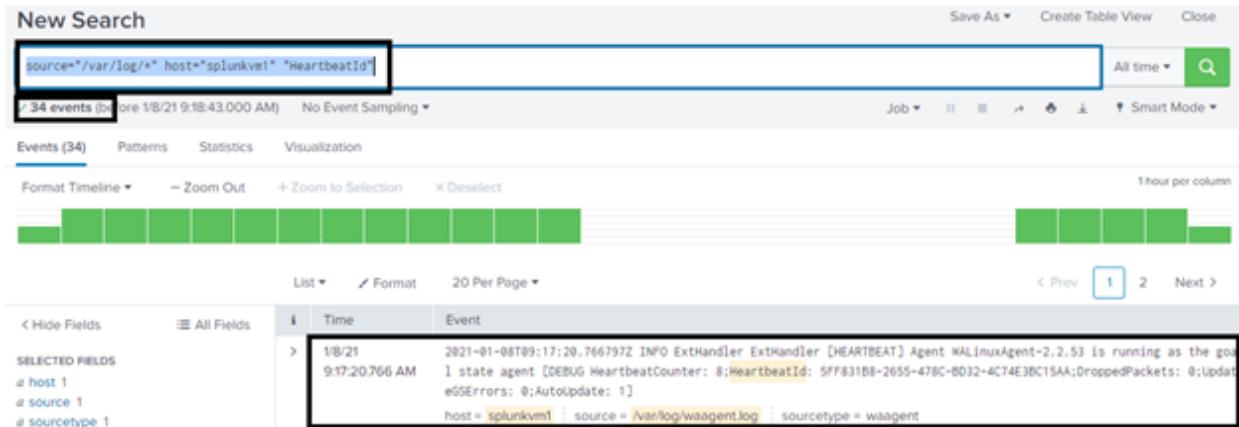
✓ File input has been created successfully.

Configure your inputs by going to [Settings > Data Inputs](#)

**Start Searching**

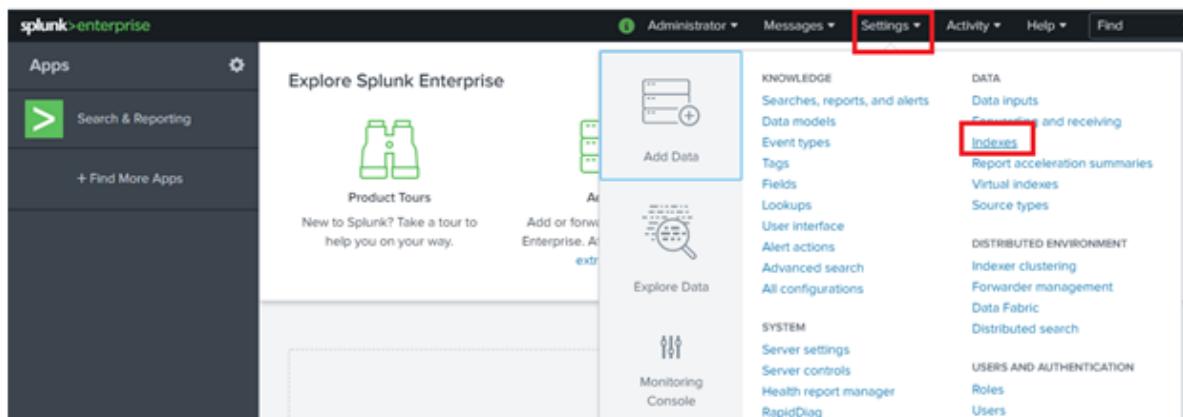
Search your data now or see examples and tutorials.

```
source="/var/log/*" host="splunkvm1" "HeartbeatId"
```



## 2.7 ADDING NEW INDEX IN SPLUNK:

Go to the Splunk software and click on to the settings and open the Indexes function to add more Indexes.



splunk>enterprise Apps

Administrator Messages Settings Activity Help Find

Indexes

A repository for data in Splunk Enterprise. Indexes reside in flat files on the Splunk Enterprise instance known as the indexer. Learn more ↗

New Index

11 Indexes Filter 20 per page

Name	Actions	Type	App	Current Size	Max Size	Event Count	Earliest Event	Latest Event	Home Path	Frozen Path
_audit	Edit Delete Disable	Events	system	4 MB	488.28 GB	26.9K	a day ago	in a few seconds	\$SPLUNK_D/B/audit/db	N/A

## New Index

X

### General Settings

Index Name

weblogIndex

Set index name (e.g., INDEX\_NAME). Search using index=INDEX\_NAME.

Index Data Type

Events

Metrics

The type of data to store (event-based or metrics).

Home Path

optional

Hot/warm db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/db).

Cold Path

optional

Cold db path. Leave blank for default (\$SPLUNK\_DB/INDEX\_NAME/coldb).

Thawed Path

optional

Save

Cancel

Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.

Max Size of Entire Index

10

GB ▾

Maximum target size of entire index.

Max Size of Hot/Warm/Cold Bucket

auto

GB ▾

Maximum target size of buckets. Enter 'auto\_high\_volume' for high-volume indexes.

Frozen Path

optional

Frozen bucket archive path. Set this if you want Splunk to automatically archive frozen buckets.

App

Search & Reporting ▾

**New Index**

---

Hot/Warm/Cold Bucket

Maximum target size of buckets. Enter 'auto\_high\_volume' for high-volume indexes.

Frozen Path

optional

Frozen bucket archive path. Set this if you want Splunk to automatically archive frozen buckets.

App

Search & Reporting ▾

**Storage Optimization**

Tsidx Retention Policy

**Enable Reduction** Disable Reduction

Warning: Do not enable reduction without understanding the full implications. It is extremely difficult to rebuild reduced buckets. [Learn More](#)

Reduce tsidx files older than 90 Days ▾

Age is determined by the latest event in a bucket.

---

**Save** Cancel

## 2.8 CREATING A ROLE IN SPLUNK

The screenshot shows the Splunk Enterprise interface. The top navigation bar includes 'Administrator', 'Messages', 'Settings' (which is highlighted with a red box), 'Activity', 'Help', and 'Find'. On the left, there's a sidebar with 'Apps' and a 'Search & Reporting' icon. The main content area has a 'Explore Splunk Enterprise' section with 'Product Tours' and 'Add Data' (which is highlighted with a blue box). To the right, there are several categories: 'KNOWLEDGE' (Searches, reports, and alerts; Data models; Event types; Tags; Fields; Lookups; User interface; Alert actions; Advanced search; All configurations), 'DATA' (Data inputs; Forwarding and receiving; Indexes; Report acceleration summaries; Virtual indexes; Source types), 'DISTRIBUTED ENVIRONMENT' (Indexer clustering; Forwarder management; Data Fabric; Distributed search), 'SYSTEM' (Server settings; Server controls; Health report manager; RapidDiag; Instrumentation), and 'USERS AND AUTHENTICATION' (Roles; **Users** (highlighted with a red box); Tokens). A red box also highlights the 'Settings' button in the top navigation.

The screenshot shows the 'Users' table in the Splunk Enterprise Settings page. The table header includes columns for Name, Actions, Authentication system, Full name, Email address, Time zone, Default app, Default app inherited from, Roles, Last Login, and Status. There is one entry: 'admin' (Actions: Edit, Authentication system: Splunk, Full name: Administrator, Email address: changeme@example.com, Time zone: launcher, Default app: system, Roles: admin, Last Login: 1/8/2021, Status: Active). A red box highlights the 'New User' button at the bottom right of the table.

Name	Actions	Authentication system	Full name	Email address	Time zone	Default app	Default app inherited from	Roles	Last Login	Status
admin	Edit	Splunk	Administrator	changeme@example.com	launcher	system		admin	1/8/2021, 2:39:20 PM	Active

## Create User

X

Name  sainatarajan

Full name

Email address

Set password  \*\*\*\*\*

Confirm password

Password must contain at least ? ✓ 8 characters

Time zone ?

Default app ?

|| teams.microsoft.com is sharing your screen.

## Create User

X

✓ 8 characters

Time zone ?

Default app ?

Assign roles ?

admin	admin
can_delete	can_delete
power	
splunk-system-role	

Create a role for this user

Require password change  on first login

Save

Administrator

Logout

Name	Actions	Authentication system	Full name	Email address	Time zone	Default app	Default app inherited from	Roles	Last Logon
admin	Edit	Splunk	Administrator	changeme@example.com	UTC	launcher	system	admin	1/8/2021, 2:39:20 PM
sainatarajan	Edit	Splunk	D Sai Natarajan	d.sainatarajan@duratechsolutions.in	UTC	launcher	system	admin, can_delete	

sainatarajan

.....

Sign In

First time signing in?  
You have been logged out. Log in to return to the system.

The screenshot shows the Splunk Enterprise interface with the 'Administrator' user logged in. The top navigation bar includes 'Messages', 'Settings' (which is currently selected and highlighted with a red box), 'Activity', 'Help', and 'Find'. Below the navigation bar is a sidebar with three main sections: 'Add Data', 'Explore Data', and 'Monitoring Console'. The 'Add Data' section contains icons for databases and files, with a 'Add Data' button below. The 'Explore Data' section contains an icon of a magnifying glass over a grid, with a 'Explore Data' button below. The 'Monitoring Console' section contains an icon of three vertical bars with switches, with a 'Monitoring Console' button below. The main content area is titled 'SETTINGS' and contains several categories: 'KNOWLEDGE' (Searches, reports, and alerts; Data models; Event types; Tags; Fields; Lookups; User interface; Alert actions; Advanced search; All configurations), 'DATA' (Data inputs; Forwarding and receiving; Indexes; Report acceleration summaries; Virtual indexes; Source types), 'DISTRIBUTED ENVIRONMENT' (Indexer clustering; Forwarder management; Data Fabric; Distributed search), 'SYSTEM' (Server settings; Server controls; Health report manager; RapidDiag; Instrumentation; Licensing; Workload management), and 'USERS AND AUTHENTICATION' (Roles, which is highlighted with a red box; Users; Tokens; Password Management; Authentication Methods). A 'New Role' button is located at the bottom right of the 'Roles' section.

The screenshot shows the 'Roles' page in Splunk Enterprise. At the top, there is a header with 'splunk>enterprise' and 'Apps'. The main content area is titled 'Roles' and displays a table of existing roles. The table has columns for 'Name', 'Actions', 'Native capabilities', 'Inherited capabilities', and 'Default App'. The rows show five roles: 'admin' (Actions: Edit, Native capabilities: 92, Inherited capabilities: 32), 'can\_delete' (Actions: Edit, Native capabilities: 4, Inherited capabilities: 0), 'power' (Actions: Edit, Native capabilities: 9, Inherited capabilities: 23), 'splunk-system-role' (Actions: Edit, Native capabilities: 0, Inherited capabilities: 124), and 'user' (Actions: Edit, Native capabilities: 23, Inherited capabilities: 0). A green 'New Role' button is located at the bottom right of the table area.

Name	Actions	Native capabilities	Inherited capabilities	Default App
admin	Edit	92	32	
can_delete	Edit	4	0	
power	Edit	9	23	
splunk-system-role	Edit	0	124	
user	Edit	23	0	

## New Role

Name \*  newrole

1. Inheritance    2. Capabilities    3. Indexes    4. Restrictions    5. Resources

Specify roles from which this role inherits capabilities and indexes. Inherited capabilities and indexes cannot be disabled. If multiple roles are specified, this role inherits capabilities and indexes from all selected roles.

<input type="checkbox"/>	Role name	filter	Showing all ▾
<input type="checkbox"/>	admin		

## New Role

1. Inheritance    2. Capabilities    3. Indexes    4. Restrictions    5. Resources

Specify roles from which this role inherits capabilities and indexes. Inherited capabilities and indexes cannot be disabled. If multiple roles are specified, this role inherits capabilities and indexes from all selected roles.

<input type="checkbox"/>	Role name	filter	Showing all ▾
<input type="checkbox"/>	admin		
<input type="checkbox"/>	can_delete		
<input checked="" type="checkbox"/>	power		
<input type="checkbox"/>	splunk-system-role		
<input checked="" type="checkbox"/>	user		

power user

Cancel Create

### New Role

1. Inheritance    **2. Capabilities**    3. Indexes    4. Restrictions    5. Resources

Select specific capabilities for this role.

Capability Name	filter	
accelerate_datamodel		Inherited
accelerate_search		Inherited
admin_all_objects		Inherited
<input checked="" type="checkbox"/> apps_backup		
<input checked="" type="checkbox"/> apps_restore		
change_authentication		

**Create**

**Cancel**

**Showing all**

### New Role

Name \* **newrole**

1. Inheritance    2. Capabilities    3. Indexes    4. Restrictions    **5. Resources**

#### This role

Default app  
**search**

Role search job limit  
Set a limit for how many search jobs that all users with this role can run at the same time. **?**

**Create**

**Cancel**

Name	Actions	Native capabilities	Inherited capabilities	Default App
admin	Edit	92	32	
can_delete	Edit	4	0	
newrole	Edit	2	32	search
power	Edit	9	23	
splunk-system-role	Edit	0	124	
user	Edit	23	0	

Some Example Queries with following steps in the Splunk Search Bar:

- sourcetype=access\_\* action=purchase status!=200 | stats count by productId | sort -count
- sourcetype=access\_\* action=purchase status=200 | timechart count as "Units Sold" by productId usenull=false useother=false

**Visualization -> Type -> Line Chart**

**Format -> Legend -> Left**

**Save As -> Dashboard Panel**

**New -> Title, Panel Title**

**Panel Content -> Line Chart**

- sourcetype=vendor\_sales | timechart count as "Units Sold By Code" by Code
- sourcetype=vendor\_sales | timechart count as "Units Sold By Code" by code => Will give wrong results -> Column names are case sensitive

**Run Search from Search History -> add to search**

**Save As Dashboard Panel -> Existing -> Select the previous dashboard created**

**View Dashboard**

**Edit Dashboard -> Change Format**

**Click & Drag the ::::::: to shift the chart to a different location**

- sourcetype=access\_\* action=purchase status=200 | stats count as "Units Sold"

**Visualization -> Format -> Single Value**

**Format -> Caption -> Units Sold Online**

**Save As Dashboard Panel -> Existing**

- sourcetype=vendor\_sales | stats count as "Units Sold By Vendors"  
    Visualization -> Format -> Single Value  
    Format -> Caption -> Units Sold By Vendors  
    Save As Dashboard Panel -> Existing
- sourcetype=vendor\_sales | stats count as "Units Sold By Vendor" by VendorID

## **PIVOTS**

The Pivot tool lets you report on a specific data set without the Splunk Search Processing Language (SPL). First, identify a dataset that you want to report on, and then use a drag-and-drop interface to design and generate **pivots** that present different aspects of that data in the form of tables, charts, and other visualizations.

The pivot command makes simple pivot operations fairly straightforward, but can be pretty complex for more sophisticated pivot operations. Fundamentally this command is a wrapper around the stats and xseries commands.

The pivot command does not add new behavior, but it might be easier to use if you are already familiar with how Pivot works. See the [Pivot Manual](#). Also, read how to [open non-transforming searches in Pivot](#).

Run pivot searches against a particular **data model** object. This requires a large number of inputs: the data model, the data model object, and pivot elements.

### **Syntax**

| pivot <datamodel-name> <object-name> <pivot-element>

#### **Required arguments**

**datamodel-name**

**Syntax:** <string>

**Description:** The name of the data model to search.

**objectname**

**Syntax:** <string>

**Description:** The name of a data model object to search.

## pivot element

**Syntax:** (<cellvalue>)\* (SPLITROW <rowvalue>)\* (SPLITCOL colvalue [options])\* (FILTER <filter expression>)\* (LIMIT <limit expression>)\* (ROWSUMMARY <true | false>)\* (COLSUMMARY <true | false>)\* (SHOWOTHER <true | false>)\* (NUMCOLUMNS <num>)\* (rowsort [options])\*

**Description:** Use pivot elements to define your pivot table or chart. Pivot elements include cell values, split rows, split columns, filters, limits, row and column formatting, and row sort options. Cell values always come first. They are followed by split rows and split columns, which can be interleaved, for example: avg(val), SPLITCOL foo, SPLITROW bar, SPLITCOL baz.

### Cell value

<cellvalue>

**Syntax:** <function>(fieldname) [AS <label>]

**Description:** Define the values of a cell and optionally rename it. Here, label is the name of the cell in the report.

The set of allowed functions depend on the data type of the fieldname:

- ★ **Strings:** list, values, first, last, count, and distinct\_count (dc)
- ★ **Numbers:** sum, count, avg, max, min, stdev, list, and values
- ★ **Timestamps:** duration, earliest, latest, list, and values
- ★ **Object or child counts:** count

### Descriptions for row split-by elements

SPLITROW <rowvalue>

**Syntax:** SPLITROW <field> [AS <label>] [RANGE start=<value> end=<value> max=<value> size=<value>] [PERIOD (auto | year | month | day | hour | minute | second)] [TRUENAME <label>] [FALSENAME <label>]

**Description:** You can specify one or more of these options on each SPLITROW. The options can appear in any order. You can rename the <field> using "AS <label>", where "label" is the name of the row in the report.

Other options depend on the data type of the <field> specified:

- ★ RANGE applies only for numbers. You do not need to specify all of the options (start, end, max, and size).
- ★ PERIOD applies only for timestamps. Use it to specify the period to bucket by.

- ★ TRUELABEL applies only for booleans. Use it to specify the label for true values.
- ★ FALSELABEL applies only for booleans. Use it to specify the label for false values.

## Descriptions for column split-by elements

### SPLITCOL colvalue <options>

**Syntax:** fieldname [ RANGE start=<value> end=<value> max=<value> size=<value>] [PERIOD (auto | year | month| day | hour | minute | second)] [TRUELABEL <label>] [FALSELABEL <label>]

**Description:** You can have none, some, or all of these options on each SPLITCOL. They may appear in any order.

Other options depend on the data type of the field specified (fieldname):

- ★ RANGE applies only for numbers. The options (start, end, max, and size) do not all have to be specified.
- ★ PERIOD applies only for timestamps. Use it to specify the period to bucket by.
- ★ TRUELABEL applies only for booleans. Use it to specify the label for true values.
- ★ FALSELABEL applies only for booleans. Use it to specify the label for false values.

## Descriptions for filter elements

### Filter <filter expression>

**Syntax:** <fieldname> <comparison-operator> <value>

**Description:** The expression used to identify values in a field. The comparison operator that you use depends on the type of field value.

- ★ **Strings:** is, contains, in, isNot, doesNotContain, startsWith, endsWith, isNull, isNotNull
  - For example: ... filter *fieldname* in (*value1*, *value2*, ...)
- ★ **ipv4:** is, contains, isNot, doesNotContain, startsWith, isNull, isNotNull
- ★ **Numbers:** =, !=, <, <=, >, >=, isNull, isNotNull
- ★ **Booleans:** is, isNull, isNotNull

## Descriptions for limit elements

### Limit <limit expression>

**Syntax:** LIMIT <fieldname> BY <limittype> <number> <stats-function>(<fieldname>)

**Description:** Use to limit the number of elements in the pivot. The limittype argument specifies where to place the limit. The valid values are top or bottom. The number

argument must be a positive integer. You can use any stats function, such as min, max, avg, and sum.

**Example:** LIMIT foo BY TOP 10 avg(bar)

### Usage

The pivot command is a [report-generating command](#). See [Command types](#).

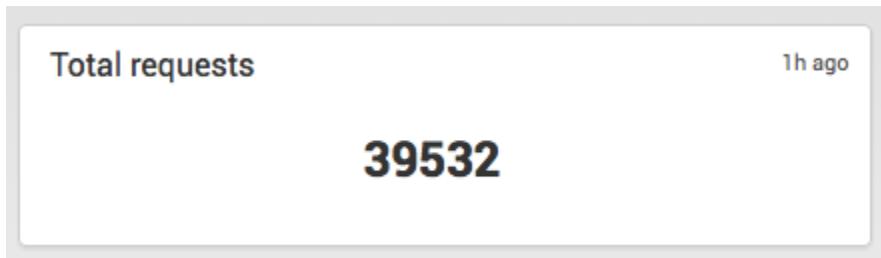
Generating commands use a leading pipe character and should be the first command in a search.

### Examples

**Example 1:** This command counts the number of events in the "HTTP Requests" object in the "Tutorial" data model.

```
| pivot Tutorial HTTP_requests count(HTTP_requests) AS "Count of HTTP requests"
```

This can be formatted as a single value report in the dashboard panel:



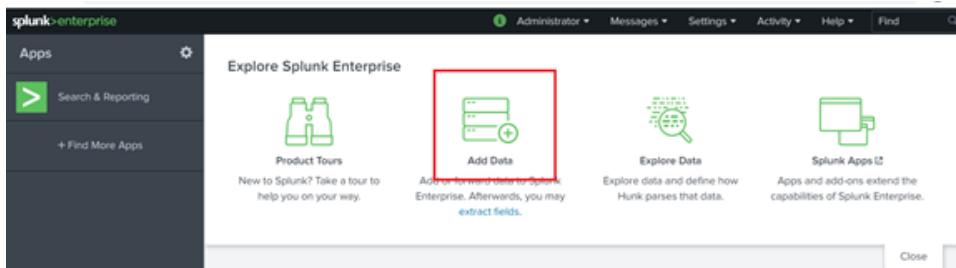
**Example 2:** Using the Tutorial data model, create a pivot table for the count of "HTTP Requests" per host.

```
| pivot Tutorial HTTP_requests count(HTTP_requests) AS "Count" SPLITROW host AS "Server" SORT 100 host
```

Total requests by server		1h ago
Server	Count	
www1	13628	
www2	12912	
www3	12992	

## Creating a Pivot in Splunk

Pivots in Splunk:



Select Upload and upload the file

A screenshot of the 'Or get data in with the following methods' section. It shows three options: 'Upload' (highlighted with a red box), 'Monitor', and 'Forward'. Each option has a brief description and a link to more information. The URL at the bottom of the page is 'http://localhost:8000/manage/search/adddatamethods/selectsource?input\_mode=0'.

A screenshot of the 'Add Data' wizard, Step 1: Select Source. It shows a file selection dialog box. The 'Select File' button is highlighted with a red box. The 'Open' button in the dialog is also highlighted with a red box. The URL at the bottom of the page is 'http://localhost:8000/manage/search/adddatamethods/selectsource?input\_mode=0'.

After uploading the data select the host field value in the input settings and Submit

Add Data

Select Source   Input Settings   Review   Done

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Constant value  
 Regular expression on path  
 Segment in path

Host field value

Add Data

Select Source   Input Settings   Review   Done

**Review**

Input Type	Uploaded File
File Name	tutorialdata.zip
Source Type	Automatic
Host	tutorialData
Index	Default

< Back **Submit >** Done

Go to settings and select Data models

Administrator ▾ Messages ▾ **Settings ▾** Activity ▾ Help ▾ Find

 Add Data



<b>KNOWLEDGE</b>	<b>DATA</b>
<a href="#">Searches, reports, and alerts</a>	<a href="#">Data inputs</a>
<b>Data models</b>	<a href="#">Forwarding and receiving</a>
<a href="#">Event types</a>	<a href="#">Indexes</a>
<a href="#">Tags</a>	<a href="#">Report acceleration summaries</a>
<a href="#">Fields</a>	<a href="#">Virtual indexes</a>
<a href="#">Lookups</a>	<a href="#">Source types</a>
<a href="#">User interface</a>	
<a href="#">Alert actions</a>	

**DISTRIBUTED ENVIRONMENT**

**Data Models**

Data models enable users to easily create reports in the Pivot tool. [Learn More](#)

Upload Data Model **New Data Model**

0 Data Models	App: Home (launcher) ▾	Visible in the App ▾	Owner: Any ▾	filter	Actions	Type ▾	App ▾	Owner ▾	Sharing ▾	20 per page ▾

Create a new Data model with a title and App

## New Data Model

X

Title

ID ?

The data model ID can only contain letters, numbers, dashes, and underscores. Do not start the data model ID with a period.

App

Description

Cancel

Create

## pivotTutorial

pivotTutorial

< All Data Models

Datasets

Add Dataset ▾

Root Event

Root Search

### Add Event Dataset

Data Model: pivotTutorial

Documentation ↗

! In handler 'datamodeledit': Error in 'pivotTutorial': Dataset constraints must specify at least one index.

Dataset Name

tutorialData

Constraints

index-default [source-type-access, -]

The search must have an explicit index constraint to maximize performance.

Example:

index-main uri="\*.php" OR uri="\*.js"

index-main NOT (referenceId OR referenceId\*)

The dataset ID can only contain letters, numbers, dashes, and underscores. Do not start the dataset ID with a period.

Cancel

Preview

Save

## Select the tutorialData and Select Pivot

The screenshot shows the 'pivotTutorial' interface. In the left sidebar, under 'EVENTS', the 'tutorialData' dataset is selected, highlighted with a red box. At the top right, there are buttons for 'Edit', 'Download', 'Pivot' (which is highlighted with a red box), and 'Documentation'. The main area displays the 'tutorialData' dataset details, including its constraints. One constraint is shown: 'index=main sourcetype=access\_.\*'. Below this, there's a 'Bulk Edit' button and an 'INHERITED' section listing fields: '\_time' (Time type), 'host' (String type, Override), 'source' (String type, Override), and 'sourcetype' (String type, Override).

Add the dataset name and the necessary constraints.

The screenshot shows the 'Add Event Dataset' dialog for the 'tutorialPivot' data model. In the 'Dataset Name' field, 'tutorialData' is entered, highlighted with a red box. In the 'Constraints' field, 'index=main sourcetype=access\_.\*' is entered, also highlighted with a red box. At the bottom right, there are 'Cancel', 'Preview', and 'Save' buttons, with 'Save' highlighted with a red box.

The screenshot shows the 'tutorialPivot' interface. In the left sidebar, under 'EVENTS', the 'tutorialData' dataset is selected, highlighted with a red box. At the top right, there are buttons for 'Edit', 'Download', 'Pivot' (highlighted with a red box), and 'Documentation'. The main area displays the 'tutorialData' dataset details, including its constraints. The previously added constraint 'index=main sourcetype=access\_.\*' is now listed in the 'CONSTRAINTS' section, highlighted with a red box. Below it, there's a 'Bulk Edit' button and an 'INHERITED' section listing fields: '\_time' (Time type), 'host' (String type, Override), 'source' (String type, Override), and 'sourcetype' (String type, Override).

## Select a Dataset

1 Object in tutorialPivot

> tutorialData

Select the pivot and check for various Visualizations

New Pivot

✓ 39,532 events (before 1/18/21 8:08:28.000 AM)

Filters

All time

Split Columns

+ (button)

Split Rows

+ (button)

Column Values

Count of tutorialData

39532

New Pivot

✓ 39,532 events (before 1/18/21 8:08:28.000 AM)

Filters

All time

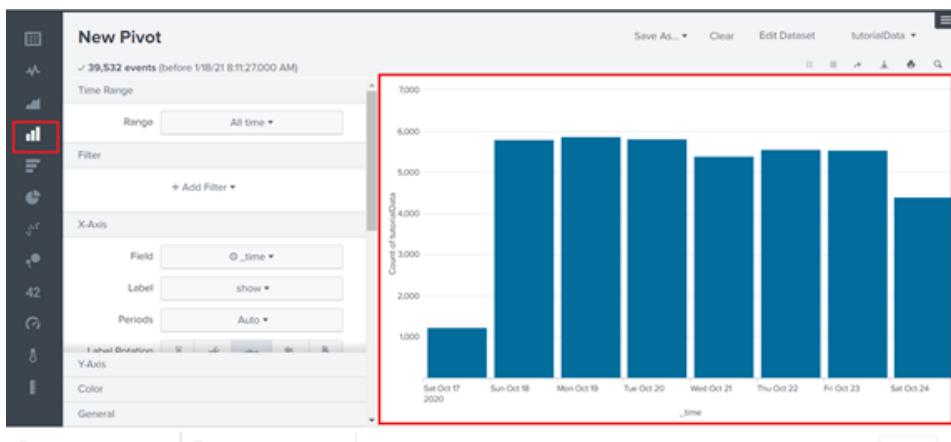
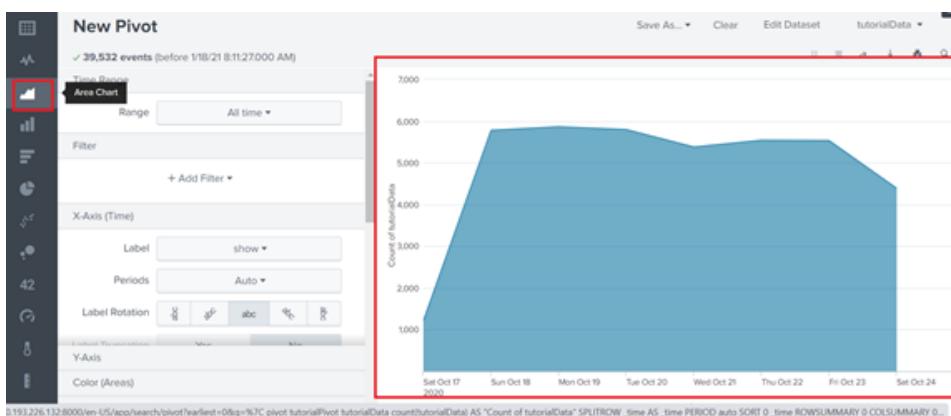
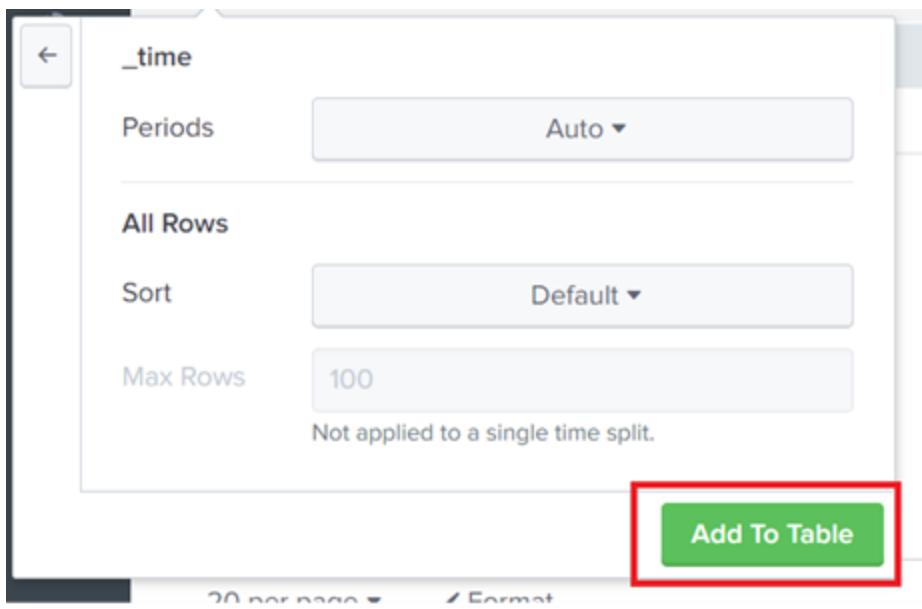
Split Rows

+ (button)

Time

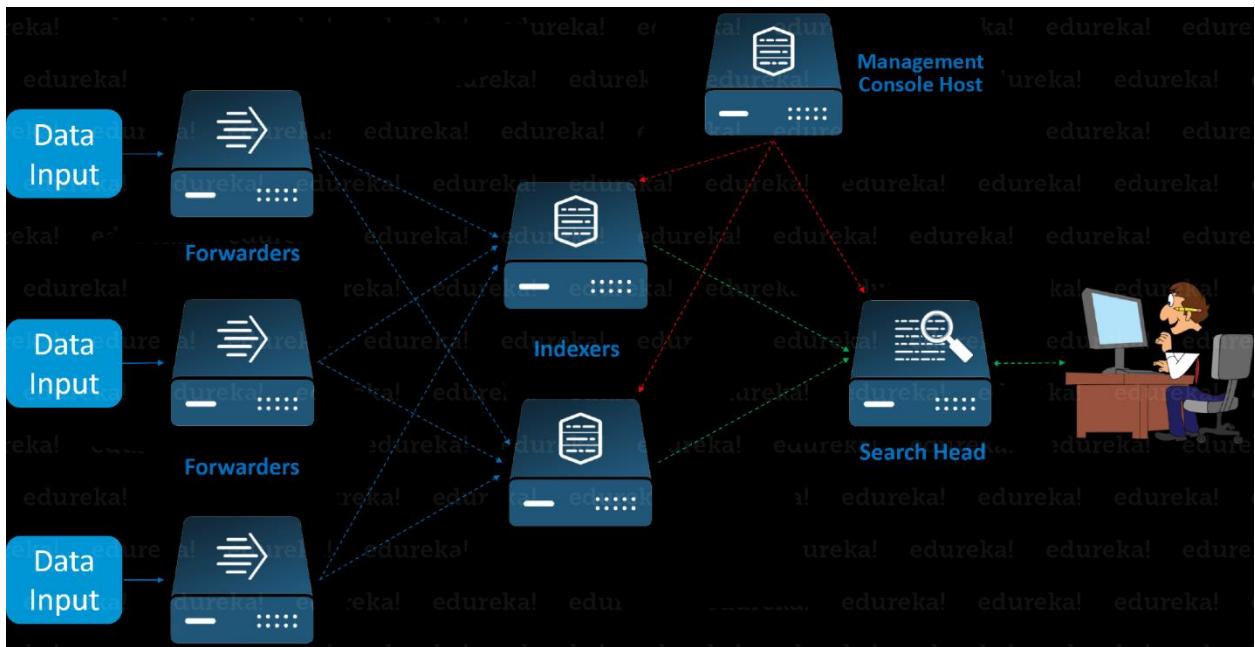
Field

a host  
a source  
a sourcetype



## Forwarders in Splunk

Forwarders forward and route data on an index-by-index basis. By default, they forward all external data, as well as data for the \_audit internal index. In some cases, they also forward data for the \_internal internal index. You can change this behavior as necessary



### Types of forwarders

There are three types of forwarders:

- ★ The **universal forwarder** contains only the components that are necessary to forward data. Learn more about the universal forwarder in the [Universal Forwarder](#) manual.
- ★ A **heavy forwarder** is a full Splunk Enterprise instance that can index, search, and change data as well as forward it. The heavy forwarder has some features disabled to reduce system resource usage.
- ★ A **light forwarder** is also a full Splunk Enterprise instance, with more features disabled to achieve as small a resource footprint as possible. The light forwarder has been deprecated as of Splunk Enterprise version 6.0. The universal forwarder supersedes the light forwarder for nearly all purposes and represents the best tool for sending data to indexers.

## The universal forwarder

The sole purpose of the universal forwarder is to forward data. Unlike a full Splunk instance, you cannot use the universal forwarder to index or search data. To achieve higher performance and a lighter footprint, it has several limitations:

- ★ The universal forwarder cannot search, index, or produce alerts with data.
- ★ The universal forwarder does not **parse** data. You cannot use it to route data to different Splunk indexers based on its contents.
- ★ Unlike full Splunk Enterprise, the universal forwarder does not include a bundled version of Python.

The universal forwarder can get data from a variety of inputs and forward the data to a Splunk deployment for indexing and searching. It can also forward data to another forwarder as an intermediate step before sending the data onward to an indexer.

The universal forwarder is a separately downloadable piece of software. Unlike the heavy and light forwarders, you do not enable it from a full Splunk Enterprise instance. Learn more about the universal forwarder in the [Universal Forwarder](#) manual.

To learn how to download, install, and deploy a universal forwarder, see [Install the universal forwarder software](#) in the [Universal Forwarder](#) manual.

## Heavy and light forwarders

While the universal forwarder is the preferred way to forward data, you might need to use heavy or light forwarders if you need to analyze or make changes to the data before you forward it, or you need to control where the data goes based on its contents. Unlike the universal forwarder, both heavy and light forwarders are full Splunk Enterprise instances with certain features disabled. Heavy and light forwarders differ in capability and the corresponding size of their resource footprints.

A **heavy forwarder** (sometimes referred to as a "regular forwarder") has a smaller footprint than an indexer but retains most of the capability, except that it cannot perform distributed searches. Some of its default functionality, such as Splunk Web, can be disabled, if necessary, to reduce the size of its footprint. A heavy forwarder parses data before forwarding it and can route data based on criteria such as source or type of event.

One key advantage of the heavy forwarder is that it can index data locally, as well as forward data to another Splunk instance. You must activate this feature. See [Configure forwarders with outputs.conf](#) in this manual for details.

A **light forwarder** has a smaller footprint with much more limited functionality. It forwards only unparsed data. The universal forwarder, which provides very similar functionality, supersedes it. The light forwarder has been deprecated but continues to be available mainly to meet legacy needs.

When you install a universal forwarder, you can migrate checkpoint settings from any (version 4.0 or greater) light forwarder that resides on the same host. See [About the universal forwarder](#) in the *Universal Forwarder* manual for a more detailed comparison of universal and light forwarders.

For detailed information on the capabilities of heavy and light forwarders, see [Heavy and light forwarder capabilities](#) in this manual.

### Forwarder comparison

This table summarizes the similarities and differences among the three types of forwarders:

Features and capabilities	Universal forwarder	Light forwarder	Heavy forwarder
Type of Splunk Enterprise instance	Dedicated executable	Full Splunk Enterprise, with most features disabled	Full Splunk Enterprise, with some features disabled
Footprint (memory, CPU load)	Smallest	Small	Medium-to-large (depending on enabled features)
Bundles Python?	No	Yes	Yes
Handles data inputs?	All types (but scripted inputs might require Python installation)	All types	All types
Forwards to Splunk Enterprise?	Yes	Yes	Yes
Forwards to 3rd party systems?	Yes	Yes	Yes
Serves as intermediate forwarder?	Yes	Yes	Yes
Indexer acknowledgment	Optional	Optional (version 4.2 and later)	Optional (version 4.2 and later)

(guaranteed delivery)?			
Load balancing?	Yes	Yes	Yes
Data cloning?	Yes	Yes	Yes
Per-event filtering?	No	No	Yes
Event routing?	No	No	Yes
Event parsing?	Sometimes	No	Yes
Local indexing?	No	No	Optional, by setting indexAndForward attribute in outputs.conf
Searching/alerting?	No	No	Optional
Splunk Web?	No	No	Optional

For detailed information on specific capabilities, see the rest of this topic, as well as the other forwarding topics in the manual.

### Types of forwarder data

Forwarders can transmit three types of data:

- ★ Raw
- ★ Unparsed
- ★ Parsed

The type of data a forwarder can send depends on the type of forwarder it is, as well as how you configure it. Universal forwarders and light forwarders can send raw or unparsed data. Heavy forwarders can send raw or parsed data.

**With raw data**, the forwarder sends the data unaltered over a TCP stream. It does not convert the data into the Splunk communications format. The forwarder collects the data and sends it on. This is particularly useful for sending data to a non-Splunk system.

**With unparsed data**, a universal forwarder performs minimal processing. It does not examine the data stream, but it does tag the stream with metadata to identify source, source type, and host. It also divides the data stream into 64-kilobyte blocks and performs some rudimentary timestamping on the stream that the receiving indexer can use in case the events themselves have no discernible timestamps. The universal forwarder does not identify, examine, or tag individual events except when you configure it to parse files with structure data (such as comma-separated value files.)

**With parsed data**, a heavy forwarder breaks the data into individual events, which it tags and then forwards to a Splunk indexer. It can also examine the events. Because the data has been parsed, the forwarder can perform conditional routing based on event data, such as field values.

The parsed and unparsed formats are both referred to as **cooked** data, to distinguish them from raw data. By default, forwarders send cooked data (universal forwarders send unparsed data and heavy forwarders send parsed data.) To send raw data instead, set the `sendCookedData=false` attribute/value pair in `outputs.conf`.

## Creating a Heavy Forwarder in Splunk

Enable Receiving in Splunk Indexer

The screenshot shows the Splunk web interface with the following details:

- Header:** Administrator, Messages, **Settings** (highlighted with a red box), Activity, Help, Find.
- Left Sidebar:** Add Data (with a plus icon).
- Middle Column:**
  - KNOWLEDGE:** Searches, reports, and alerts; Data models; Event types; Tags; Fields; Lookups.
  - DATA:** Data inputs; **Forwarding and receiving** (highlighted with a red box); Indexes; Report acceleration summaries; Virtual indexes; Source types.
- Forwarding and receiving:**
  - Forward data:** Set up forwarding between two or more Splunk instances.
    - Forwarding defaults
    - Configure forwarding (+ Add new)
  - Receive data:** Configure this instance to receive data forwarded from other instances.
    - Configure receiving (highlighted with a red box)
    - + Add new

**Receive data**

Forwarding and receiving » Receive data

filter Search icon

25 per page Dropdown

New Receiving Port Green button with red border

There are no configurations of this type. Click the "New Receiving Port" button to create a new configuration.

**Add new**

Forwarding and receiving » Receive data » Add new

**Configure receiving**

Set up this Splunk instance to receive data from forwarder(s).

Listen on this port \* 9997 Red box

For example, 9997 will receive data on TCP port 9997.

Cancel Save Green button with red border

**Receive data**

Forwarding and receiving » Receive data

New Receiving Port Green button with red border

Successfully saved "9997".

Showing 1-1 of 1 item

filter Search icon

25 per page Dropdown

Listen on this port *	Status *	Actions
9997	Enabled   Disable	Delete

## Create a new Ubuntu VM with 1 CPU & 1 GB RAM(B1s) Create a virtual machine

Virtual machine name \* heavyForwarder Green checkmark

Region \* (Asia Pacific) Central India Down arrow

Availability options No infrastructure redundancy required Down arrow

Image \* Ubuntu Server 18.04 LTS - Gen1 Down arrow

See all images

Azure Spot instance

Size \* Standard\_B1s - 1 vcpu, 1 GiB memory (₹540.40/month) Down arrow

See all sizes

Administrator account

Authentication type SSH public key Blue radio button

---

Review + create < Previous Next : Disks >

Ensure the Splunk Indexer VM & the new Forwarder VM are in the same Virtual Network in Azure VNet of Indexer:

**splunkvm1 | Networking**

IP configuration: ipconfig1 (Primary)

Network Interface: splunkvm1520

NIC Public IP: 20.193.226.132

NIC Private IP: 10.0.0.11 Accelerated networking: Disabled

VNet of the Forwarder:

**heavyForwarder | Networking**

IP configuration: ipconfig1 (Primary)

Network Interface: heavyforwarder211

NIC Public IP: 52.140.96.91

NIC Private IP: 10.0.0.12 Accelerated networking: Disabled

**heavyForwarder**

Public IP address: 52.140.96.91

Connect to the Heavy Forwarder VM:

```
resources, services, and docs (13+)
azureuser@heavyForwarder: ~

System load: 0.07          Processes:        114
Usage of /: 4.5% of 28.90GB  Users logged in:   0
Memory usage: 20%          IP address for eth0: 10.0.0.12
Swap usage:  0%

0 packages can be updated.
0 of these updates are security updates.

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

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

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@heavyForwarder:~$
```

```
wget -O splunk-8.1.1-08187535c166-Linux-x86_64.tgz
https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86\_64&platform=linux&version=8.1.1&product=splunk&filename=splunk-8.1.1-08187535c166-Linux-x86\_64.tgz&wget=true
```

```
azureuser@heavyForwarder: ~
=splunk-8.1.1-08187535c166-Linux-x86_64.tgz&wget=true'
--2021-01-18 05:46:07-- https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.1&product=splunk&filename=splunk-8.1.1-08187535c166-Linux-x86_64.tgz&wget=true
Resolving www.splunk.com (www.splunk.com) ... 96.17.150.163, 96.17.150.155
Connecting to www.splunk.com (www.splunk.com) |96.17.150.163|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://download.splunk.com/products/splunk/releases/8.1.1/linux/splunk-8.1.1-08187535c166-Linux-x86_64.tgz [following]
--2021-01-18 05:46:08-- https://download.splunk.com/products/splunk/releases/8.1.1/linux/splunk-8.1.1-08187535c166-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com) ... 13.227.138.69, 13.227.138.129, 13.227.138.114, ...
Connecting to download.splunk.com (download.splunk.com) |13.227.138.69|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 508304047 (485M) [application/x-gzip]
Saving to: 'splunk-8.1.1-08187535c166-Linux-x86_64.tgz'

x86_64.tgz      67% [=====>] 326.98M 33.7MB/s eta 4s
```

```
tar -xvf splunk-8.1.1-08187535c166-Linux-x86_64.tgz
```

```
2021-01-18 05:46:27 (25.5 MB/s) - 'splunk-8.1.1-08187535c166-Linux-x86_64.tgz' saved [508304047/508304047]
```

```
azureuser@heavyForwarder:~$ tar -xvf splunk-8.1.1-08187535c166-Linux-x86_64.tgz
```

```
sudo mv splunk
```

```
/opt/ cd
```

```
/opt/splunk/bin
```

```
splunk/swidtag/splunk-Splunk-Enterprise-primary.swidtag  
splunk/splunk-8.1.1-08187535c166-linux-2.6-x86_64-manifest  
azureuser@heavyForwarder:~$ sudo mv splunk /opt/  
azureuser@heavyForwarder:~$ cd /opt/splunk/bin  
azureuser@heavyForwarder:/opt/splunk/bin$
```

```
sudo ./splunk start --accept-license
```

```
^C
```

```
azureuser@heavyForwarder:/opt/splunk/bin$ sudo ./splunk start --accept-license
```

```
azureuser@heavyForwarder:/opt/splunk/bin$ sudo ./splunk start --accept-license
```

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.

Create credentials for the administrator account.

Characters do not appear on the screen when you type in credentials.

```
Please enter an administrator username: sai  
Password must contain at least:
```

```
* 8 total printable ASCII character(s)
```

```
Please enter a new password:
```

```
Please confirm new password:
```

```
Copying '/opt/splunk/etc/openldap/ldap.conf.default' to '/opt/splunk/etc/openldap/ldap.conf'.  
Generating RSA private key, 2048 bit long modulus
```

```
.....+++++
```

```
Waiting for web server at http://127.0.0.1:8000 to be available..... Done
```

If you get stuck, we're here to help.  
Look for answers here: <http://docs.splunk.com>

The Splunk web interface is at <http://heavyForwarder:8000>

```
azureuser@heavyForwarder:/opt/splunk/bin$  
azureuser@heavyForwarder:/opt/splunk/bin$
```

## Add Inbound Rule for port 8000 on Heavy Forwarder VM

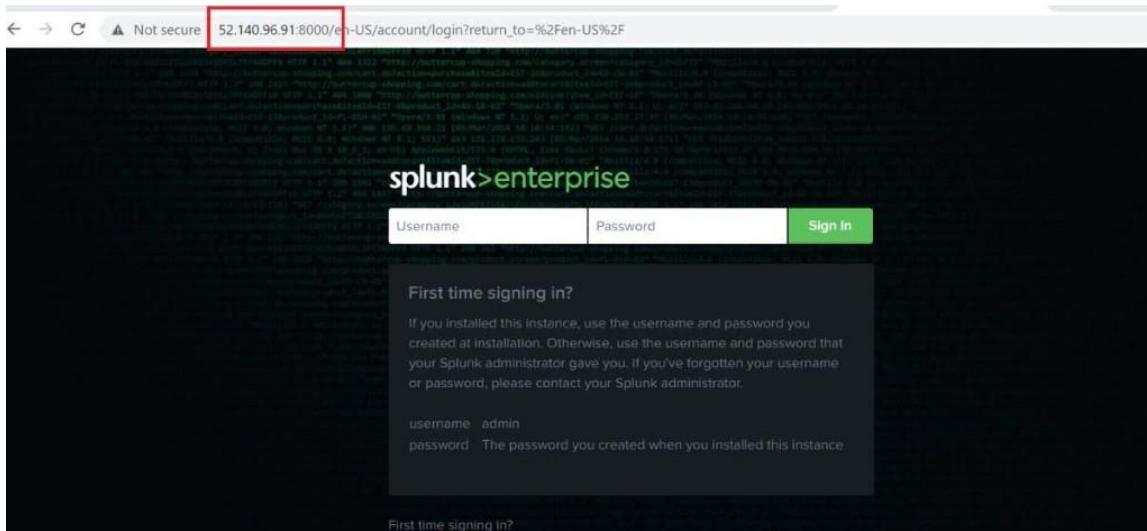
The screenshot shows the Azure portal interface for managing a virtual machine named 'heavyForwarder'. On the left, there's a list of virtual machines with 'heavyForwarder' selected. The main pane shows the 'Networking' settings for this VM. A 'Basic' security rule is being configured with the following details:

- Source:** Any
- Source port ranges:** Any
- Destination:** Any
- Destination port ranges:** 8000 (highlighted with a red box)
- Protocol:** Any (highlighted with a red box)
- Action:** Add (highlighted with a red box)

The screenshot shows the 'heavyForwarder' VM details page. The 'Networking' section displays the following information:

- Resource group: NEWGROUP
- Status: Running
- Location: Central India
- Subscription: Visual Studio Enterprise Subscription – MPN
- Subscription ID: 2d1eaf32-f707-40db-a62c-2b7d297b42a6
- Operating system: Linux (ubuntu 18.04)
- Size: Standard B1s (1 vcpus, 1 GiB mem)
- Public IP address: 52.140.96.91 (highlighted with a red box)
- Virtual network/subnet: newgroup-vnet/default
- DNS name: Configure

Open the splunk UI in browser with url: IP:8000



## Change the Licensing of the Heavy Forwarder VM to Forwarder License:

### Change license group

The type of license group determines what sorts of licenses can be used in the pools on this license server. [Learn more](#)

Enterprise license

This license adds support for multi-user and distributed deployments, alerting, role-based security, single sign-on, scheduled PDF delivery, and unlimited data volumes.

There are no valid Splunk *Enterprise* licenses installed. You will be prompted to install a license if you choose this option.

Forwarder license

Use this group when configuring Splunk as a forwarder. [Learn more](#)

Free license

Use this group when you are running Splunk Free. This license has no authentication or user and role management, and has a 500MB/day daily indexing volume. [Learn more](#)

Enterprise Trial license

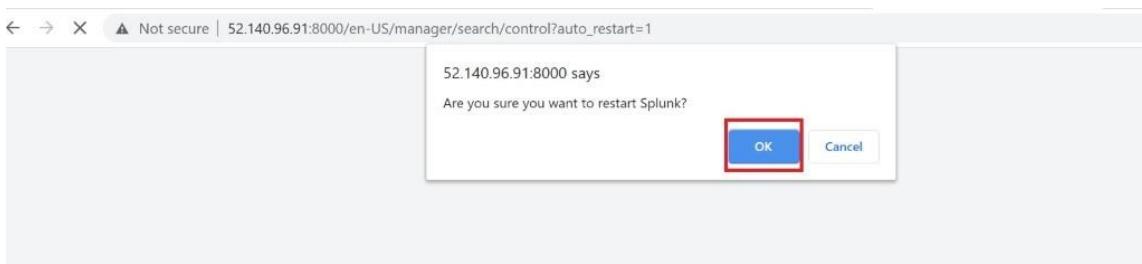
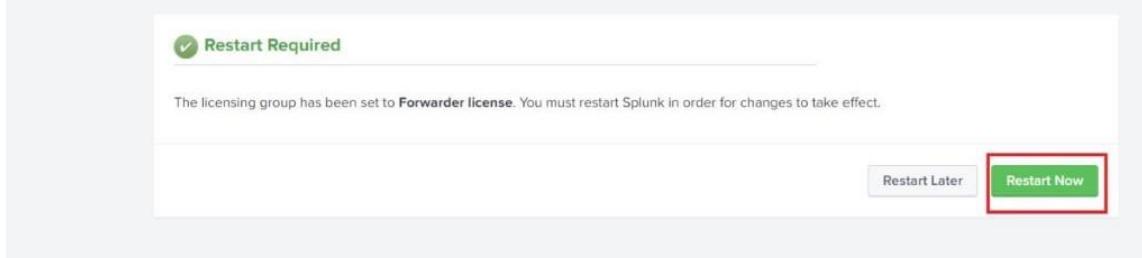
This is your included download trial. IMPORTANT: If you switch to another license, you cannot return to the Trial. You must install an Enterprise license or switch to Splunk Free.

[Cancel](#)

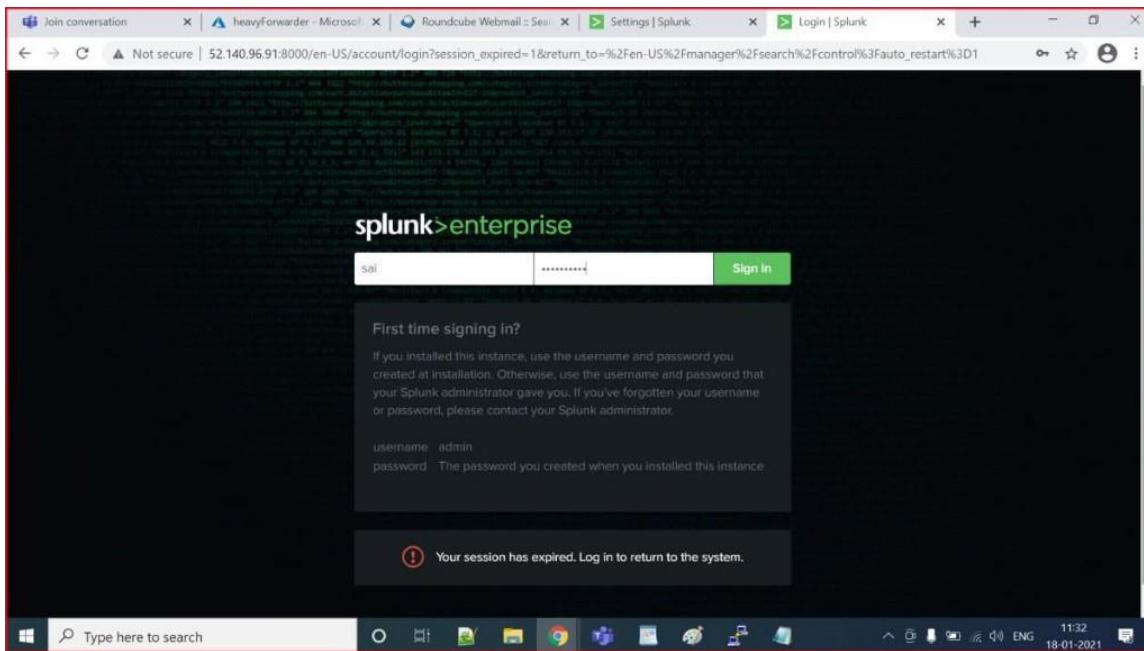
[Save](#)

## Change license group

Licensing > Change license group



## Login again to Splunk



## Configure Forwarding:

The screenshot shows the Splunk web interface with the 'Settings' menu item highlighted by a red box. The 'Forwarding and receiving' option under the 'DATA' section is also highlighted by a red box. The interface includes a sidebar with 'Add Data' and 'Monitoring Console' options, and a main content area with sections for KNOWLEDGE, DATA, SYSTEM, and DISTRIBUTED ENVIRONMENT.

**DATA**

- Data inputs
- Forwarding and receiving**
- Indexes

**DISTRIBUTED ENVIRONMENT**

- Indexer clustering
- Forwarder management
- Data Fabric
- Distributed search

**SYSTEM**

- Server settings
- Server controls
- Health report manager
- RapidDiag
- Instrumentation
- Licensing
- Workload management

**USERS AND AUTHENTICATION**

- Roles
- Users
- Tokens
- Password Management
- Authentication Methods

The screenshot shows the 'Forwarding and receiving' configuration page. It includes a 'Forward data' section with a 'Configure forwarding' button highlighted by a red box. A green button labeled 'New Forwarding Host' is also highlighted by a red box. The page displays a search bar and a message indicating no configurations exist.

**Forwarding and receiving**

**Forward data**

Forwarding and receiving > Forward data

App: All Owner: Any Visible in the App: filter 25 per page

There are no configurations of this type. Click the "New Forwarding Host" button to create a new configuration.

Open the Putty of the Splunk Indexer:

```

azureuser@splunkvm1:/opt/splunk/bin$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.11 netmask 255.255.255.0 broadcast 10.0.0.255
        inet fe80::222:48ff:fe6e:6dfd prefixlen 64 scopeid 0x20<link>
            ether 00:22:48:6e:6d:fd txqueuelen 1000 (Ethernet)
            RX packets 389670 bytes 544597991 (544.5 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 53474 bytes 13429302 (13.4 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 27789 bytes 38081582 (38.0 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 27789 bytes 38081582 (38.0 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

azureuser@splunkvm1:/opt/splunk/bin$ 

```

Copy the IP Address of the eth0 inet valu from the indexer VM

Paste the IP Address copied from Indexer node on the add Forwarder page in the Heavy Forwarder UI and port number should be 9997

IP:9997

Add new

Forwarding and receiving > Forward data > Add new

Enter host:port to forward data to. Data will be auto load balanced to each host:port.

Host \*

Set as host:port or IP:port.  
You must also enable receiving on this host.

### Forward data

Forwarding and receiving > Forward data

New Forwarding Host

Successfully saved "10.0.0.11:9997" in system.

Showing 1-1 of 1 item

App: All

Owner: Any

Visible in the App

filter

Q

25 per page

Host	Automatic Load Balancing	Status	Actions
10.0.0.11:9997	Enabled	Enabled   Disable	Clone   Delete

Add Data to be forwarder by Heavy Forwarder:

← → C Not secure | 52.140.96.91:8000/en-US/app/launcher/home

splunk>enterprise

Administrator Messages Settings Activity Help Find

Apps

> Search & Reporting

+ Find More Apps

Explore Splunk

Add Data

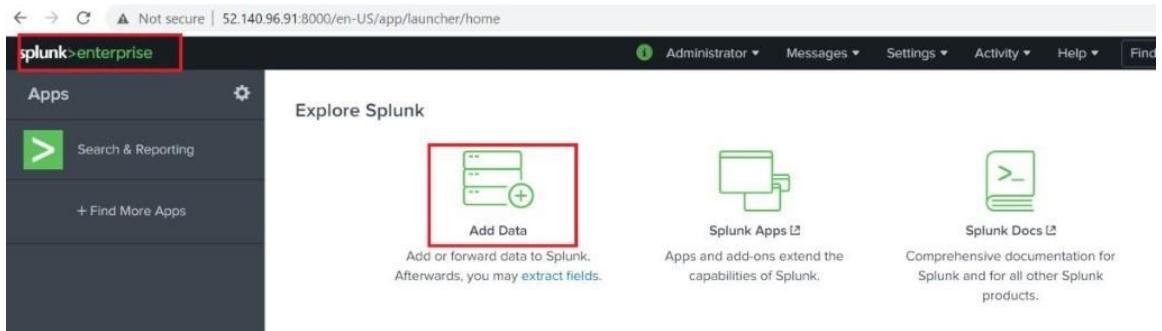
Add or forward data to Splunk. Afterwards, you may extract fields.

Splunk Apps

Apps and add-ons extend the capabilities of Splunk.

Splunk Docs

Comprehensive documentation for Splunk and for all other Splunk products.



### Or get data in with the following methods

Upload files from my computer

Local log files  
Local structured files (e.g. CSV)  
Tutorial for adding data

Monitor files and ports on this Splunk platform instance

Files - HTTP - WMI - TCP/UDP - Scripts  
Modular inputs for external data sources

Forward data from a Splunk forwarder

Files - TCP/UDP - Scripts



/manager/search/adddatamethods/selectsource?input\_mode=1

Add Data

Select Source Set Source Type Input Settings Review Done

Next >

Files & Directories

Upload a file, index a local file, or monitor an entire directory.

HTTP Event Collector

Configure tokens that clients can use to send data over HTTP or HTTPS.

TCP / UDP

Configure the Splunk platform to listen on a network port.

Scripts

Get data from any API, service, or database with a script.

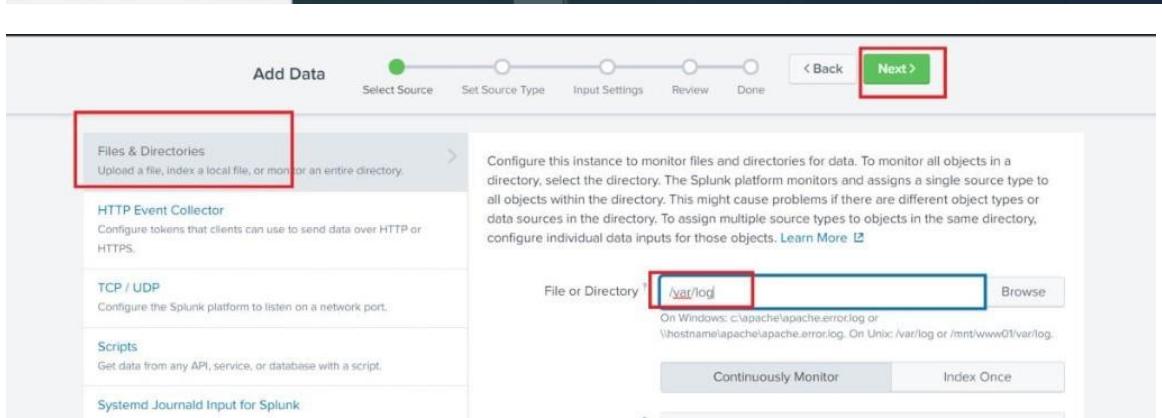
Systemd Journald Input for Splunk

Configure this instance to monitor files and directories for data. To monitor all objects in a directory, select the directory. The Splunk platform monitors and assigns a single source type to all objects within the directory. This might cause problems if there are different object types or data sources in the directory. To assign multiple source types to objects in the same directory, configure individual data inputs for those objects. Learn More

File or Directory  Browse

On Windows: c:\apache\apache.error.log or \\hostname\apache\apache.error.log. On Unix: /var/log or /mnt/www01/var/log.

Continuously Monitor Index Once



Add Data

Select Source   Input Settings   Review   Done

**App context**

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. [Learn More](#)

App Context: Search & Reporting (search) ▾

**Host**

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Host field value: heavyForwarder

Constant value   Regular expression on path   Segment in path

Add Data

Select Source   Input Settings   Review   Done

**Review**

Input Type: Directory Monitor  
 Source Path: /var/log  
 Whitelist: N/A  
 Blacklist: N/A  
 Source Type: Automatic  
 App Context: search  
 Host: heavyForwarder  
 Index: default

**Submit**

Open the Web UI of the Splunk Indexer:

splunk>enterprise

Administrator ▾   Messages ▾   Settings ▾   Activity ▾   Help ▾   Find

Apps

Search & Reporting

+ Find More Apps

Explore Splunk Enterprise

Product Tours

New to Splunk? Take a tour to help you on your way.

Add Data

Add or forward data to Splunk Enterprise. Afterwards, you may extract fields.

Explore Data

Explore data and define how Hunk parses that data.

Splunk Apps

Apps and add-ons extend the capabilities of Splunk Enterprise.

Go to Search & Reporting and Click on Data Summary

## Universal Forwarder in Splunk

The universal forwarder **collects data from a data source or another forwarder and sends it to a forwarder or a Splunk deployment**. With a universal forwarder, you can send data to Splunk Enterprise, Splunk Light, or Splunk Cloud. It also replaces the Splunk Enterprise light forwarder.

## Understanding the Universal Forwarder

Forward data without negatively impacting production performance.

	Universal Forwarder	Regular (Heavy) Forwarder
Monitor All Supported Inputs	✓	✓
Routing, Filtering, Cloning	✓	✓
Splunk Web		✓
Python Libraries		✓
Event Based Routing		✓
Scripted Inputs		✓



Copyright © 2011, Splunk Inc.  
splunk.com

28

splunk > listen to your data.

Splunk Universal Forwarder:

[https://www.splunk.com/en\\_us/download/universal-forwarder.html](https://www.splunk.com/en_us/download/universal-forwarder.html)

## Create a virtual machine

### Instance details

Virtual machine name \*  ✓

Region \*

Availability options

Image \*

[See all images](#)

Azure Spot instance

Size \*

[See all sizes](#)

### Administrator account

Home >

 **universalForwarder** ↗  
Virtual machine

🔗 Connect ▷ Start ⟲ Restart □ Stop 🕒 Capture ☒ Delete ⟳ Refresh ↗ Open in mobile

☰ Overview

🔗 Activity log

🔗 Access control (IAM)

🔗 Tags

🔗 Diagnose and solve problems

Settings

🔗 Networking

🔗 Connect

#### Essentials

Resource group [\(change\)](#)  
**newgroup**

Operating system  
Linux

Status

Running

Location

Central India

Subscription [\(change\)](#)  
Visual Studio Enterprise Subscription – MPN

Size

Standard B1s (1 vcpus, 1 GiB memory)

Public IP address

**20.193.224.38**

Virtual network/subnet  
newgroup-vnet/default

DNS name

[Configure](#)

```
azureuser@universalForwarder:~$ System information as of Mon Jan 18 07:28:11 UTC 2021
System load: 0.56          Processes: 114
Usage of /: 4.5% of 28.90GB  Users logged in: 0
Memory usage: 20%          IP address for eth0: 10.0.0.13
Swap usage: 0%
0 packages can be updated.
0 of these updates are security updates.

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

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

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
azureuser@universalForwarder:~$
```

```
cd $HOME
```

```
wget -O splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz
'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.0&product=universalforwarder&filename=splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz&wget=true'
```

```
azureuser@universalForwarder:~$ See "man sudo_root" for details.

azureuser@universalForwarder:~$ wget -O splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.0&product=universalforwarder&filename=splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz&wget=true'
--2021-01-18 07:29:35-- https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.0&product=universalforwarder&filename=splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz&wget=true
Resolving www.splunk.com (www.splunk.com)... 96.17.150.163, 96.17.150.155
Connecting to www.splunk.com (www.splunk.com)|96.17.150.163|:443... connected
```

```
tar -xvf splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz
```

```
2021-01-10 07:29:45 (0.39 MB/s) - 'splunkforwarder-8.1.0-Linux-x86_64.tgz' saved [43024324/43024324]

azureuser@universalForwarder:~$ tar -xvf splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz
splunkforwarder/
splunkforwarder/etc/
splunkforwarder/etc/disabled-apps/
splunkforwarder/etc/disabled-apps/README
splunkforwarder/etc/log-dfs-search.cfg
splunkforwarder/etc/datetime.xml
splunkforwarder/etc/prettyprint.xsl
splunkforwarder/etc/copyright.txt
```

```
sudo mv splunkforwarder /opt/splunkforwarder
```

```
splunkforwarder/swidtag/splunk-UniversalForwarder-primary.swidtag
azureuser@universalForwarder:~$ sudo mv splunkforwarder /opt/splunkforwarder
azureuser@universalForwarder:~$ ls
splunkforwarder-8.1.0-f57c09e87251-Linux-x86_64.tgz
azureuser@universalForwarder:~$ ls /opt
splunkforwarder
azureuser@universalForwarder:~$
```

```
cd /opt/splunkforwarder/bin
```

```
azureuser@universalForwarder:~$ cd /opt/splunkforwarder/bin
azureuser@universalForwarder:/opt/splunkforwarder/bin$ 
azureuser@universalForwarder:/opt/splunkforwarder/bin$ ls
```

```
sudo ./splunk start --accept-license
```

```
az user@universalForwarder:/opt/splunkforwarder/bin
classify      openssl          priptopng  slim
copyright.txt pid_check.sh    pripngsch  splunk
accutil       nrchunkmgr     priptopam  splunkd
azureuser@universalForwarder:/opt/splunkforwarder/bin$ sudo ./splunk start --accept-license
```

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.

Create credentials for the administrator account.

Characters do not appear on the screen when you type in credentials.

Please enter an administrator username:

Password must contain at least:

\* 8 total printable ASCII character(s).

Please enter a new password:

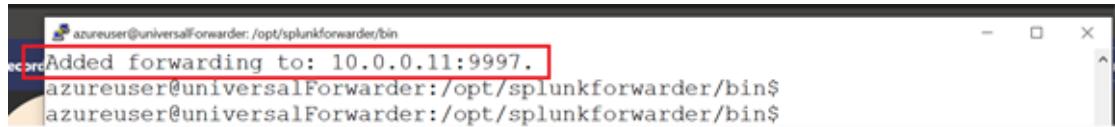
Please confirm new password:

Splunk> Be an IT superhero. Go home early.

Checking prerequisites...

Checking mmt port 180891: open

```
sudo ./splunk add forward-server 10.0.0.11:9997 -auth sai:newpassword
```



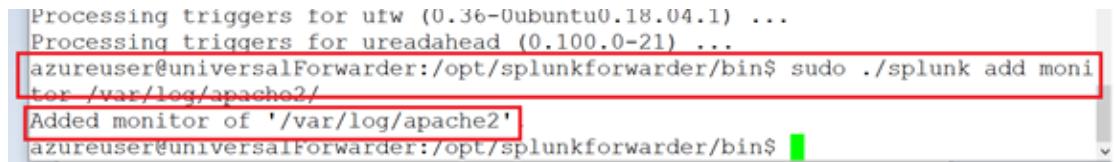
```
azureuser@universalForwarder:/opt/splunkforwarder/bin$ core Added forwarding to: 10.0.0.11:9997.  
azureuser@universalForwarder:/opt/splunkforwarder/bin$ azureuser@universalForwarder:/opt/splunkforwarder/bin$
```

Added forwarding to: 10.0.0.11:9997.

`sudo apt-get update`

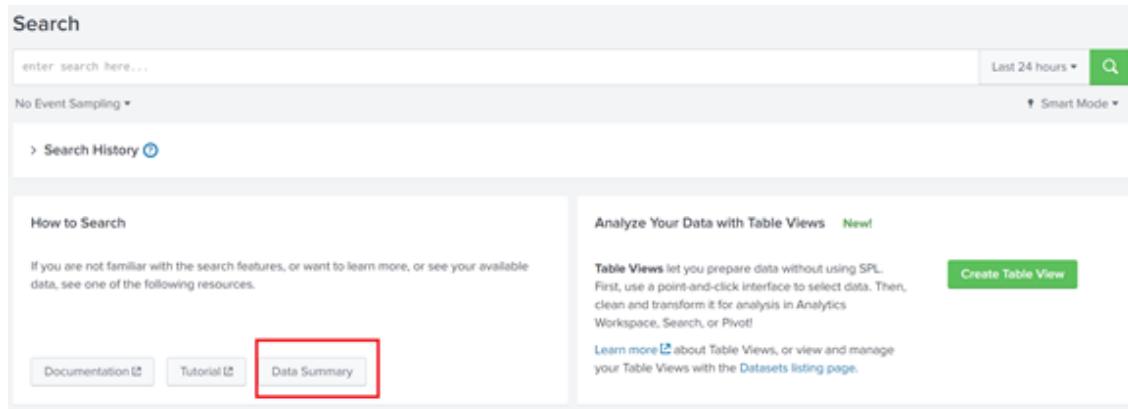
`sudo apt-get install apache2 -y`

`sudo ./splunk add monitor /var/log/apache2/`



```
Processing triggers for utw (0.36-0ubuntu0.18.04.1) ...  
Processing triggers for ureadahead (0.100.0-21) ...  
azureuser@universalForwarder:/opt/splunkforwarder/bin$ sudo ./splunk add monitor /var/log/apache2/  
Added monitor of '/var/log/apache2'  
azureuser@universalForwarder:/opt/splunkforwarder/bin$
```

Open the Search & Reporting of the Splunk Indexer Node:



The screenshot shows the Splunk search interface. At the top, there is a search bar with placeholder text "enter search here...". To the right of the search bar are filters for "Last 24 hours" and a magnifying glass icon. Below the search bar, there is a "No Event Sampling" button and a "Smart Mode" dropdown. A "Search History" link is also present. The main content area is divided into two sections: "How to Search" on the left and "Analyze Your Data with Table Views" on the right. Under "How to Search", there is a note about search features and links to "Documentation", "Tutorial", and "Data Summary". The "Data Summary" link is highlighted with a red box. On the right, there is information about "Table Views" and a "Create Table View" button. A note below the table view section says "Learn more about Table Views, or view and manage your Table Views with the Datasets listing page."

Data Summary X

Hosts (2) Sources (13) Sourcetypes (12)

filter Q

Host	Count	Last Update
heavyForwarder	3,945	1/18/21 7:23:56.000 AM
universalForwarder	2	1/18/21 7:44:42.000 AM

On the Putty of Universal Forwarder Node:

```
curl localhost
```

The screenshot shows a terminal window with the following text:  
azureuser@universalForwarder:/opt/splunkforwarder/bin\$ curl localhost

On the Web UI of the Indexer Node;

Data Summary X

Hosts (2) Sources (14) Sourcetypes (13)

filter Q

Host	Count	Last Update
heavyForwarder	3,945	1/18/21 7:23:56.000 AM
universalForwarder	8	1/18/21 7:47:39.000 AM

New Search

host=universalForwarder

Last 24 hours

8 events (1/17/21 7:00:00.000 AM to 1/18/21 7:49:40.000 AM) No Event Sampling Job Smart

Events (8) Patterns Statistics Visualization

Format Timeline - Zoom Out + Zoom to Selection Deselect 1 hour

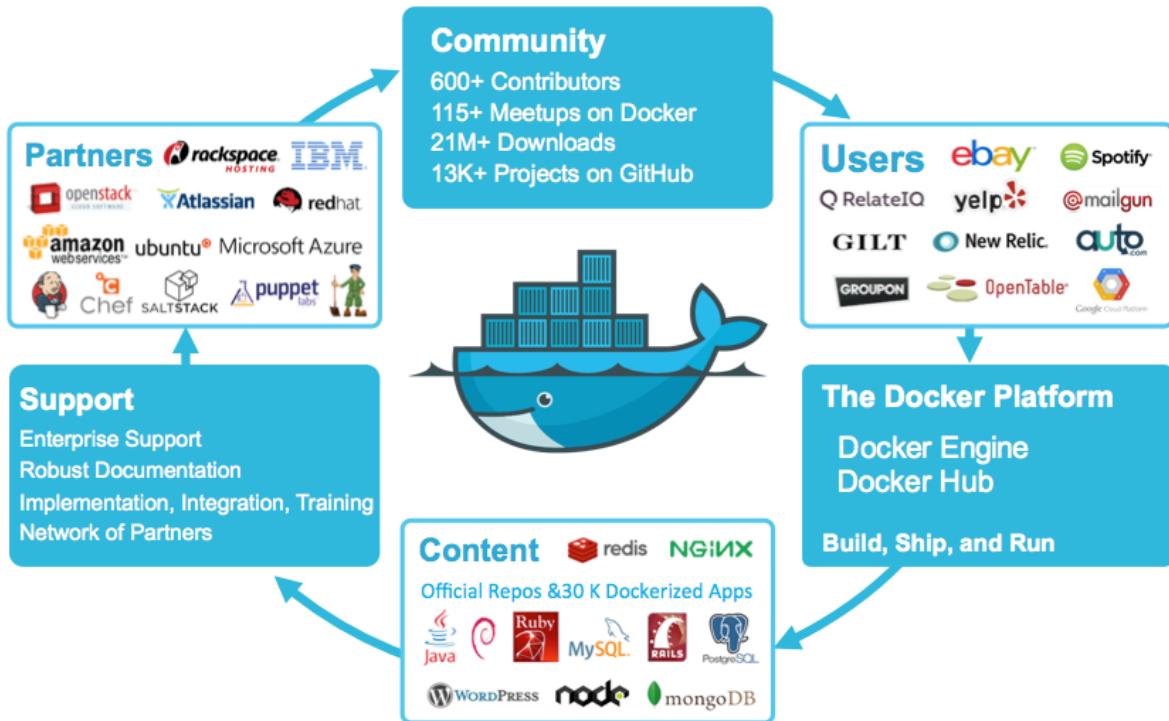
	Time	Event
>	1/18/21 7:47:39.000 AM	127.0.0.1 - - [18/Jan/2021:07:47:39 +0000] "GET / HTTP/1.1" 200 11173 "-" "curl/7.58.0"
>	1/18/21 7:47:38.000 AM	host = universalForwarder source = /var/log/apache2/access.log sourcetype = access-too_small
>	1/18/21 7:47:37.000 AM	127.0.0.1 - - [18/Jan/2021:07:47:37 +0000] "GET / HTTP/1.1" 200 11173 "-" "curl/7.58.0"
>	1/18/21	host = universalForwarder source = /var/log/apache2/access.log sourcetype = access-too_small
>	1/18/21	127.0.0.1 - - [18/Jan/2021:07:47:37 +0000] "GET / HTTP/1.1" 200 11173 "-" "curl/7.58.0"

Splunk HeavyForw...pdf Splunk HeavyForw...pdf

## DOCKER

Docker is a container management service. The keywords of Docker are **developed**, **ship** and **run** anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

The initial release of Docker was in March 2013 and since then, it has become the buzzword for modern world development, especially in the face of Agile-based projects.



## Features of Docker

- ★ Docker has the ability to reduce the size of development by providing a smaller footprint of the operating system via containers.
- ★ With containers, it becomes easier for teams across different units, such as development, QA and Operations to work seamlessly across applications.
- ★ You can deploy Docker containers anywhere, on any physical and virtual machines and even on the cloud.
- ★ Since Docker containers are pretty lightweight, they are very easily scalable.

## Components of Docker

Docker has the following components

- ★ **Docker for Mac** – It allows one to run Docker containers on the Mac OS.
- ★ **Docker for Linux** – It allows one to run Docker containers on the Linux OS.
- ★ **Docker for Windows** – It allows one to run Docker containers on the Windows OS.
- ★ **Docker Engine** – It is used for building Docker images and creating Docker containers.
- ★ **Docker Hub** – This is the registry which is used to host various Docker images.

- ★ **Docker Compose** – This is used to define applications using multiple Docker containers.

## Installing Docker on Ubuntu Machine

Create a VM with size as Standard B2s

### Create a virtual machine

Instance details

Virtual machine name *	dockervm
Region *	(Asia Pacific) Central India
Availability options	No infrastructure redundancy required
Image *	Ubuntu Server 18.04 LTS - Gen1 <a href="#">See all images</a>
Azure Spot instance	<input type="checkbox"/>
Size *	Standard_B2s - 2 vcpus, 4 GiB memory (₹2,161.61/month) <a href="#">See all sizes</a>

```
sudo apt-get update
```

```
sudo apt-get install -y \
```

```
    apt-transport-https \
```

```
    ca-certificates \
```

```
    curl \
```

```
    gnupg-agent \
```

```
    software-properties-common
```

```
Reading package lists... Done
azureuser@dockervm:~$ sudo apt-get install -y \
>     apt-transport-https \
>     ca-certificates \
>     curl \
>     gnupg-agent \
>     software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
Setting up gnupg-agent (2.2.4-1ubuntu1.3) ...
azureuser@dockervm:~$ curl -fsSL https://download.docker.com/linux/ubuntu/
gpg | sudo apt-key add -
OK
azureuser@dockervm:~$ sudo add-apt-repository \
>     "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
>     $(lsb_release -cs) \
>     stable"
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease
```

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io -y
```

```
azureuser@dockervm:~$ 
azureuser@dockervm:~$ sudo apt-get install docker-ce docker-ce-cli contain
erd.io -y
Reading package lists... Done
Building dependency tree

```

```
//sudo gpasswd -a userid docker      #azureuser is the current logged in user
```

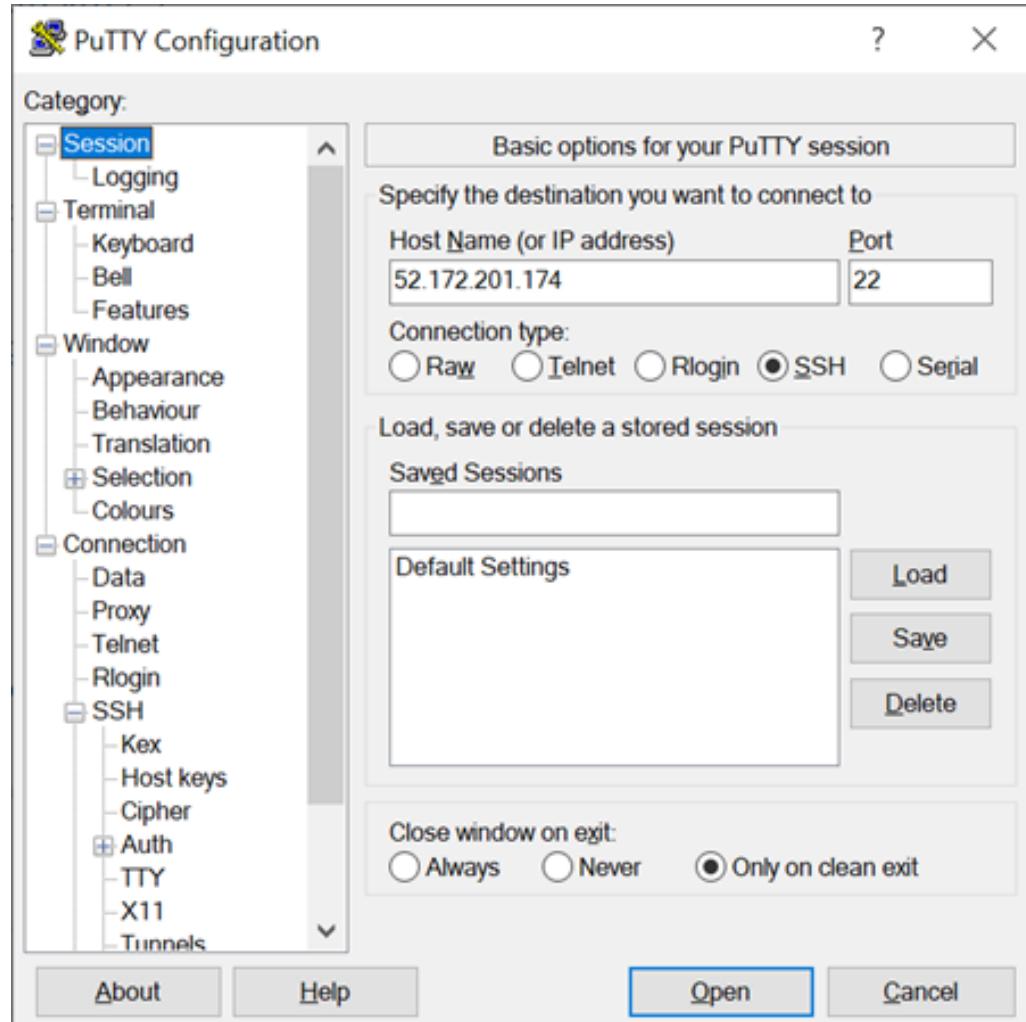
```
sudo gpasswd -a azureuser docker
```

```
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
azureuser@dockervm:~$ sudo gpasswd -a azureuser docker
Adding user azureuser to group docker
```

exit

```
Adding user azureuser to group docker
azureuser@dockervm:~$ exit
```

Relaunch Putty & Connect to the same VM:



docker run hello-world : Creates an image Hello-world in the docker

```
Last login: Wed Jan 20 07:32:56 2021 from 122.178.134.252
azureuser@dockervm:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:31b9c7d48790f0d8c50ab433d9c3b7e17666d6993084c002c2ff1ca
09b96391d
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
```

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

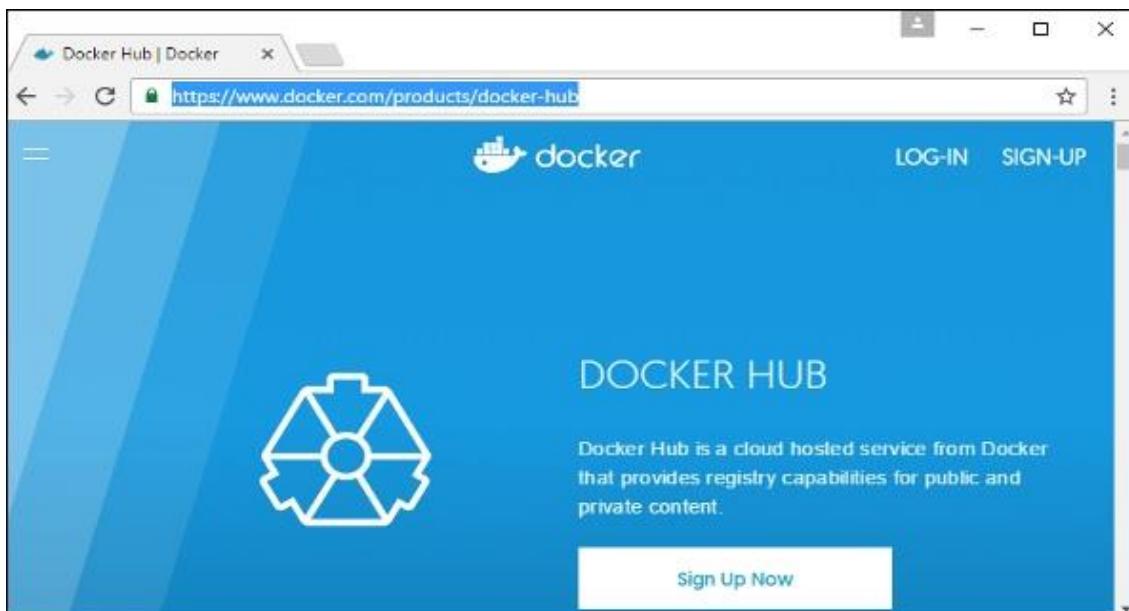
```
$ docker run -it ubuntu bash
```

## Docker Hub

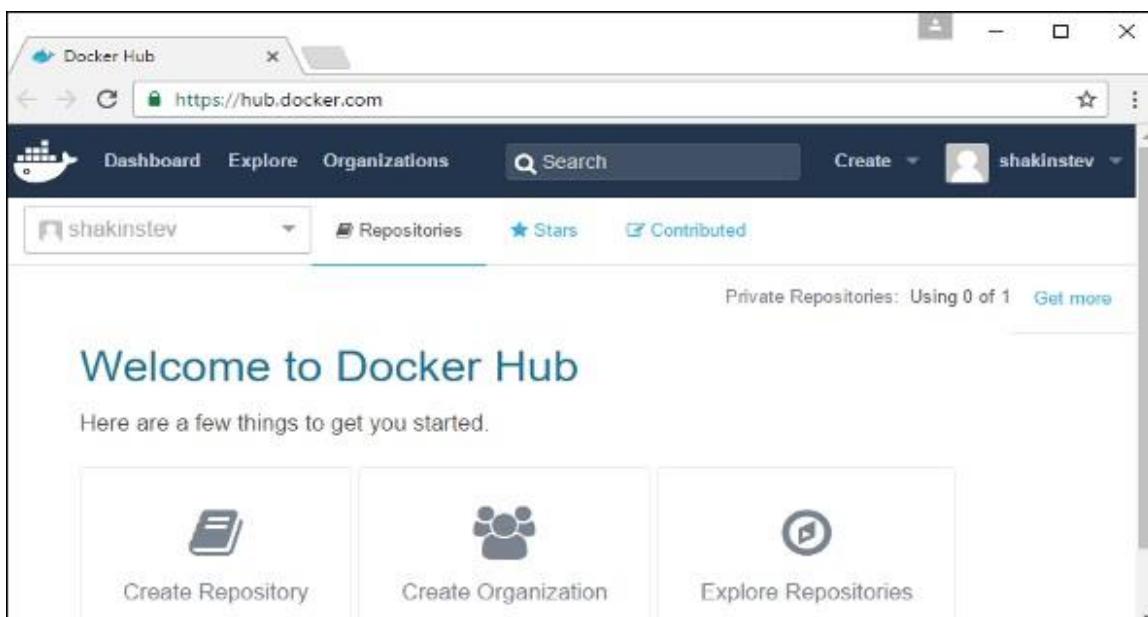
Docker Hub is a registry service on the cloud that allows you to download Docker images that are built by other communities. You can also upload your own Docker built images to Docker hub. In this chapter, we will see how to download and use the Jenkins Docker image from Docker hub.

The official site for Docker hub is – [https://www.docker.com/community-edition#/add\\_ons](https://www.docker.com/community-edition#/add_ons)

**Step 1** – First you need to do a simple sign-up on Docker hub.



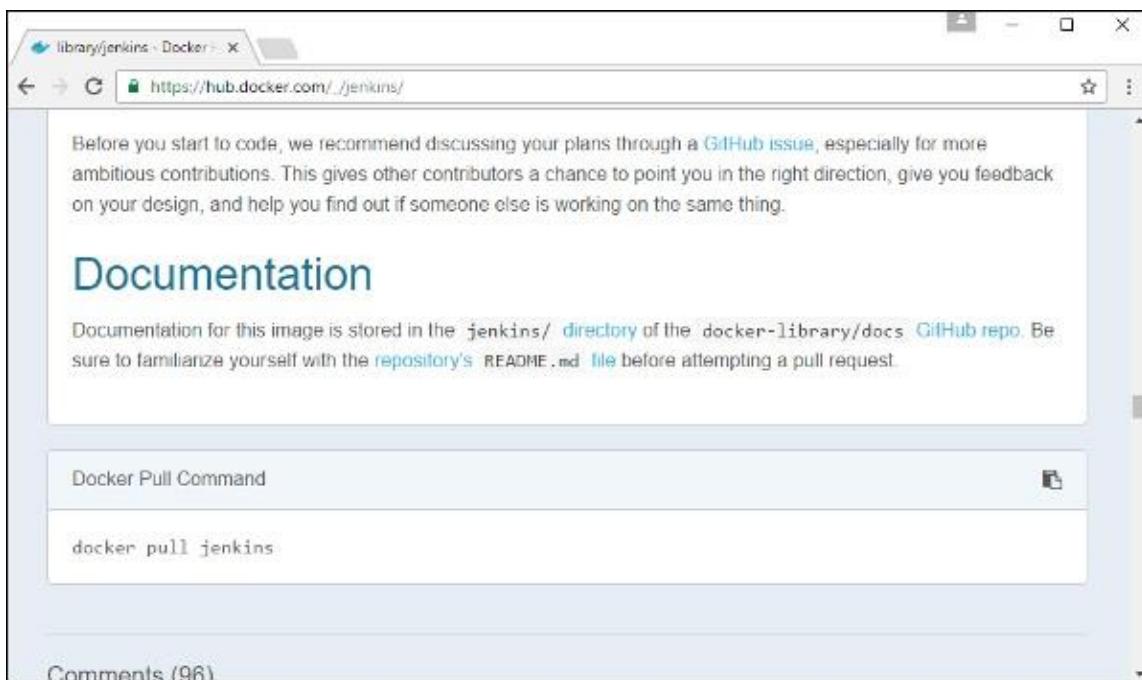
**Step 2** – Once you have signed up, you will be logged into Docker Hub.



**Step 3** – Next, let's browse and find the Jenkins image.



**Step 4** – If you scroll down on the same page, you can see the Docker **pull** command. This will be used to download the Jenkins image onto the local Ubuntu server.



**Step 5** – Now, go to the Ubuntu server and run the following command –

```
sudo docker pull jenkins
```

```
a079defbaeff: Pull complete
66181a89effa: Pull complete
f4d8f7d94b9c: Pull complete
98e5c3e08215: Pull complete
992fde8f3336: Pull complete
65b58e072756: Pull complete
0b0b6d6525a1: Pull complete
4e7171e4505a: Pull complete
469745638476: Pull complete
49d5aaafff78: Pull complete
c01281524fd6: Pull complete
00a759703a0b: Pull complete
da411a858795: Pull complete
7b8a0b4fd7d0: Pull complete
cbd9e145ea6b: Pull complete
700f8f527cd7: Pull complete
88d27231965c: Pull complete
a067af206313: Pull complete
211049e028a4: Pull complete
7249723069d8: Pull complete
6465c437f020: Pull complete
954c67861e66: Pull complete
6a14c8afbb3a: Pull complete
ec070f7e511e: Pull complete
983246da862f: Pull complete
998d1854867e: Pull complete
Digest: sha256:878e055f96c90af9281fd859f7c69ac289e0178594ff36bbb85e53b78969
Status: Downloaded newer image for jenkins:latest
demo@ubuntuserver:~$
```

To run Jenkins, you need to run the following command –

```
sudo docker run -p 8080:8080 -p 50000:50000 jenkins
```

Note the following points about the above **sudo** command –

- ★ We are using the **sudo** command to ensure it runs with root access.
- ★ Here, **jenkins** is the name of the image we want to download from Docker hub and install on our Ubuntu machine.
- ★ **-p** is used to map the port number of the internal Docker image to our main Ubuntu server so that we can access the container accordingly.

```
*****
***** Jenkins initial setup is required. An admin user has been created and a password
***** generated.
***** Please use the following password to proceed to installation:
***** 69a504bd19634390b4e67fdd0a908e67
***** This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
*****

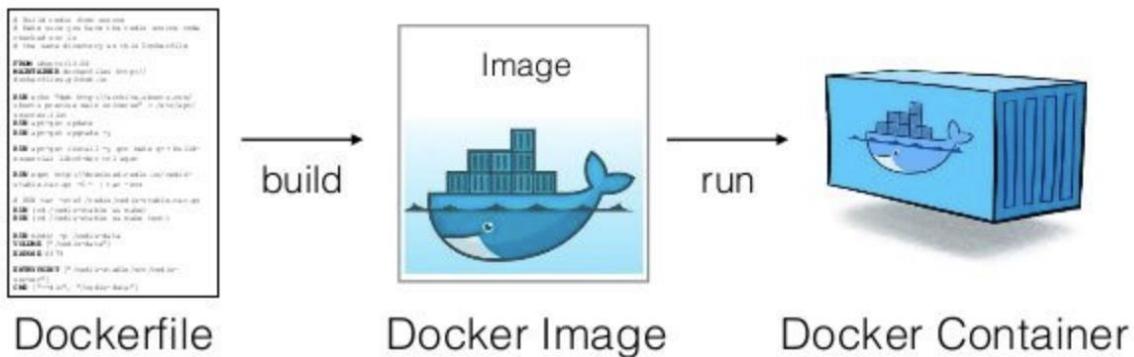


--> setting agent port for jnlp
--> setting agent port for jnlp... done
Dec 01, 2016 8:16:21 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Dec 01, 2016 8:16:22 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Dec 01, 2016 8:16:22 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Dec 01, 2016 8:16:22 PM hudson.WebAppMain$3 run
INFO: Jenkins is fully up and running
Dec 01, 2016 8:16:25 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tools.JDKInstaller
Dec 01, 2016 8:16:25 PM hudson.model.AsyncPeriodicWork$1 run
INFO: Finished Download metadata. 18,218 ms
```

You will then have Jenkins successfully running as a container on the Ubuntu machine

## Docker-Images

In Docker, everything is based on Images. An image is a combination of a file system and parameters. Let's take an example of the following command in Docker.



`docker run hello-world`

- ★ The Docker command is specific and tells the Docker program on the Operating System that something needs to be done.
- ★ The **run** command is used to mention that we want to create an instance of an image, which is then called a **container**.
- ★ Finally, "hello-world" represents the image from which the container is made.

Now let's look at how we can use the CentOS image available in Docker Hub to run CentOS on our Ubuntu machine. We can do this by executing the following command on our Ubuntu machine –

`sudo docker run -it centos /bin/bash`

Note the following points about the above **sudo** command –

- ★ We are using the **sudo** command to ensure that it runs with **root** access.
- ★ Here, **centos** is the name of the image we want to download from Docker Hub and install on our Ubuntu machine.
- ★ **-it** is used to mention that we want to run in **interactive mode**.
- ★ **/bin/bash** is used to run the bash shell once CentOS is up and running.

### Displaying Docker Images

To see the list of Docker images on the system, you can issue the following command.

`docker images`

This command is used to display all the images currently installed on the system.

### Example

**sudo docker images**

### Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker images
[sudo] password for demo:
REPOSITORY          TAG      IMAGE ID      CREATED
VIRTUAL SIZE
newcentos           latest   7a86f8ffcb25    9 days ago
196.5 MB
jenkins             latest   998d1854867e    2 weeks ago
714.1 MB
centos              latest   97cad5e16cb6    4 weeks ago
196.5 MB
demo@ubuntuserver:~$ _
```

From the above output, you can see that the server has three images: **centos**, **newcentos**, and **jenkins**. Each image has the following attributes –

- ★ **TAG** – This is used to logically tag images.
- ★ **Image ID** – This is used to uniquely identify the image.
- ★ **Created** – The number of days since the image was created.
- ★ **Virtual Size** – The size of the image.

### Downloading Docker Images

Images can be downloaded from Docker Hub using the Docker **run** command. Let's see in detail how we can do this.

### Syntax

The following syntax is used to run a command in a Docker container.

### docker run image

### Example

**sudo docker run centos**

This command will download the **centos** image, if it is not already present, and run the OS as a container.

### Output

When we run the above command, we will get the following result –

```
demo@ubuntuserver:~$ sudo docker run centos
Unable to find image 'centos:latest' locally
latest: Pulling from centos

3690474eb5b4: Pull complete
af0819ed1fac: Pull complete
05fe84bf6d3f: Pull complete
97cad5e16cb6: Pull complete
Digest: sha256:934ff980b04db1b7484595bac0c8e6f838e1917ad3a38f904ece64f70bbc
Status: Downloaded newer image for centos:latest
demo@ubuntuserver:~$ _
```

You will now see the CentOS Docker image downloaded. Now, if we run the Docker **images** command to see the list of images on the system, we should be able to see the **centos** image as well.

```
demo@ubuntuserver:~$ sudo docker run centos
Unable to find image 'centos:latest' locally
latest: Pulling from centos

3690474eb5b4: Pull complete
af0819ed1fac: Pull complete
05fe84bf6d3f: Pull complete
97cad5e16cb6: Pull complete
Digest: sha256:934ff980b04db1b7484595bac0c8e6f838e1917ad3a38f904ece64f70bbc
Status: Downloaded newer image for centos:latest
demo@ubuntuserver:~$ sudo docker images
REPOSITORY          TAG           IMAGE ID            CREATED             VIRTUAL SIZE
jenkins              latest        998d1854867e      2 weeks ago         714.1 MB
centos               latest        97cad5e16cb6      4 weeks ago        196.5 MB
demo@ubuntuserver:~$
```

## Removing Docker Images

The Docker images on the system can be removed via the **docker rmi** command. Let's look at this command in more detail.

**docker rmi**

This command is used to remove Docker images.

### Syntax

**docker rmi ImageID**

### Example

**sudo docker rmi 7a86f8ffcb25**

Here, **7a86f8ffcb25** is the Image ID of the **newcentos** image.

## **Output**

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker rmi 7a86f8ffcb25
Untagged: newcentos:latest
Deleted: 7a86f8ffcb258e42c11d971a04b1145151b80122e566bc2b544f8fc3f94caf 1e
demo@ubuntuserver:~$
```

Let's see some more Docker commands on images.

**docker images -q**

This command is used to return only the Image ID's of the images.

## **Syntax**

docker images

## **Example**

**sudo docker images -q**

## **Output**

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker images -q
998d1854867e
97cad5e16cb6
demo@ubuntuserver:~$ _
```

## **docker inspect**

This command is used see the details of an image or container.

## **Syntax**

**docker inspect Repository**

## **Example**

**sudo docker inspect jenkins**

## **Output**

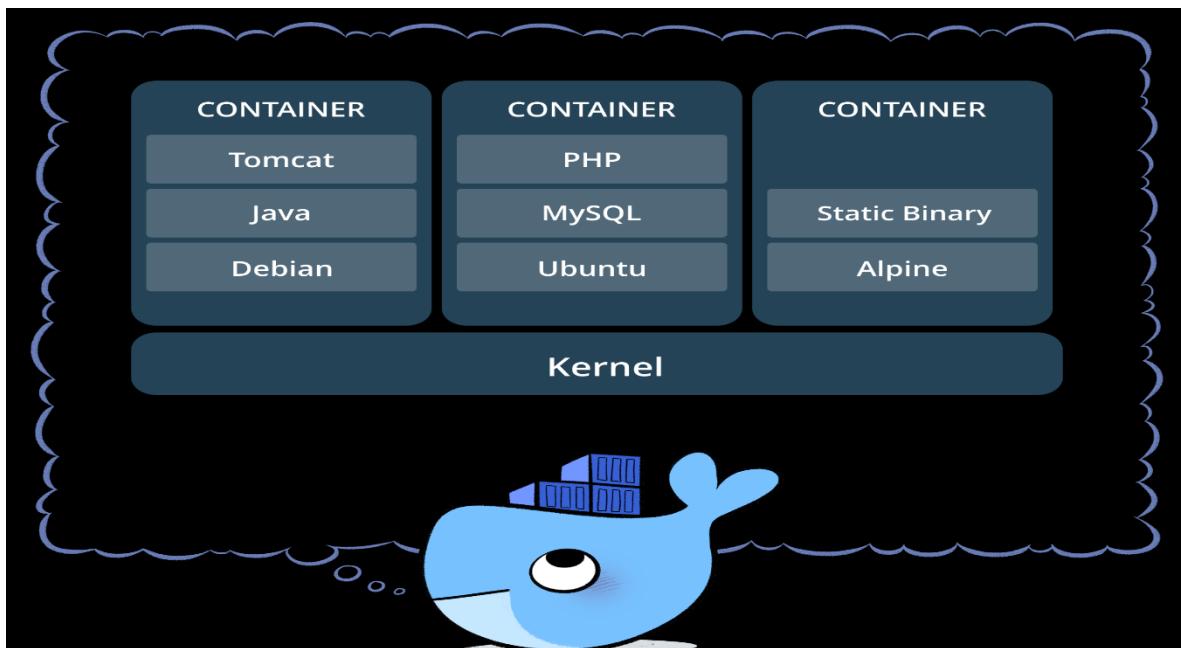
When we run the above command, it will produce the following result –

```

    "Hostname": "6b3797ab1e90",
    "Image": "sha256:532b1ef702484a402708f3b65a61e6ddf307bbf2fdfa01be55
a7678ce6c",
    "Labels": {},
    "MacAddress": "",
    "Memory": 0,
    "MemorySwap": 0,
    "NetworkDisabled": false,
    "OnBuild": [],
    "OpenStdin": false,
    "PortSpecs": null,
    "StdinOnce": false,
    "Tty": false,
    "User": "jenkins",
    "Volumes": {
        "/var/jenkins_home": {}
    },
    "WorkingDir": ""
},
"Created": "2016-11-16T20:52:37.568557509Z",
"DockerVersion": "1.12.3",
"Id": "998d1854867eb7873a9f45ff4c3ab25bcf5378c77fc955d344e47cb27e5df723
",
"Os": "linux",
"Parent": "983246da862f43a967b36cc2fc1af580df3f79760dfd841c1954e7325301
",
"Size": 5960,
"VirtualSize": 714121162
}
]
demo@ubuntuserver:~$
```

## Docker Containers

Containers are instances of Docker images that can be run using the Docker run command. The basic purpose of Docker is to run containers.



## **Running a Container**

Running of containers is managed with the Docker **run** command. To run a container in an interactive mode, first launch the Docker container.

```
sudo docker run -it centos /bin/bash
```

Then hit Crtl+p+q and you will return to your OS shell.

```
demo@ubuntuserver:~$ sudo docker run -it centos /bin/bash
[root@9f215ed0b0d3 ~]#
```

You will then be running in the instance of the CentOS system on the Ubuntu server.

## **Listing of Containers**

One can list all of the containers on the machine via the **docker ps** command. This command is used to return the currently running containers.

```
docker ps
```

### **Syntax**

```
docker ps .
```

### **Example**

```
sudo docker ps
```

### **Output**

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
9f215ed0b0d3        centos:latest      "/bin/bash"         About a minute ago   Up About a minute   cocky_colden
demo@ubuntuserver:~$
```

Let's see some more variations of the **docker ps** command.

## **docker ps -a**

This command is used to list all of the containers on the system

### **Syntax**

```
docker ps -a
```

## Example

```
sudo docker ps -a
```

## Output

When we run the above command, it will produce the following result –

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS		PORTS	
NAMES			
9f215ed0b0d3	centos:latest	"/bin/bash"	4 minutes ago
Up 4 minutes			
cocky_colden			
e5a02936065a	centos:latest	"/bin/bash"	39 minutes ago
Exited (0) 39 minutes ago			
ecstatic_hodgkin			
9b286dd1f16a	jenkins:latest	"/bin/tini -- /usr/l	18 hours ago
Exited (0) About an hour ago	0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000		
cp_jolly_wright			
3646aa260a2d	jenkins:latest	"/bin/tini -- /usr/l	9 days ago
Exited (0) 9 days ago	0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000		
cp_reverent_morse			
demo@ubuntuserver:~\$			

## docker history

With this command, you can see all the commands that were run with an image via a container.

## Syntax

```
docker history ImageID
```

## Example

```
sudo docker history centos
```

The above command will show all the commands that were run against the **centos** image.

## Output

When we run the above command, it will produce the following result –

```

demo@ubuntuserver:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
VIRTUAL SIZE
jenkins            latest   998d1854867e    2 weeks ago
714.1 MB
centos              latest   97cad5e16cb6    4 weeks ago
196.5 MB
demo@ubuntuserver:~$ sudo docker history centos
IMAGE           CREATED      CREATED BY
SIZE
97cad5e16cb6    4 weeks ago   /bin/sh -c #(nop)  CMD ["/bin/bash"]
0 B
05fe84bf6d3f    4 weeks ago   /bin/sh -c #(nop)  LABEL name=CentOS B
e Ima 0 B
af0819ed1fac    4 weeks ago   /bin/sh -c #(nop) ADD file:54df3580ac9
66389 196.5 MB
3690474eb5b4    3 months ago  /bin/sh -c #(nop)  MAINTAINER https://
thub. 0 B
demo@ubuntuserver:~$ _

```

## **docker top**

With this command, you can see the top processes within a container.

### Syntax

**docker top ContainerID**

### Example

**sudo docker top 9f215ed0b0d3**

The above command will show the top-level processes within a container.

### Output

When we run the above command, it will produce the following result –

```

demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
STATUS              PORTS              NAMES
9f215ed0b0d3        centos:latest      "/bin/bash"        12 minutes ago   Up 12 minutes       cocky_colden
demo@ubuntuserver:~$ sudo docker top 9f215ed0b0d3
JID                  PID                PPID               C
STIME                TTY                TIME               CMD
root                1606               678                0
18:13                pts/0              00:00:00          /bin/bash
demo@ubuntuserver:~$ 

```

## **docker stop**

This command is used to stop a running container.

### Syntax

**docker stop ContainerID**

### Example

```
sudo docker stop 9f215ed0b0d3
```

The above command will stop the Docker container **9f215ed0b0d3**.

### Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
9f215ed0b0d3        centos:latest      "/bin/bash"        22 minutes ago   Up 22 minutes       cocky_colden
demo@ubuntuserver:~$ sudo docker stop 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ sudo docker rm 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ _
```

## docker rm

This command is used to delete a container.

### Syntax

```
docker rm ContainerID
```

### Example

```
sudo docker rm 9f215ed0b0d3
```

The above command will remove the Docker container **9f215ed0b0d3**.

### Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
9f215ed0b0d3        centos:latest      "/bin/bash"        22 minutes ago   Up 22 minutes       cocky_colden
demo@ubuntuserver:~$ sudo docker stop 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ sudo docker rm 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ _
```

## docker stats

This command is used to provide the statistics of a running container.

### Syntax

```
docker stats ContainerID
```

## Options

- ★ **ContainerID** – This is the Container ID for which the stats need to be provided.

## Return Value

The output will show the CPU and Memory utilization of the Container.

### Example

```
sudo docker stats 9f215ed0b0d3
```

The above command will provide CPU and memory utilization of the Container **9f215ed0b0d3**.

### Output

When we run the above command, it will produce the following result –

CONTAINER	CPU %	MEM USAGE/LIMIT	MEM %
NET I/O			
07b0b6f434fe	0.00%	416 KiB / 1.416 GiB	0.03%
648 B / 648 B			

## docker attach

This command is used to attach to a running container.

### Syntax

```
docker attach ContainerID
```

### Example

```
sudo docker attach 07b0b6f434fe
```

The above command will attach to the Docker container **07b0b6f434fe**.

## Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
07b0b6f434fe        centos:latest      "/bin/bash"        3 minutes ago     Up 3 minutes       cocky_pare
demo@ubuntuserver:~$ sudo docker attach 07b0b6f434fe
[root@07b0b6f434fe ~]# _
```

Once you have attached to the Docker container, you can run the above command to see the process utilization in that Docker container.

```

top - 15:24:06 up 2:06, 0 users, load average: 0.00, 0.01, 0.02
Tasks: 2 total, 1 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1484856 total, 1057152 free, 52368 used, 375336 buff/cache
KiB Swap: 1519612 total, 1519612 free, 0 used. 1403868 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 1 root    20   0 11784 2992 2644 S 0.0 0.2 0:00.01 bash
 15 root   20   0 51864 3772 3272 R 0.0 0.3 0:00.00 top

```

## **docker pause**

This command is used to pause the processes in a running container.

### Syntax

docker pause ContainerID

### Example

`sudo docker pause 07b0b6f434fe`

The above command will pause the processes in a running container **07b0b6f434fe**.

### Output

When we run the above command, it will produce the following result –

```

demo@ubuntuserver:~$ sudo docker ps
[sudo] password for demo:
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
07b0b6f434fe        centos:latest      "/bin/bash"        18 minutes ago   Up 18 minutes     cocky_pare
demo@ubuntuserver:~$ sudo docker pause 07b0b6f434fe
07b0b6f434fe
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
07b0b6f434fe        centos:latest      "/bin/bash"        19 minutes ago   Up 19 minutes (Paused)   cocky_pare
demo@ubuntuserver:~$ 

```

## docker unpause

This command is used to **unpause** the processes in a running container.

### Syntax

`docker unpause ContainerID`

### Example

`sudo docker unpause 07b0b6f434fe`

The above command will unpause the processes in a running container: 07b0b6f434fe

## Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker unpause 07b0b6f434fe  
07b0b6f434fe  
demo@ubuntuserver:~$
```

## docker kill

This command is used to kill the processes in a running container.

### Syntax

```
docker kill ContainerID
```

### Example

```
sudo docker kill 07b0b6f434fe
```

The above command will kill the processes in the running container **07b0b6f434fe**.

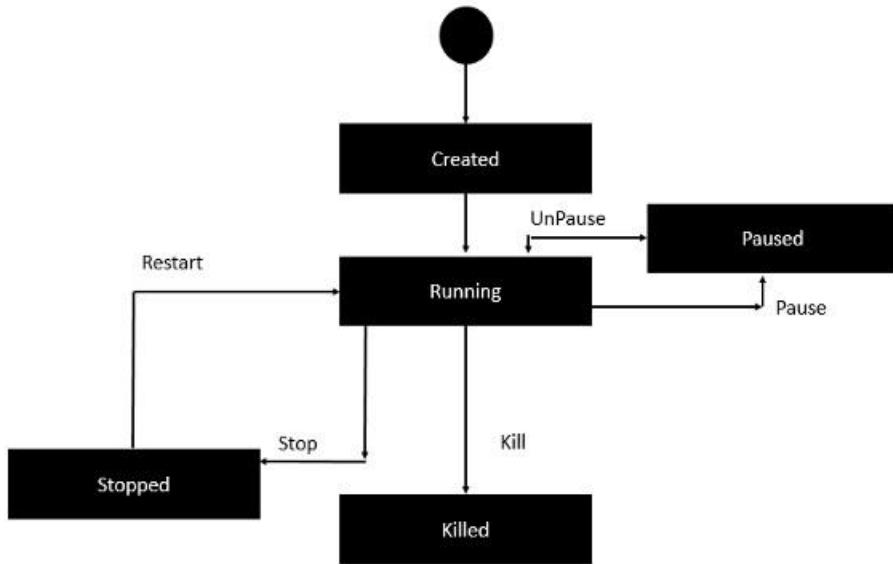
## Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES  
07b0b6f434fe        centos:latest      "/bin/bash"        23 minutes ago   Up 23 minutes       cocky_pare  
demo@ubuntuserver:~$ sudo docker kill 07b0b6f434fe  
07b0b6f434fe  
demo@ubuntuserver:~$
```

## Docker – Container Lifecycle

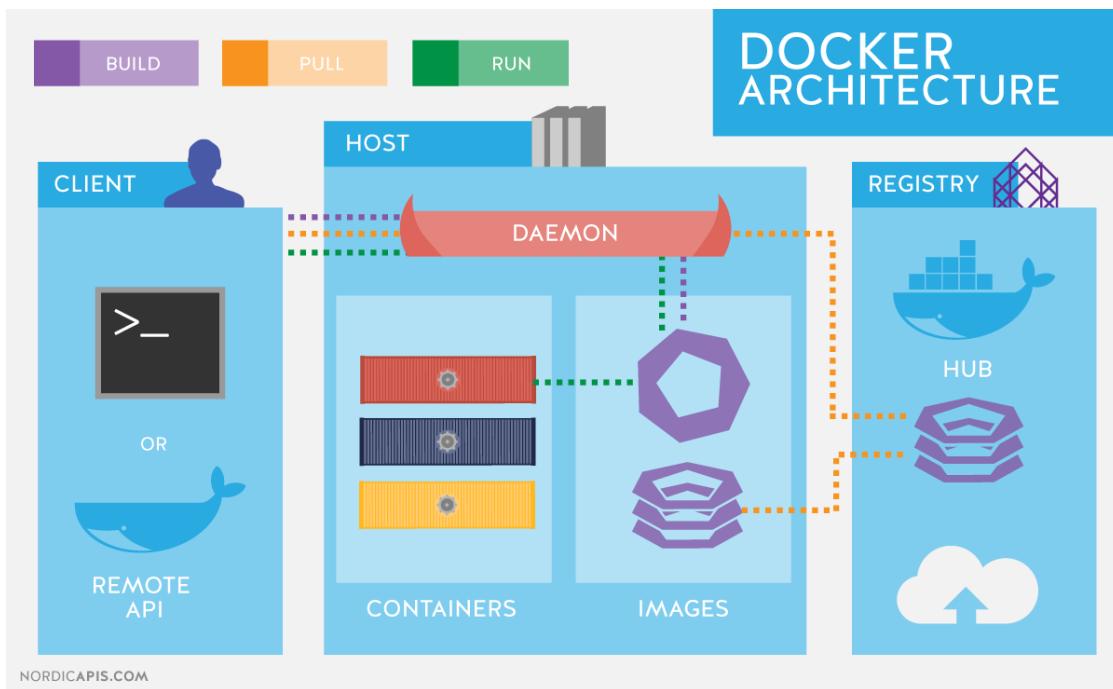
The following illustration explains the entire lifecycle of a Docker container.



- ★ Initially, the Docker container will be in the **created** state.
- ★ Then the Docker container goes into the **running** state when the Docker **run** command is used.
- ★ The Docker **kill** command is used to kill an existing Docker container.
- ★ The Docker **pause** command is used to pause an existing Docker container.
- ★ The Docker **stop** command is used to pause an existing Docker container.
- ★ The Docker **run** command is used to put a container back from a **stopped** state to a **running** state.

## Docker Architecture

The following image shows the standard and traditional architecture of **virtualization**.



- ★ The server is the physical server that is used to host multiple virtual machines.
- ★ The Host OS is the base machine such as Linux or Windows.
- ★ The Hypervisor is either VMWare or Windows Hyper V that is used to host virtual machines.
- ★ You would then install multiple operating systems as virtual machines on top of the existing hypervisor as Guest OS.
- ★ You would then host your applications on top of each Guest OS.
- ★ The server is the physical server that is used to host multiple virtual machines. So this layer remains the same.
- ★ The Host OS is the base machine such as Linux or Windows. So this layer remains the same.
- ★ Now comes the new generation which is the Docker engine. This is used to run the operating system which earlier used to be virtual machines as Docker containers.
- ★ All of the Apps now run as Docker containers.

The clear advantage in this architecture is that you don't need to have extra hardware for Guest OS. Everything works as Docker containers.

### **service docker stop**

This command is used to stop the Docker **daemon** process.

#### **Syntax**

**service docker stop**

### Example

```
sudo service docker stop
```

### Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo service docker stop
[sudo] password for demo:
docker stop/waiting
demo@ubuntuserver:~$
```

### service docker start

This command is used to start the Docker daemon process.

### Syntax

```
service docker start
```

### Example

```
sudo service docker start
```

### Output

When we run the above command, it will produce the following result –

```
demo@ubuntuserver:~$ sudo service docker stop
[sudo] password for demo:
docker stop/waiting
demo@ubuntuserver:~$ sudo service docker start
docker start/running, process 1942
demo@ubuntuserver:~$ _
```

## nsenter

This method allows one to attach to a container without exiting the container.

### Syntax

```
nsenter -m -u -n -p -i -t containerID command
```

### Options

- ★ **-u** is used to mention the **Uts namespace**
- ★ **-m** is used to mention the **mount namespace**
- ★ **-n** is used to mention the **network namespace**
- ★ **-p** is used to mention the **process namespace**
- ★ **-i** is used to make the container run in interactive mode.
- ★ **-t** is used to connect the I/O streams of the container to the host OS.
- ★ **containerID** – This is the ID of the container.
- ★ **Command** – This is the command to run within the container.

## Example

```
sudo nsenter -m -u -n -p -i -t 2978 /bin/bash
```

## Output

```
root@ubuntudemo:~# sudo nsenter -m -u -n -p -i -t 2978 /bin/bash
bash-4.2# exit
exit
root@ubuntudemo:~# sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
ef42a4c5e663        centos:latest      "/bin/bash"         9 minutes ago    Up 9 minutes       stoic_banach
root@ubuntudemo:~# _
```

## Docker File

A Docker File is a simple text file with instructions on how to build your images.

We can use any command that call on the command line. Docker builds images automatically by reading the instructions from the Dockerfile. The docker build command is used to build an image from the Dockerfile

# Dockerfile

BUILD	Both	RUN
FROM	WORKDIR	CMD
MAINTAINER	USER	ENV
COPY		EXPOSE
ADD		VOLUME
RUN		ENTRYPOINT
ONBUILD		
.dockerignore		

The following steps explain how you should go about creating a Docker File.

**Step 1** – Create a file called **Docker File** and edit it using **vim**. Please note that the name of the file has to be "Dockerfile" with "D" as capital.

```
vi DockerFile
```

```
root@ubuntudemo:~# sudo vim Dockerfile
```

**Step 2** – Build your Docker File using the following instructions.

```
#This is a sample Image FROM ubuntu MAINTAINER demousr@gmail.com RUN apt-get update RUN apt-get install -y nginx CMD ["echo","Image created"]
```

The following points need to be noted about the above file –

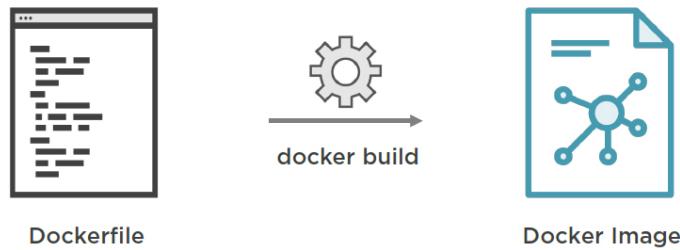
- ★ The first line "#This is a sample Image" is a comment. You can add comments to the Docker File with the help of the **#** command
- ★ The next line has to start with the **FROM** keyword. It tells docker, from which base image you want to base your image from. In our example, we are creating an image from the **ubuntu** image.
- ★ The next command is the person who is going to maintain this image. Here you specify the **MAINTAINER** keyword and just mention the email ID.
- ★ The **RUN** command is used to run instructions against the image. In our case, we first update our Ubuntu system and then install the nginx server on our **ubuntu** image.
- ★ The last command is used to display a message to the user.

**Step 3** – Save the file. In the next chapter, we will discuss how to build the image.

```
#This is a sample Image
FROM ubuntu
MAINTAINER demousr@gmail.com
RUN apt-get update
RUN apt-get install -y nginx
CMD ["echo","Image created"]
```

## Docker - Building Files

After creating the Dockerfile now we need to build the Dockerfile using the docker-build command.



This method allows the users to build their own Docker images.

### **Syntax**

```
docker build -t ImageName:TagName dir
```

### **Options**

- ★ **-t** – is to mention a tag to the image
- ★ **ImageName** – This is the name you want to give to your image.
- ★ **TagName** – This is the tag you want to give to your image.
- ★ **Dir** – The directory where the Docker File is present.

### **Example**

```
sudo docker build -t myimage:0.1.
```

Here, **myimage** is the name we are giving to the Image and **0.1** is the tag number we are giving to our image.

Since the Docker File is in the present working directory, we used **"."** at the end of the command to signify the present working directory.

### **Output**

From the output, you will first see that the Ubuntu Image will be downloaded from Docker Hub, because there is no image available locally on the machine.

```
root@ubuntudemo:~# sudo docker build -t myimage:0.1 .
Sending build context to Docker daemon 11.78 kB
Sending build context to Docker daemon
Step 0 : FROM ubuntu
latest: Pulling from ubuntu

4d330285ec99: Downloading 2.535 MB/50.1 MB
497dd7934d13: Download complete
bca38844f775: Download complete
e711979f32e2: Download complete
f38e3ca2b73a: Download complete
103d358b91a9: Download complete
```

Finally, when the build is complete, all the necessary commands would have run on the image.

```
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (@INC contains: /etc/perl /usr/local/lib/x86_64-linux-gnu/perl/5.22.1 /usr/local/share/perl/5.22.1 /usr/lib/x86_64-linux-gnu/perl5/5.22 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl/5.22 /usr/share/perl/5.22 /usr/local/lib/site_perl /usr/lib/x86_64-linux-gnu/perl-base .) at /usr/share/perl5/Debian/DebianFrontEnd/Readline.pm line 7.)
debconf: falling back to frontend: Teletype
Setting up nginx-core (1.10.0-0ubuntu0.16.04.4) ...
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Setting up nginx (1.10.0-0ubuntu0.16.04.4) ...
Processing triggers for libc-bin (2.23-0ubuntu4) ...
Processing triggers for sgml-base (1.26+nmu4ubuntu1) ...
Processing triggers for systemd (229-4ubuntu12) ...
--> e124d8a46ac3
Removing intermediate container e163d67fb876
Step 4 : CMD echo Image created
--> Running in 445fbe32a3ac
--> 8f7e840e407c
Removing intermediate container 445fbe32a3ac
Successfully built 8f7e840e407c
root@ubuntudemo:~#
```

You will then see the successfully built message and the ID of the new image. When you run the Docker **images command**, you would then be able to see your new image.

**docker images**

```

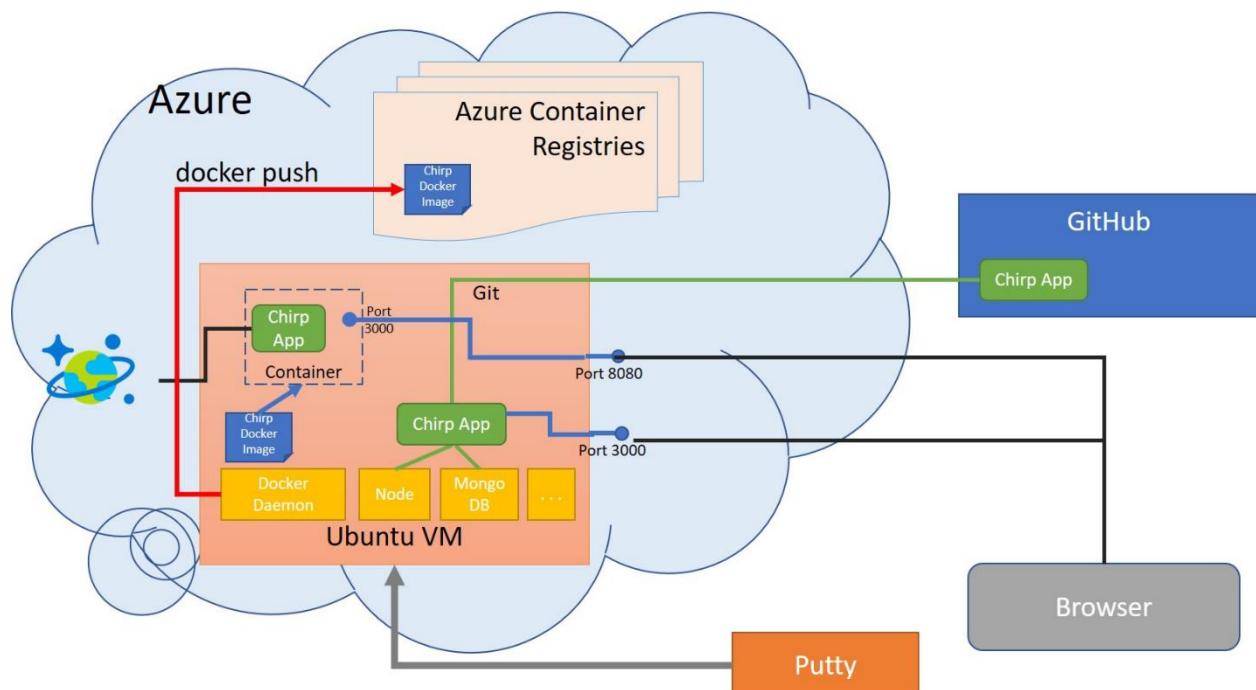
root@ubuntudemo:~# sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
myimage              0.1      8f7e840e407c    About a minute ago
ubuntu               latest   103d358b91a9    6 days ago
centos               latest   97cad5e16cb6    4 weeks ago
jpetazzo/nsenter     latest   a249cf324221    4 months ago
root@ubuntudemo:~#

```

You can now build containers from your new Image.

## Docker - Public Repositories

Public repositories can be used to host Docker images which can be used by everyone else. An example is the images which are available in Docker Hub. Most of the images such as Centos, Ubuntu, and Jenkins are all publicly available for all. We can also make our images available by publishing it to the public repository on Docker Hub.



For our example, we will use the **myimage** repository built in the "Building Docker Files" exercise and upload that image to Docker Hub. Let's first review the images on our Docker host to see what we can push to the Docker registry.

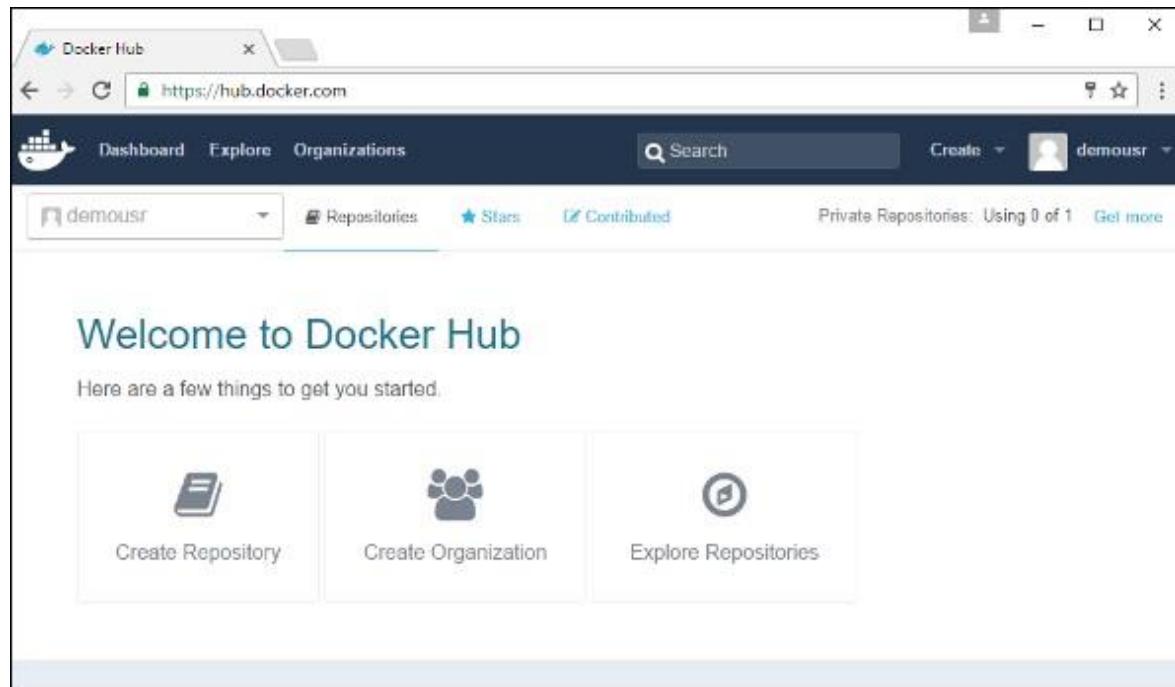
## docker images

```
demo@ubuntudemo:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
myimage              0.1      7738aa0e1b36    9 minutes ago
225.3 MB
centos               latest   67591570dd29    2 days ago
191.8 MB
ubuntu               latest   104bec311bcd    2 days ago
129 MB
demo@ubuntudemo:~$ _
```

Here, we have our **myimage:0.1** image which was created as a part of the "Building Docker Files" chapter. Let's use this to upload to the Docker public repository.

The following steps explain how you can upload an image to public repository.

**Step 1** – Log into Docker Hub and create your repository. This is the repository where your image will be stored. Go to <https://hub.docker.com/> and log in with your credentials.



**Step 2** – Click the button "Create Repository" on the above screen and create a repository with the name **demorep**. Make sure that the visibility of the repository is public.

Create Repository - Docker

https://hub.docker.com/add/repository/

5. Set it to be a private or public repository

demouser demorep

Short Description (100 Characters)

Full Description

Visibility

public

Create

Once the repository is created, make a note of the **pull** command which is attached to the repository.

demouser/demorep - Docker

https://hub.docker.com/r/demouser/demorep/

Full Description

Full description is empty for this repo.

Docker Pull Command

docker pull demouser/demorep

Owner:

demouser

The **pull** command which will be used in our repository is as follows –

**docker pull demouser/demorep**

**Step 3** – Now go back to the Docker Host. Here we need to tag our **myimage** to the new repository created in Docker Hub. We can do this via the Docker **tag command**.

We will learn more about this **tag command** later in this chapter.

**Step 4** – Issue the Docker login command to login into the Docker Hub repository from the command prompt. The Docker login command will prompt you for the username and password to the Docker Hub repository.

### **docker login**

```
demo@ubuntudemo:~$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: demousr
Password:
Login Succeeded
demo@ubuntudemo:~$
```

**Step 5** – Once the image has been tagged, it's now time to push the image to the Docker Hub repository. We can do this via the Docker **push** command. We will learn more about this command later in this chapter.

## **docker tag**

This method allows one to tag an image to the relevant repository.

### **Syntax**

```
docker tag imagID Repositoryname
```

### **Options**

- ★ **imagID** – This is the ImageID which needs to be tagged to the repository.
- ★ **Repositoryname** – This is the repository name to which the ImageID needs to be tagged to.

### **Example**

```
sudo docker tag ab0c1d3744dd demousr/demorep:1.0
```

### **Output**

A sample output of the above example is given below.

```
demo@ubuntudemo:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
myimage              0.1      ab0c1d3744dd    6 minutes ago
225.3 MB
centos               latest    67591570dd29    2 days ago
191.8 MB
ubuntu               latest    104bec311bcd    2 days ago
129 MB
demo@ubuntudemo:~$ sudo docker tag ab0c1d3744dd demousr/demorep:1.0
demo@ubuntudemo:~$
```

## docker push

This method allows one to push images to the Docker Hub.

### Syntax

```
docker push Repositoryname
```

### Options

- ★ **Repositoryname** – This is the repository name which needs to be pushed to the Docker Hub.

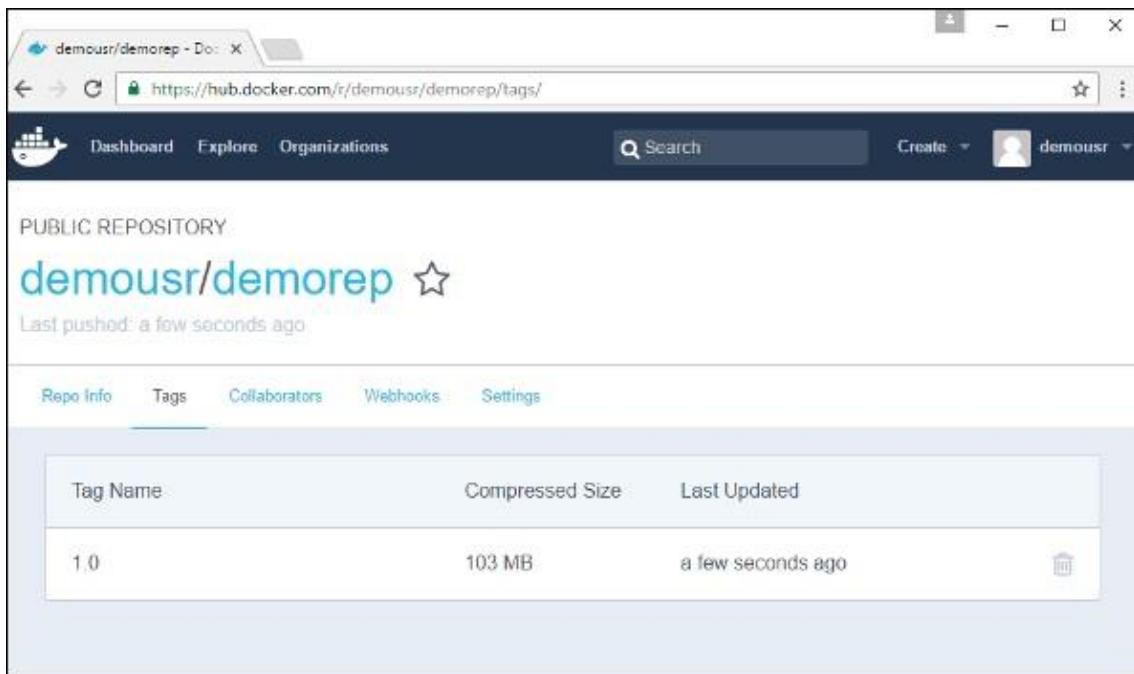
### Example

```
sudo docker push demousr/demorep:1.0
```

### Output

```
demo@ubuntudemo:~$ sudo docker push demousr/demorep:1.0
The push refers to a repository [docker.io/demousr/demorep]
2fa3ddba4e69: Layer already exists
ef84b80e23cc: Layer already exists
5972ebe5b524: Layer already exists
3d515508d4eb: Layer already exists
bbe6cef52379: Layer already exists
87f743c24123: Pushed
32d75bc97c41: Layer already exists
1.0: digest: sha256:1bcdcae3a9270a95798f02cd287b91956c5a6cf9fae08d82eb3d11f3a22d4
8d42 size: 1781
demo@ubuntudemo:~$
```

If you go back to the Docker Hub page and go to your repository, you will see the tag name in the repository.



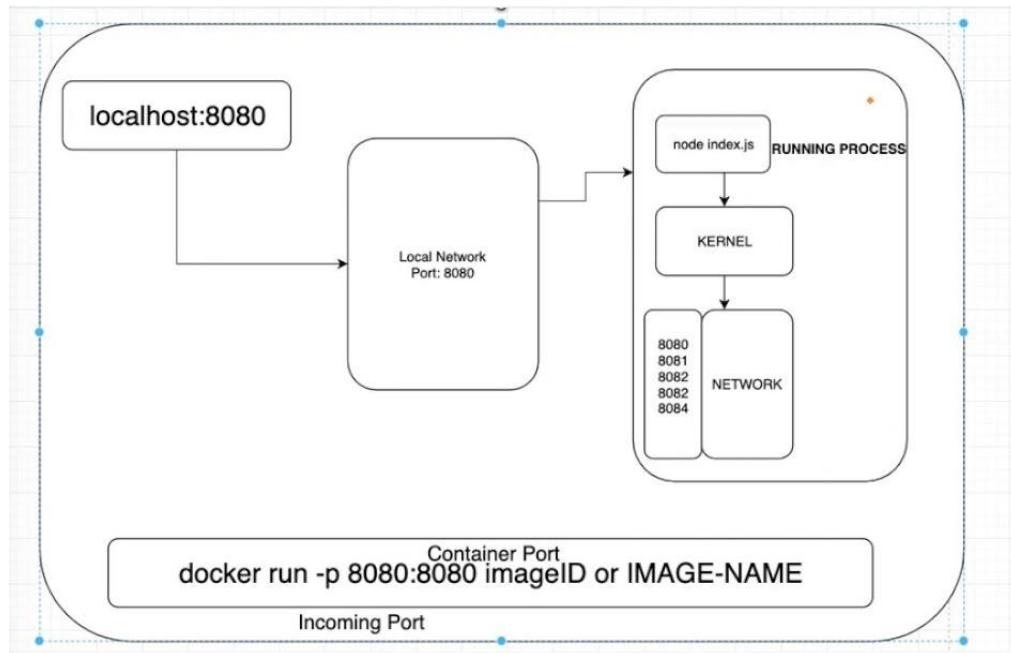
Now let's try to pull the repository we uploaded onto our Docker host. Let's first delete the images, **myimage:0.1** and **demousr/demorep:1.0**, from the local Docker host. Let's use the Docker **pull command** to pull the repository from the Docker Hub.

```
demo@ubuntudemo:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
centos              latest   67591570dd29    2 days ago
191.8 MB
ubuntu              latest   104bec311bcd    2 days ago
129 MB
demo@ubuntudemo:~$ sudo docker pull demousr/demorep:1.0
1.0: Pulling from demousr/demorep
b3e1c725a85f: Already exists
4daad8bddde31: Already exists
63fe8c0068a8: Already exists
4a70713c436f: Already exists
bd842a2105a8: Already exists
9b0dd3bf5478: Pull complete
6d3c35e0a8a2: Pull complete
Digest: sha256:1bcdae3a9270a95798f02cd287b91956c5a6cf9fae08d82eb3d11f3a22d48d42
Status: Downloaded newer image for demousr/demorep:1.0
demo@ubuntudemo:~$
```

From the above screenshot, you can see that the Docker **pull** command has taken our new repository from the Docker Hub and placed it on our machine.

## Docker - Managing Ports

In Docker, the containers themselves can have applications running on ports. When you run a container, if you want to access the application in the container via a port number, you need to map the port number of the container to the port number of the Docker host.



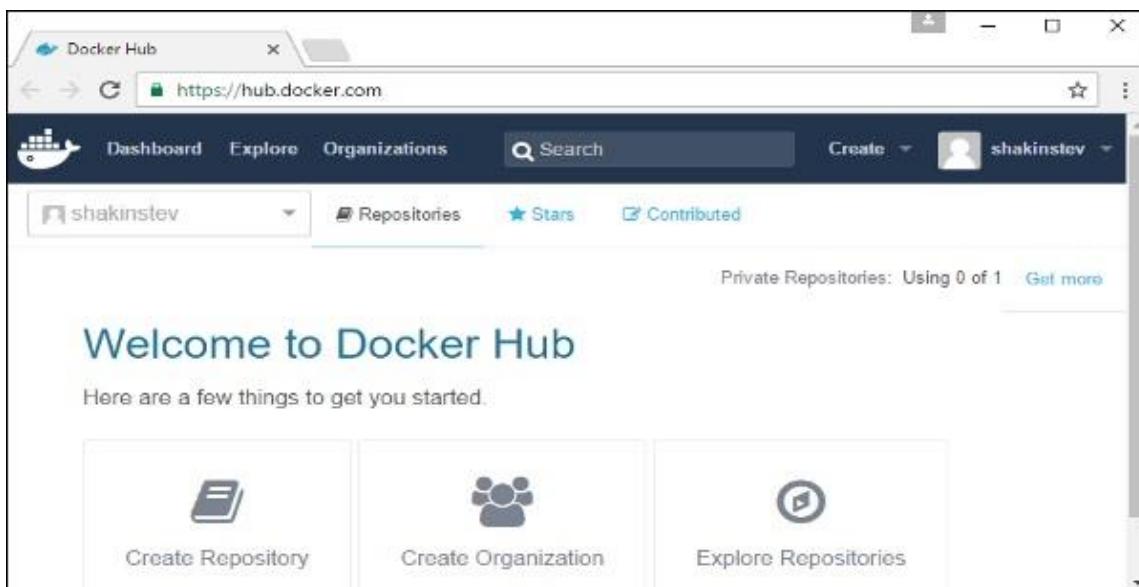
Let's look at an example of how this can be achieved.

In our example, we are going to download the Jenkins container from Docker Hub. We are then going to map the Jenkins port number to the port number on the Docker host.

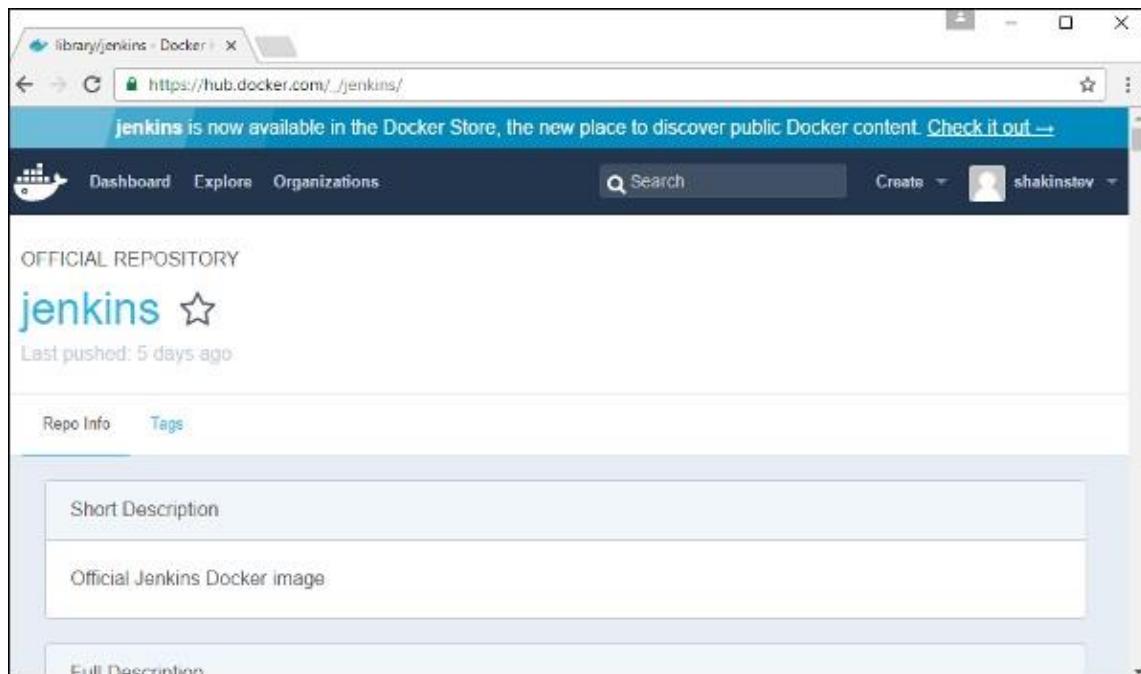
**Step 1** – First, you need to do a simple sign-up on Docker Hub.



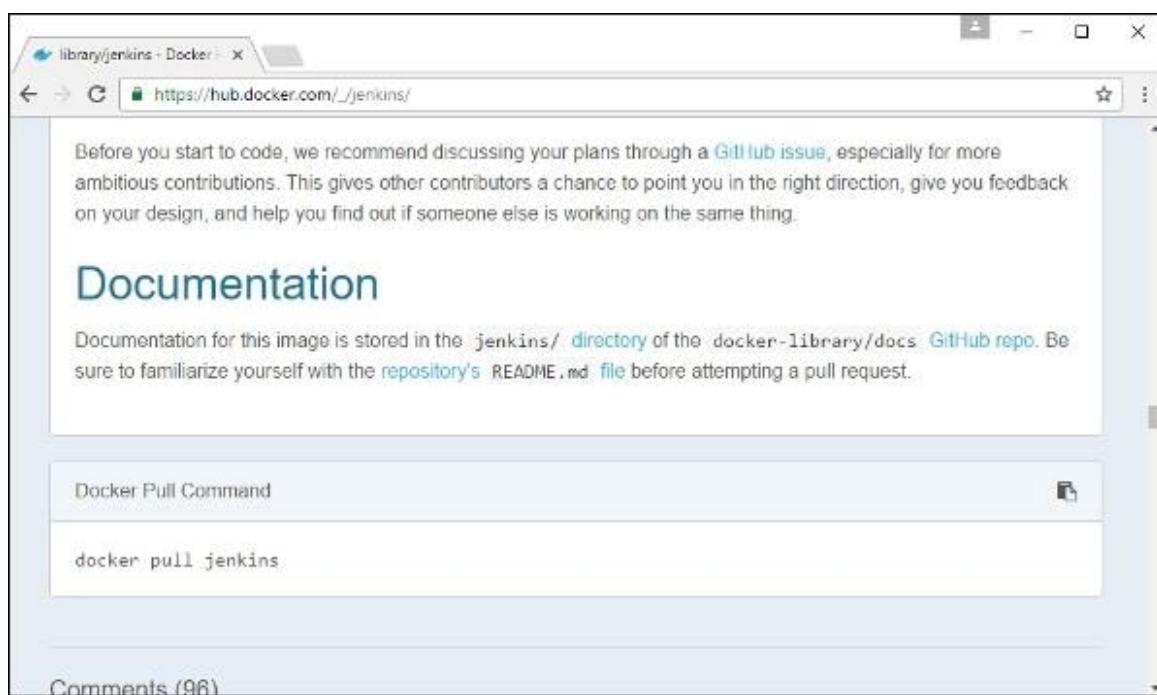
**Step 2** – Once you have signed up, you will be logged into Docker Hub.



**Step 3** – Next, let's browse and find the Jenkins image.



**Step 4** – If you scroll down on the same page, you can see the Docker **pull** command. This will be used to download the Jenkins Image onto the local Ubuntu server.



**Step 5** – Now go to the Ubuntu server and run the command –

**docker pull jenkins**

```
a079defbaeff: Pull complete
66181a89effa: Pull complete
f4d8f7d94b9c: Pull complete
98e5c3e08215: Pull complete
992fde8f3336: Pull complete
65b58e072756: Pull complete
0b0b6d6525a1: Pull complete
4e7171e4505a: Pull complete
469745638476: Pull complete
49d5aaafff78: Pull complete
c01281524fd6: Pull complete
00a759703a0b: Pull complete
da411a858795: Pull complete
7b8a0b4fd7d0: Pull complete
cbd9e145ea6b: Pull complete
700f8f527cd7: Pull complete
88d22231965c: Pull complete
a067af206313: Pull complete
211049e028a4: Pull complete
7249723069d8: Pull complete
6465c437f020: Pull complete
954c67861e66: Pull complete
6a14c8afbb3a: Pull complete
ec070f7e511e: Pull complete
983246da862f: Pull complete
998d1854867e: Pull complete
Digest: sha256:878e055f96c90af9281fd859f7c69ac289e0178594ff36bbb85e53b789691bec
Status: Downloaded newer image for jenkins:latest
demo@ubuntuserver:~$
```

**Step 6** – To understand what ports are exposed by the container, you should use the Docker **inspect command** to inspect the image.

Let's now learn more about this **inspect** command.

## **docker inspect**

This method allows one to return low-level information on the container or image.

### **Syntax**

```
docker inspect Container/Image
```

## Example

```
docker inspect jenkins
```

## Output

```
[  
  {  
    "Id": "sha256:ff6f0851ef574309ccd37c29e024f4d2a475d2436c8ebfa1180d45d8eb  
c9d1f2",  
    "RepoTags": [  
      "jenkins:latest"  
    ],  
    "RepoDigests": [  
      "jenkins@sha256:8d39e83e2e97f4f5f1ff6980f9bda7f7b3e7fb9208d94b377d4  
75a7e3054a5b"  
    ],  
    "Parent": "",  
    "Comment": "",  
    "Created": "2016-12-01T20:17:24.232532333Z",  
    "Container": "34804931e11a95400d6c40263174f32d498fd3ffff160f6deae807a323  
365c76",  
    "ContainerConfig": {  
      "Hostname": "6b3797ab1e90",  
      "Domainname": "",  
      "User": "jenkins",  
      "AttachStdin": false,  
      "AttachStdout": false,  
      "AttachStderr": false,  
      "ExposedPorts": {  
        "50000/tcp": {},  
        "8080/tcp": {}  
      },  
      "Tty": false,  
      "OpenStdin": false,  
--More--(14%)
```

The output of the **inspect** command gives a JSON output. If we observe the output, we can see that there is a section of "ExposedPorts" and see that there are two ports mentioned. One is the **data port** of 8080 and the other is the **control port** of 50000.

To run Jenkins and map the ports, you need to change the Docker **run** command and add the 'p' option which specifies the port mapping. So, you need to run the following command –

```
sudo docker run -p 8080:8080 -p 50000:50000 jenkins
```

The left-hand side of the port number mapping is the Docker host port to map to and the right-hand side is the Docker container port number.

When you open the browser and navigate to the Docker host on port 8080, you will see Jenkins up and running.



## Docker - Private Registries

You may not want to host the repositories on Docker Hub. For this, there is a repository container itself from Docker. Let's see how we can download and use the container for registry.

**Step 1** – Use the Docker **run** command to download the private registry. This can be done using the following command.

```
docker run -d -p 5000:5000 --name registry registry:2
```

The following points need to be noted about the above command –

- ★ **Registry** is the container managed by Docker which can be used to host private repositories.
- ★ The port number exposed by the container is 5000. Hence with the **-p command**, we are mapping the same port number to the 5000 port number on our localhost.
- ★ We are just tagging the registry container as "2", to differentiate it on the Docker host.

- ★ The **-d** option is used to run the container in detached mode. This is so that the container can run in the background

```
demo@ubuntudemo:~$ sudo docker run -d -p 5000:5000 --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry

3690ec4760f9: Already exists
930045f1e8fb: Already exists
feeaa90cbdbc: Already exists
61f85310d350: Already exists
b6082c239858: Already exists
Digest: sha256:1152291c7f93a4ea2ddc95e46d142c31e743b6dd70e194af9e6ebe530f782c17
Status: Downloaded newer image for registry:2
bf47ab9f46963746d8686994339a6a7fc33b4ac889c8f92ffe392cdfa8414068
demo@ubuntudemo:~$
```

**Step 2** – Let's do a **docker ps** to see that the registry container is indeed running.

```
demo@ubuntudemo:~$ sudo docker run -d -p 5000:5000 --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry

3690ec4760f9: Already exists
930045f1e8fb: Already exists
feeaa90cbdbc: Already exists
61f85310d350: Already exists
b6082c239858: Already exists
Digest: sha256:1152291c7f93a4ea2ddc95e46d142c31e743b6dd70e194af9e6ebe530f782c17
Status: Downloaded newer image for registry:2
bf47ab9f46963746d8686994339a6a7fc33b4ac889c8f92ffe392cdfa8414068
demo@ubuntudemo:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
          bf47ab9f4696        registry:2      "/entrypoint.sh /etc/"   About a minute ago   Up About a minute   0.0.0.0:5000->5000/tcp   registry
demo@ubuntudemo:~$ _
```

We have now confirmed that the registry container is indeed running.

**Step 3** – Now let's tag one of our existing images so that we can push it to our local repository. In our example, since we have the **centos** image available locally, we are going to tag it to our private repository and add a tag name of **centos**.

```
docker tag 67591570dd29 localhost:5000/centos
```

The following points need to be noted about the above command –

- ★ **67591570dd29** refers to the Image ID for the **centos** image.
- ★ **localhost:5000** is the location of our private repository.
- ★ We are tagging the repository name as **centos** in our private repository.

```

demo@ubuntudemo:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
demousr/demorep    1.0      ab0c1d3744dd  24 hours ago
225.3 MB
centos             latest   67591570dd29  3 days ago
191.8 MB
jenkins            latest   ff6f0851ef57  2 weeks ago
714.1 MB
registry           2        c9bd19d022f6  8 weeks ago
33.3 MB
demo@ubuntudemo:~$ sudo docker tag 67591570dd29 localhost:5000/centos
demo@ubuntudemo:~$ -

```

**Step 4** – Now let's use the Docker **push** command to push the repository to our private repository.

```
docker push localhost:5000/centos
```

Here, we are pushing the **centos** image to the private repository hosted at **localhost:5000**.

```

demo@ubuntudemo:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
demousr/demorep    1.0      ab0c1d3744dd  24 hours ago
225.3 MB
centos             latest   67591570dd29  3 days ago
191.8 MB
jenkins            latest   ff6f0851ef57  2 weeks ago
714.1 MB
registry           2        c9bd19d022f6  8 weeks ago
33.3 MB
demo@ubuntudemo:~$ sudo docker tag 67591570dd29 localhost:5000/centos
demo@ubuntudemo:~$ sudo docker push localhost:5000/centos
The push refers to a repository [localhost:5000/centos]
34e7b85d83e4: Pushed
latest: digest: sha256:c577af3197aacedf79c5a204cd7f493c8e07ffbcce7f88f7600bf19c68
8c38799 size: 529
demo@ubuntudemo:~$ -

```

**Step 5** – Now let's delete the local images we have for **centos** using the **docker rmi** commands. We can then download the required **centos** image from our private repository.

```
docker rmi centos:latest sudo docker rmi 67591570dd29
```

```

demo@ubuntudemo:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
demouser/demorep   1.0      ab0c1d3744dd  24 hours ago
225.3 MB
centos             latest   67591570dd29  3 days ago
191.8 MB
localhost:5000/centos  latest   67591570dd29  3 days ago
191.8 MB
jenkins            latest   ff6f0851ef57  2 weeks ago
714.1 MB
registry            2       c9bd19d022f6  8 weeks ago
33.3 MB
demo@ubuntudemo:~$ sudo docker rmi centos:latest
Untagged: centos:latest
Untagged: centos@sha256:c577af3197aacedf79c5a204cd7f493c8e07fffbce7f88f7600bf19c6
88c38799
demo@ubuntudemo:~$ sudo docker rmi 67591570dd29
Untagged: localhost:5000/centos:latest
Untagged: localhost:5000/centos@sha256:c577af3197aacedf79c5a204cd7f493c8e07fffbce
7f88f7600bf19c688c38799
Deleted: sha256:67591570dd29de0e124ee89d50458b098dbd83b12d73e5fdaf8b4dcbd4ea50f8
Deleted: sha256:34c7b85d83e48a22bd5dfa2b6b9ee9565b7ef672f09b3d2409c61635f9bca4db
demo@ubuntudemo:~$ 

```

**Step 6** – Now that we don't have any **centos** images on our local machine, we can now use the following Docker **pull** command to pull the **centos** image from our private repository.

**docker pull localhost:5000/centos**

Here, we are pulling the **centos** image to the private repository hosted at **localhost:5000**.

```

demo@ubuntudemo:~$ sudo docker pull localhost:5000/centos
Using default tag: latest
latest: Pulling from centos

45a2e645736c: Pull complete
Digest: sha256:c577af3197aacedf79c5a204cd7f493c8e07fffbce7f88f7600bf19c688c38799
Status: Downloaded newer image for localhost:5000/centos:latest
demo@ubuntudemo:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
demouser/demorep   1.0      ab0c1d3744dd  24 hours ago
225.3 MB
localhost:5000/centos  latest   67591570dd29  3 days ago
191.8 MB
jenkins            latest   ff6f0851ef57  2 weeks ago
714.1 MB
registry            2       c9bd19d022f6  8 weeks ago
33.3 MB
demo@ubuntudemo:~$ 

```

If you now see the images on your system, you will see the **centos** image as well.

## Example of a Dockerfile

In our example, we are going to use the Apache Web Server on Ubuntu to build our image. Let's follow the steps given below, to build our web server Docker file.

**Step 1** – The first step is to build our Docker File. Let's use **vim** and create a Docker File with the following information.

```
FROM ubuntu
RUN apt-get update
RUN apt-get install -y apache2
RUN apt-get install -y apache2-utils
RUN apt-get clean
EXPOSE 80
CMD ["apache2ctl", "-D", "FOREGROUND"]
```

The following points need to be noted about the above statements –

- ★ We are first creating our image to be from the Ubuntu base image.
- ★ Next, we are going to use the RUN command to update all the packages on the Ubuntu system.
- ★ Next, we use the RUN command to install apache2 on our image.
- ★ Next, we use the RUN command to install the necessary utility apache2 packages on our image.
- ★ Next, we use the RUN command to clean any unnecessary files from the system.
- ★ The EXPOSE command is used to expose port 80 of Apache in the container to the Docker host.
- ★ Finally, the CMD command is used to run apache2 in the background.

```
FROM ubuntu
MAINTAINER demousr@gmail.com
RUN apt-get update
RUN apt-get install -y apache2
RUN apt-get install -y apache2-utils
RUN apt-get clean
EXPOSE 80
CMD ["apache2ctl", "-D", "FOREGROUND"]
```

Now that the file details have been entered, just save the file.

**Step 2** – Run the Docker **build** command to build the Docker file. It can be done using the following command –

```
docker build -t="mywebserver".
```

We are tagging our image as **mywebserver**. Once the image is built, you will get a successful message that the file has been built.

```

Processing triggers for libc-bin (2.23-0ubuntu5) ...
Processing triggers for systemd (229-4ubuntu12) ...
Processing triggers for sgml-base (1.26+nmu4ubuntu1) ...
--> 3deecd5800ea
Removing intermediate container a34fbe45c6f0
Step 5 : RUN apt-get install -y apache2-utils
--> Running in 3924b32e72c0
Reading package lists...
Building dependency tree...
Reading state information...
apache2-utils is already the newest version (2.4.18-2ubuntu3.1).
apache2-utils set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
--> 9ddc59d1764b
Removing intermediate container 3924b32e72c0
Step 6 : RUN apt-get clean
--> Running in cb73b67c8109
--> 4a13c4c36e57
Removing intermediate container cb73b67c8109
Step 7 : EXPOSE 80
--> Running in 85245722be33
--> e4d2eb0fc674
Removing intermediate container 85245722be33
Step 8 : CMD apache2ctl -D FOREGROUND
--> Running in 49d3437f799f
--> 5ca8134b8d87
Removing intermediate container 49d3437f799f
Successfully built 5ca8134b8d87
demo@ubuntudemo:~$
```

**Step 3** – Now that the web server file has been built, it's now time to create a container from the image. We can do this with the Docker **run** command.

```
docker run -d -p 80:80 mywebserver
```

REPOSITORY	TAG	IMAGE ID	CREATED
mywebserver	latest	5ca8134b8d87	4 minutes ago
demouser/demorep	1.0	ab0c1d3744dd	26 hours ago
ubuntu	latest	104bec311bcd	3 days ago
jenkins	latest	ff6f0851ef57	2 weeks ago
registry	2	c9bd19d022f6	8 weeks ago

```

demo@ubuntudemo:~$ sudo docker run -d -p 80:80 mywebserver
42c70f5e90a2915d1954af2207de75657231c906feb9366f15a4e5c128c0675a
demo@ubuntudemo:~$ _
```

The following points need to be noted about the above command –

- ★ The port number exposed by the container is 80. Hence with the **-p** command, we are mapping the same port number to the 80 port number on our localhost.
- ★ The **-d** option is used to run the container in detached mode. This is so that the container can run in the background.

If you go to port 80 of the Docker host in your web browser, you will now see that Apache is up and running.



Docker has a host of instruction commands. These are commands that are put in the Docker File. Let's look at the ones which are available.

## **CMD Instruction**

This command is used to execute a command at runtime when the container is executed.

### **Syntax**

CMD command param1

### **Example**

In our example, we will enter a simple **Hello World** echo in our Docker File and create an image and launch a container from it.

**Step 1** – Build the Docker File with the following commands –

```
FROM ubuntu MAINTAINER demousr@gmail.com CMD ["echo", "hello world"]
```

Here, the CMD is just used to print **hello world**.

```
FROM ubuntu
MAINTAINER demousr@gmail.com
CMD ["echo", "hello world"]
```

**Step 2** – Build the image using the Docker **build** command.

```
demo@ubuntudemo:~$ sudo docker build -t="mynewdemo" .
Sending build context to Docker daemon 21.5 kB
Step 1 : FROM ubuntu
--> 104bec311bcd
Step 2 : MAINTAINER demousr@gmail.com
--> Using cache
--> 429c19673474
Step 3 : CMD echo hello world
--> Running in 6589f66cffd4
--> 90ab8626a009
Removing intermediate container 6589f66cffd4
Successfully built 90ab8626a009
demo@ubuntudemo:~$ _
```

**Step 3** – Run a container from the image.

```
demo@ubuntudemo:~$ sudo docker run mynewdemo
hello world
demo@ubuntudemo:~$ _
```

## **ENTRYPOINT**

This command can also be used to execute commands at runtime for the container. But we can be more flexible with the ENTRYPOINT command.

### **Syntax**

ENTRYPOINT command param1

### **Example**

Let's take a look at an example to understand more about ENTRYPOINT. In our example, we will enter a simple **echo** command in our Docker File and create an image and launch a container from it.

**Step 1** – Build the Docker File with the following commands –

```
FROM ubuntu MAINTAINER demousr@gmail.com ENTRYPOINT ["echo"]
```

```
FROM ubuntu
MAINTAINER demousr@gmail.com
ENTRYPOINT ["echo"]
```

**Step 2** – Build the image using the Docker **build** command.

```
demo@ubuntudemo:~$ sudo docker build -t="entrydemo" .
Sending build context to Docker daemon 22.53 kB
Step 1 : FROM ubuntu
--> 104bec311bcd
Step 2 : MAINTAINER demousr@gmail.com
--> Using cache
--> 429c19673474
Step 3 : ENTRYPOINT echo
--> Running in 4a06da685d12
--> c26b8ef5a8c9
Removing intermediate container 4a06da685d12
Successfully built c26b8ef5a8c9
demo@ubuntudemo:~$
```

**Step 3** – Run a container from the image.

```
demo@ubuntudemo:~$ sudo docker build -t="entrydemo" .
Sending build context to Docker daemon 22.53 kB
Step 1 : FROM ubuntu
--> 104bec311bcd
Step 2 : MAINTAINER demousr@gmail.com
--> Using cache
--> 429c19673474
Step 3 : ENTRYPOINT echo
--> Running in 4a06da685d12
--> c26b8ef5a8c9
Removing intermediate container 4a06da685d12
Successfully built c26b8ef5a8c9
demo@ubuntudemo:~$ sudo docker run entrydemo Hello World
Hello World
demo@ubuntudemo:~$
```

## ENV

This command is used to set environment variables in the container.

### Syntax

ENV key value

### Example

In our example, we will enter a simple **echo** command in our Docker File and create an image and launch a container from it.

**Step 1** – Build the Docker File with the following commands –

```
FROM ubuntu MAINTAINER demousr@gmail.com ENV var1=Tutorial var2=point
```

```
FROM ubuntu
MAINTAINER demousr@gmail.com
ENV var1=Tutorial var2=point
```

**Step 2** – Build the image using the Docker **build** command.

```
demo@ubuntudemo:~$ sudo docker build -t="envdemo" .
Sending build context to Docker daemon 23.04 kB
Step 1 : FROM ubuntu
--> 104bec311bcd
Step 2 : MAINTAINER demousr@gmail.com
--> Using cache
--> 429c19673474
Step 3 : ENV var1 Tutorial var2 point
--> Running in 8bd8eccb5986
--> 1def7e9aa854
Removing intermediate container 8bd8eccb5986
Successfully built 1def7e9aa854
demo@ubuntudemo:~$ _
```

**Step 3** – Run a container from the image.

```
demo@ubuntudemo:~$ sudo docker build -t="envdemo" .
Sending build context to Docker daemon 23.04 kB
Step 1 : FROM ubuntu
--> 104bec311bcd
Step 2 : MAINTAINER demousr@gmail.com
--> Using cache
--> 429c19673474
Step 3 : ENV var1 Tutorial var2 point
--> Running in 8bd8eccb5986
--> 1def7e9aa854
Removing intermediate container 8bd8eccb5986
Successfully built 1def7e9aa854
demo@ubuntudemo:~$ sudo docker run -it envdemo /bin/bash
root@b48d9e69cc34:/# _
```

**Step 4** – Finally, execute the **env** command to see the environment variables.

```

demo@ubuntudemo:~$ sudo docker run -it envdemo /bin/bash
root@b48d9e69cc34:/# env
var1=Tutorial
var2=point
HOSTNAME=b48d9e69cc34
TERM=xterm
LS_COLORS=rs=0:di=01:34:ln=01:36:mh=00:pi=40:33:so=01:35:do=01:35:bd=40:33:01:cd
=40:33:01:or=40:31:01:mi=00:su=37:41:sg=30:43:ca=30:41:tw=30:42:ow=34:42:st=37:4
t:ex=01:32:*.tar=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.taz=01:31:*.lha=01:
31:*.lz4=01:31:*.lzh=01:31:*.lzma=01:31:*.tlz=01:31:*.txz=01:31:*.tzo=01:31:*.t7
z=01:31:*.zip=01:31:*.z=01:31:*.2=01:31:*.dz=01:31:*.gz=01:31:*.lrz=01:31:*.lz=0
1:31:*.lzo=01:31:*.xz=01:31:*.bz2=01:31:*.bz=01:31:*.thz=01:31:*.tbz2=01:31:*.tz
=01:31:*.deb=01:31:*.rpm=01:31:*.jar=01:31:*.war=01:31:*.ear=01:31:*.sar=01:31:*
.rar=01:31:*.alz=01:31:*.ace=01:31:*.zoo=01:31:*.cpio=01:31:*.7z=01:31:*.rz=01:3
1:*.cab=01:31:*.jpg=01:35:*.jpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm
=01:35:*.ppm=01:35:*.tga=01:35:*.xbm=01:35:*.xpm=01:35:*.tif=01:35:*.tiff=01:35:
*.png=01:35:*.svg=01:35:*.svgz=01:35:*.mng=01:35:*.pcx=01:35:*.mov=01:35:*.mpg=0
1:35:*.mpeg=01:35:*.m2v=01:35:*.mkv=01:35:*.webm=01:35:*.ogm=01:35:*.mp4=01:35:*
.m4v=01:35:*.mp4v=01:35:*.vob=01:35:*.qt=01:35:*.nuv=01:35:*.wmv=01:35:*.ASF=01:
35:*.rm=01:35:*.rmvb=01:35:*.flc=01:35:*.avi=01:35:*.fli=01:35:*.flv=01:35:*.gl=
01:35:*.dl=01:35:*.xcf=01:35:*.xwd=01:35:*.yuv=01:35:*.cgm=01:35:*.emf=01:35:*.o
gv=01:35:*.ogx=01:35:*.aac=00:36:*.au=00:36:*.flac=00:36:*.m4a=00:36:*.mid=00:36:
*.midi=00:36:*.mka=00:36:*.mp3=00:36:*.mpc=00:36:*.ogg=00:36:*.ra=00:36:*.wav=0
0:36:*.oga=00:36:*.opus=00:36:*.spx=00:36:*.xspf=00:36:
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/
SHLVL=1
HOME=/root
_=~/usr/bin/env

```

## WORKDIR

This command is used to set the working directory of the container.

### Syntax

WORKDIR dirname

### Example

In our example, we will enter a simple **echo** command in our Docker File and create an image and launch a container from it.

**Step 1** – Build the Docker File with the following commands –

```
FROM ubuntu MAINTAINER demousr@gmail.com WORKDIR /newtemp CMD pwd
```

```

FROM ubuntu
MAINTAINER demousr@gmail.com
WORKDIR /newtemp
CMD pwd

```

**Step 2** – Build the image using the Docker **build** command.

```
demo@ubuntudemo:~$ sudo docker build -t="tempdemo" .
Sending build context to Docker daemon 23.55 kB
Step 1 : FROM ubuntu
--> 104bec311bcd
Step 2 : MAINTAINER demousr@gmail.com
--> Using cache
--> 429c19673474
Step 3 : WORKDIR /newtemp
--> Using cache
--> e09e6378e765
Step 4 : CMD pwd
--> Using cache
--> c7bedf4e3158
Successfully built c7bedf4e3158
demo@ubuntudemo:~$
```

**Step 3** – Run a container from the image.

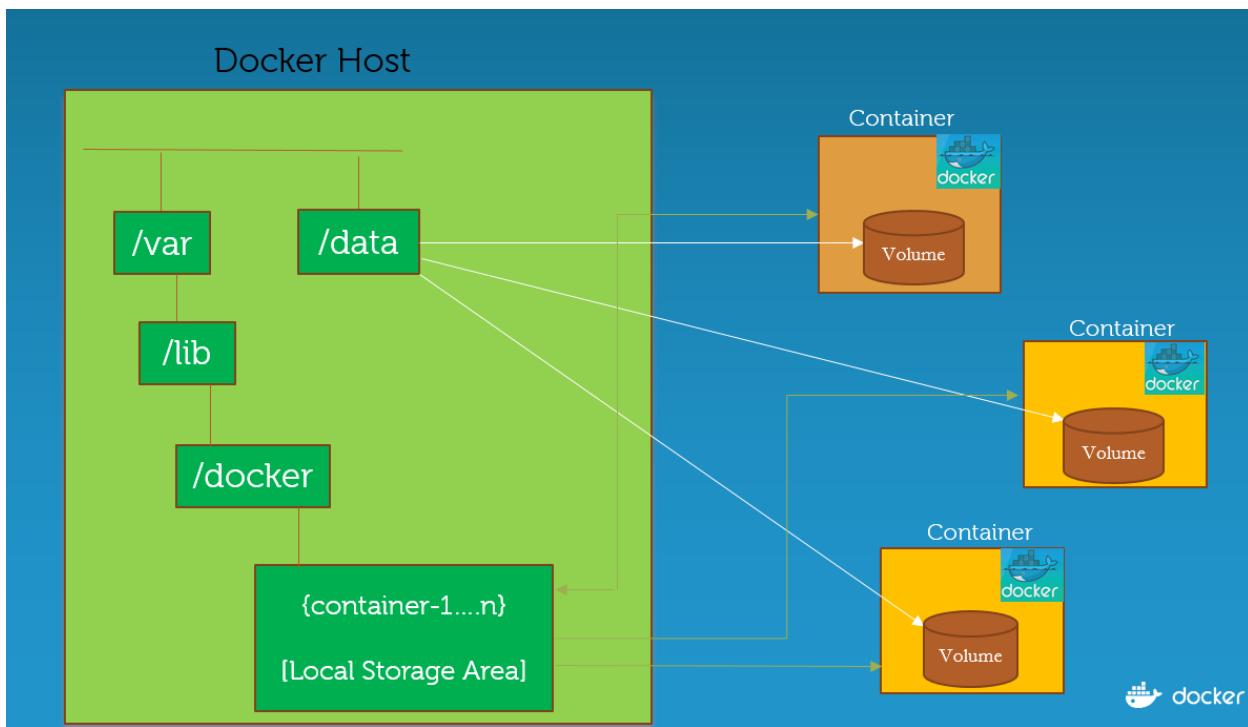
```
docker build -t="tempdemo"
```

```
docker run tempdemo
```

```
demo@ubuntudemo:~$ sudo docker build -t="tempdemo" .
Sending build context to Docker daemon 23.55 kB
Step 1 : FROM ubuntu
--> 104bec311bcd
Step 2 : MAINTAINER demousr@gmail.com
--> Using cache
--> 429c19673474
Step 3 : WORKDIR /newtemp
--> Using cache
--> e09e6378e765
Step 4 : CMD pwd
--> Using cache
--> c7bedf4e3158
Successfully built c7bedf4e3158
demo@ubuntudemo:~$ sudo docker run tempdemo
/newtemp
demo@ubuntudemo:~$ _
```

## Docker-Storage

Docker storage distinguishes three storage types. Two types are permanent: Docker volumes and bind Mounts and the third way of writing data is tmpfs. From the container perspective, it doesn't know what sort of storage is in use.



Docker has multiple storage drivers that allow one to work with the underlying storage devices. The following table shows the different storage drivers along with the technology used for the storage drivers.

<b>Technology</b>	<b>Storage Driver</b>
OverlayFS	overlay or overlay2
AUFS	aufs
Btrfs	brtfs
Device Manager	devicemanager
VFS	vfs
ZFS	zfs

Let us now discuss some of the instances in which you would use the various storage drivers –

### AUFS

- ★ This is a stable driver; can be used for production-ready applications.
- ★ It has good memory usage and is good for ensuring a smooth Docker experience for containers.
- ★ There is a high-write activity associated with this driver which should be considered.
- ★ It's good for systems which are of Platform as a service type work.

## Devicemapper

- ★ This is a stable driver; ensures a smooth Docker experience.
- ★ This driver is good for testing applications in the lab.
- ★ This driver is in line with the main Linux kernel functionality.

## Btrfs

- ★ This driver is in line with the main Linux kernel functionality.
- ★ There is a high-write activity associated with this driver which should be considered.
- ★ This driver is good for instances where you maintain multiple build pools.

## Ovelay

- ★ This is a stable driver and it is in line with the main Linux kernel functionality.
- ★ It has a good memory usage.
- ★ This driver is good for testing applications in the lab.

## ZFS

- ★ This is a stable driver and it is good for testing applications in the lab.
- ★ It's good for systems which are of Platform-as-a-Service type work.

To see the storage driver being used, issue the **docker info** command.

## Syntax

`docker info`

## Output

The following output shows that the main driver used is the **aufs** driver and that the root directory is stored in **/var/lib/docker/aufs**.

```
Root Dir: /var/lib/docker/aufs
Backing Filesystem: extfs
Dirs: 75
Dirperm1 Supported: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
  Volume: local
  Network: bridge host overlay null
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Security Options: apparmor
Kernel Version: 4.2.0-27-generic
Operating System: Ubuntu 14.04.4 LTS
OSType: linux
Architecture: x86_64
CPUs: 1
Total Memory: 993.1 MiB
Name: ubuntudemo
ID: ECDa:IFR3:2CQJ:FNXL:APJR:BT6Y:JJ75:FUE6:DNP5:PD7B:AOAD:YUB4
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Username: demouser
Registry: https://index.docker.io/v1/
WARNING: No swap limit support
Insecure Registries:
  127.0.0.0/8
```

## **Data Volumes**

In Docker, you have a separate volume that can be shared across containers. These are known as **data volumes**. Some of the features of data volume are –

- ★ They are initialized when the container is created.
- ★ They can be shared and also reused amongst many containers.
- ★ Any changes to the volume itself can be made directly.
- ★ They exist even after the container is deleted.

Let's look at our Jenkins container. Let's do a **docker inspect** to see the details of this image. We can issue the following command to write the output of the **docker inspect** command to a text file and then view the file accordingly.

```
docker inspect Jenkins > tmp.txt
```

When you view the text file using the **more command**, you will see an entry as **JENKINS\_HOME=/var/Jenkins\_home**.

This is the mapping that is done within the container via the Jenkins image.

```

"JAVA_DEBIAN_VERSION=8u111-b14-2~bpo8+1",
"CA_CERTIFICATES_JAVA_VERSION=20140324",
"JENKINS_HOME=/var/jenkins_home",
"JENKINS_SLAVE_AGENT_PORT=50000",
"TIMI_VERSION=0.9.0",
"TIMI_SHA=fa23d1e20732501c3bb8eeeca423c89ac80ed452",
"JENKINS_VERSION=2.19.4",
"JENKINS_UC=https://updates.jenkins.io",
"COPY_REFERENCE_FILE_LOG=/var/jenkins_home/copy_reference_file.log"
],
"Cmd": [
"/bin/sh",
"-c",
"#(nop) COPY file:2a6a3e16202b8ddab5edef50f712c16fe8f6980f5aea8
0c8c76b5db4f903913 in /usr/local/bin/install-plugins.sh "
],
"ArgsEscaped": true,
"Image": "sha256:86cb73edfb8b3a4681961047f1ca654c81586c7d8fdeaf7904e
cbca09ae4d720",
"Volumes": {
"/var/jenkins_home": {}
},
"WorkingDir": "",
"Entrypoint": [
"/bin/tini",
"--",
"/usr/local/bin/jenkins.sh"
]
--More-- (37%)

```

Now suppose you wanted to map the volume in the container to a local volume, then you need to specify the `-v` option when launching the container. An example is shown below

```
docker run -d -v /home/demo:/var/jenkins_home -p 8080:8080 -p 50000:50000 jenkins
```

The `-v` option is used to map the volume in the container which is **/var/jenkins\_home** to a location on our Docker Host which is **/home/demo**.

```
demo@ubuntudemo:~$ sudo docker run -d -v /home/demo:/var/jenkins_home -p 8080:80
80 -p 50000:50000 jenkins
```

Now if you go to the **/home/demo** location on your Docker Host after launching your container, you will see all the container files present there.

```

demo@ubuntudemo:~$ pwd
/home/demo
demo@ubuntudemo:~$ ls
config.xml          jobs
copy_reference_file.log  logs
Dockerfile           nodeMonitors.xml
hudson.model.UpdateCenter.xml  nodes
identity.key.enc    plugins
init.groovy.d        secret.key
jenkins.install.UpgradeWizard.state secret.key.not-so-secret
secrets
temp.txt
tmp.txt
userContent
users
war
demo@ubuntudemo:~$
```

## Changing the Storage Driver for a Container

If you wanted to change to the storage driver used for a container, you can do so when launching the container. This can be done by using the **-volume-driver** parameter when using the **docker run** command. An example is given below –

```
docker run -d --volume-driver=flocker -v /home/demo:/var/jenkins_home -p 8080:8080 -p 50000:50000 jenkins
```

The **-volume-driver** option is used to specify another storage driver for the container.

```
demo@ubuntudemo:~$ sudo docker run -d --volume-driver=flocker -v /home/demo:/var/jenkins_home -p 8080:8080 -p 50000:50000 jenkins  
9bffb1bfebee3fd15bc58ebade534bfbf18bf996d0052af7190aef5fdcf37180  
demo@ubuntudemo:~$
```

To confirm that the driver has been changed, first let's use the **docker ps** command to see the running containers and get the container ID. So, issue the following command first –

```
docker ps
```

Then issue a **docker inspect** against the container and put the output in a text file using the command.

```
docker inspect 9bffb1bfebee > temp.txt
```

```
demo@ubuntudemo:~$ sudo docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS  
           STATUS          PORTS  
S  
9bffb1bfebee        jenkins            "/bin/tini -- /usr/lo"   2 minutes ago  
Up 2 minutes         0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp   glo  
my_wing  
demo@ubuntudemo:~$ sudo docker inspect 9bffb1bfebee > temp.txt_
```

If you browse through the text file and go to the line which says **VolumeDriver**, you will see that the driver name has been changed.

```
        "Config": {},
        "NetworkMode": "default",
        "PortBindings": {
            "50000/tcp": [
                {
                    "HostIp": "",
                    "HostPort": "50000"
                }
            ],
            "8080/tcp": [
                {
                    "HostIp": "",
                    "HostPort": "8080"
                }
            ]
        },
        "RestartPolicy": {
            "Name": "no",
            "MaximumRetryCount": 0
        },
        "AutoRemove": false,
        "VolumeDriver": "flocker",
        "VolumesFrom": null,
        "CapAdd": null,
        "CapDrop": null,
        "Dns": [],
        "DnsOptions": [],
        "DnsSearch": []
--More-- (32%)
```

## Creating a Volume

A volume can be created beforehand using the **docker** command. Let's learn more about this command.

### Syntax

docker volume create --name=volumename --opt options

### Example

```
docker volume create --name = demo --opt o = size = 100m
```

In the above command, we are creating a volume of size 100MB and with a name of demo.

### Output

The output of the above command is shown below –

```
demo@ubuntudemo:~$ sudo docker volume create --name=demo --opt o=size=100m
demo
demo@ubuntudemo:~$ _
```

## Listing all the Volumes

You can also list all the **docker volumes** on a **docker host**. More details on this command is given below –

## Syntax

```
docker volume ls
```

## Output

The output of the above command is shown below –

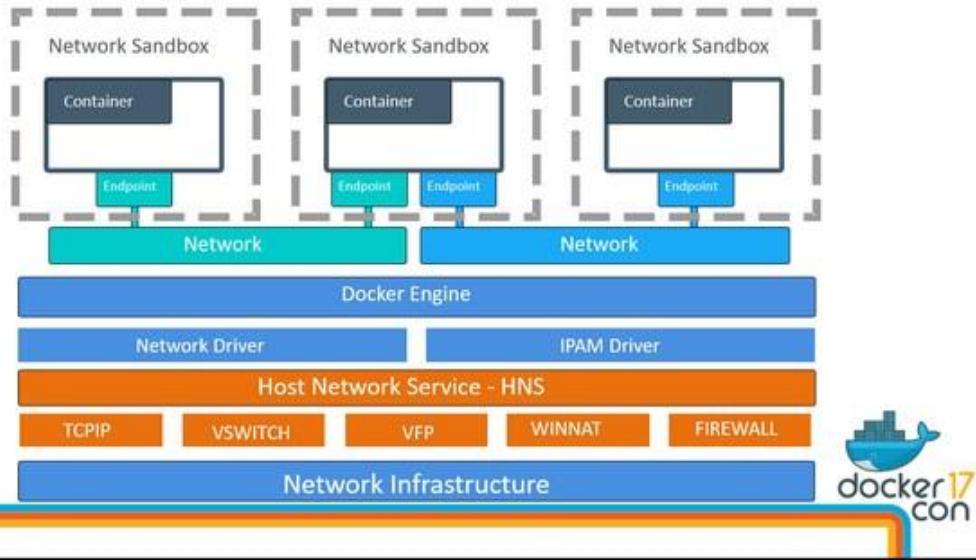
```
demo@ubuntudemo:~$ sudo docker volume ls
DRIVER    VOLUME NAME
local     0329aedc9cb821481d4a6c05619839294af86cfae3494a44b7ace23b1bc8
          0457e437c2496560355bb02e856d4443ec7e70dd6ceece12044b1cf4d40b
          a037
local     3405fcfa247666c62a05ec15988534c0d385444bc7d5475457bf108a10cb2
          f334
local     3cf320ee8bd98f558c25aff2803b300815da575bcc0e5a319e18316618e1
          f959
local     8a32b991086de55f3869ae1be7057f14dbc29c3aba70db6726a416670747
          d74c
local     9c7e3f37b4f5483c0550f6122b2e8f053d025b6174aecf14e0a12d96081e
          c450
local     demo
local     e94311df64b7ad609f851c5c66d0ec04b680c83539cc2721d32697f048fb
          fd0f
local     myvolume
demo@ubuntudemo:~$
```

## Docker – Networking

Docker networking **enables a user to link a Docker container to as many networks as he/she requires.** Docker Networks are used to provide complete isolation for Docker containers. Note: A user can add containers to more than one network. Let's move forward and look at the Advantages of networking.

Docker takes care of the networking aspects so that the containers can communicate with other containers and also with the Docker Host. If you do an **ifconfig** on the Docker Host, you will see the Docker Ethernet adapter. This adapter is created when Docker is installed on the Docker Host.

## Container Networking Model



```
demo@ubuntudemo:~$ sudo ifconfig
docker0  Link encap:Ethernet HWaddr 02:42:b4:a4:43:59
          inet addr:172.17.0.1 Bcast:0.0.0.0 Mask:255.255.0.0
              inet6 addr: fe80::42:b4ff:fea4:4359/64 Scope:Link
                  UP BROADCAST MULTICAST MTU:1500 Metric:1
                  RX packets:55 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:3448 (3.4 KB) TX bytes:2576 (2.5 KB)

eth0      Link encap:Ethernet HWaddr 08:00:27:f5:15:76
          inet addr:192.168.137.200 Bcast:192.168.137.255 Mask:255.255.255.0
              inet6 addr: fe80::a00:27ff:fed5:1576/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:199 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:13734 (13.7 KB) TX bytes:5238 (5.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
              inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING MTU:65536 Metric:1
                  RX packets:40 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:3184 (3.1 KB) TX bytes:3184 (3.1 KB)

demo@ubuntudemo:~$
```

This is a bridge between the Docker Host and the Linux Host. Now let's look at some commands associated with networking in Docker.

### **Listing All Docker Networks**

This command can be used to list all the networks associated with Docker on the host.

## Syntax

```
docker network ls
```

## Output

The output of the above command is shown below

```
demo@ubuntudemo:~$ sudo docker network ls
NETWORK ID      NAME        DRIVER      SCOPE
f07aad6ccadf   bridge      bridge      local
faae6bf679ea   host        host        local
54a2d37e7e00   none       null       local
demo@ubuntudemo:~$
```

## Inspecting a Docker network

If you want to see more details on the network associated with Docker, you can use the Docker **network inspect** command.

## Syntax

```
docker network inspect networkname
```

## Example

```
docker network inspect bridge
```

## Output

The output of the above command is shown below –

```
"Name": "bridge",
"Id": "f07aad6ccadf388082ccf9ad37db43f78adec85fb96ae0b2e9e8390c6d674242",

"Scope": "local",
"Driver": "bridge",
"EnableIPv6": false,
"IPAM": {
    "Driver": "default",
    "Options": null,
    "Config": [
        {
            "Subnet": "172.17.0.0/16",
            "Gateway": "172.17.0.1"
        }
    ]
},
"Internal": false,
"Containers": {},
"Options": {
    "com.docker.network.bridge.default_bridge": "true",
    "com.docker.network.bridge.enable_icc": "true",
    "com.docker.network.bridge.enable_ip_masquerade": "true",
    "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
    "com.docker.network.bridge.name": "docker0",
    "com.docker.network.driver.mtu": "1500"
},
"Labels": {}
}
]
demo@ubuntudemo:~$
```

Now let's run a container and see what happens when we inspect the network again. Let's spin up an Ubuntu container with the following command –

```
docker run -it ubuntu:latest /bin/bash
```

```
demo@ubuntudemo:~$ sudo docker run -it ubuntu:latest /bin/bash
```

Now if we inspect our network name via the following command, you will now see that the container is attached to the bridge.

```
docker network inspect bridge
```

```
        {
            "Subnet": "172.17.0.0/16",
            "Gateway": "172.17.0.1"
        }
    ],
    "Internal": false,
    "Containers": [
        "8e7b9a6dc121ba1c9a9fe48542db0149ee87b5efe031f518fb15751741ea0447":
        {
            "Name": "suspicious_blackwell",
            "EndpointID": "d30971d663e91ec2439355bb43c99613d500e35fbaae1957af74cb650f40723",
            "MacAddress": "02:42:ac:11:00:02",
            "IPv4Address": "172.17.0.2/16",
            "IPv6Address": ""
        }
    ],
    "Options": [
        "com.docker.network.bridge.default_bridge": "true",
        "com.docker.network.bridge.enable_icc": "true",
        "com.docker.network.bridge.enable_ip_masquerade": "true",
        "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
        "com.docker.network.bridge.name": "docker0",
        "com.docker.network.driver.mtu": "1500"
    ],
    "Labels": {}
]
demo@ubuntudemo:~$
```

## **Creating Your Own New Network**

One can create a network in Docker before launching containers. This can be done with the following command –

### **Syntax**

```
docker network create --driver drivername name
```

### **Example**

```
sudo docker network create --driver bridge new_nw
```

### **Output**

The output of the above command is shown below –

```
demo@ubuntudemo:~$ sudo docker network create --driver bridge new_nw
f01b64dc09425cc4906e20b5e17765e3248ea727068e0e2172bfc4aec42586fe
demo@ubuntudemo:~$ _
```

You can now attach the new network when launching the container. So let's spin up an Ubuntu container with the following command –

```
docker run -it --network=new_nw ubuntu:latest /bin/bash
```

```
demo@ubuntudemo:~$ sudo docker run -it --network=new_nw_ubuntu:latest /bin/bash
```

And now when you inspect the network via the following command, you will see the container attached to the network.

```
docker network inspect new_nw
```

```
"Scope": "local",
"Driver": "bridge",
"EnableIPv6": false,
"IPAM": {
    "Driver": "default",
    "Options": {},
    "Config": [
        {
            "Subnet": "172.18.0.0/16",
            "Gateway": "172.18.0.1/16"
        }
    ]
},
"Internal": false,
"Containers": {
    "38604fc42bcb5f78d42a8f40f34fa245301b2020a84c9e602786d2103ca6b847": {
        "Name": "boring_dubinsky",
        "EndpointID": "74d6b14a6393bf3081d5d9ec012b5b76b2ead49e85a5f664c621761a9e69612",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
    }
},
"Options": {},
"Labels": {}
}
]
leno@ubuntudemo:~$
```

## Docker Toolbox



# TOP 50 DOCKER TOOLS



## 1. Kubernetes

Kubernetes is an open-source platform that supports the containerization workspace and services. It is a portable and extensible ecosystem, whose services, tools & supports are widely available.

## 2. Prometheus

Prometheus is installed to test the deployed application developed in containers. It collects the data from the host at intervals and evaluates it to generate alerts if necessary. It implements a high dimensional data model and has a built-in expression browser, Grafana Integration, and a console template language to provide best of the analysis support. It has powerful queries that allow time slicing of data to generate ad-hoc graphs, tables, and alerts.

## 3. Dockersh

Dockersh is used to provide a user shell for isolated, containerized environments. It is used as a login shell for machines with multiple user interactions. A Dockersh when invoked, brings up a Docker container into the active state and spawn an interactive shell in the containers' namespace.

## 4. Twinklock

For comprehensive Docker security solutions for your Docker Enterprise or Docker community edition, Twistlock becomes the undeniable first choice. It provides security against advanced threats, vulnerability, and includes powerful runtime protection.

In general, Twistlock is programmed with machine learning to enable automated policy creation and enforcement, that provides full lifecycle, full stack, container security solutions.

## **5. Kitematic**

Kitematic is an open source project developed to simplify Docker installation on Mac and Windows system. It automates the installation process of Docker and provides an interactive Graphical User Interface (GUI) to select and run the Docker containers. Kitematic integrates with the Docker Machine to provide a Virtual Machine, on which Docker Engine will be installed locally.

## **6. Docker Compose**

This is used to configure and run multi-container Docker applications. Compose works in all stages: production, staging, development, testing, as well as CI workflows.

Docker Compose works in a 3 step process:

1. Define the app environment using the Dockerfile
2. Use YML file to define all the services, so that it can be run isolated, anywhere.
3. Run docker-compose up and Compose starts running

## **Docker Logging**

Docker has logging mechanisms in place which can be used to debug issues as and when they occur. There is logging at the **daemon level** and at the **container level**. Let's look at the different levels of logging.

### **Daemon Logging**

At the daemon logging level, there are four levels of logging available –

- ★ **Debug** – It details all the possible information handled by the daemon process.
- ★ **Info** – It details all the errors + Information handled by the daemon process.
- ★ **Errors** – It details all the errors handled by the daemon process.
- ★ **Fatal** – It only details all the fatal errors handled by the daemon process.

Go through the following steps to learn how to enable logging.

**Step 1** – First, we need to stop the **docker daemon process**, if it is already running. It can be done using the following command –

```
sudo service docker stop
```

```
demo@ubuntudemo:~$ sudo service docker stop
```

**Step 2** – Now we need to start the **docker daemon process**. But this time, we need to append the **-l** parameter to specify the logging option. So let's issue the following command when starting the **docker daemon process**.

```
dockerd -l debug &
```

The following points need to be noted about the above command –

- ★ **dockerd** is the executable for the **docker daemon process**.
- ★ The **-l** option is used to specify the logging level. In our case, we are putting this as debug
- ★ **&** is used to come back to the command prompt after the logging has been enabled.

```
demo@ubuntudemo:~$ sudo dockerd -l debug &
```

Once you start the Docker process with logging, you will also now see the **Debug Logs** being sent to the console.

```
DEBU[0001] Registering POST, /build
DEBU[0001] Registering POST, /swarm/init
DEBU[0001] Registering POST, /swarm/join
DEBU[0001] Registering POST, /swarm/leave
DEBU[0001] Registering GET, /swarm
DEBU[0001] Registering POST, /swarm/update
DEBU[0001] Registering GET, /services
DEBU[0001] Registering GET, /services/{id:.*}
DEBU[0001] Registering POST, /services/create
DEBU[0001] Registering POST, /services/{id:.*}/update
DEBU[0001] Registering DELETE, /services/{id:.*}
DEBU[0001] Registering GET, /nodes
DEBU[0001] Registering GET, /nodes/{id:.*}
DEBU[0001] Registering DELETE, /nodes/{id:.*}
DEBU[0001] Registering POST, /nodes/{id:.*}/update
DEBU[0001] Registering GET, /tasks
DEBU[0001] Registering GET, /tasks/{id:.*}
DEBU[0001] Registering GET, /networks
DEBU[0001] Registering GET, /networks/{id:.*}
DEBU[0001] Registering POST, /networks/create
DEBU[0001] Registering POST, /networks/{id:.*}/connect
DEBU[0001] Registering POST, /networks/{id:.*}/disconnect
DEBU[0001] Registering DELETE, /networks/{id:.*}
INFO[0001] API listen on /var/run/docker.sock
DEBU[0003] libcontainerd: containerd connection state change: READY
```

Now, if you execute any Docker command such as **docker images**, the Debug information will also be sent to the console.

```
demo@ubuntudemo:~$ sudo docker images
DEBU[0009] Calling GET /v1.24/images/json
REPOSITORY      TAG          IMAGE ID      CREATED
SIZE
node           latest        7c4d899628d5    3 days ago
660.4 MB
nginx          latest        01f818af747d    11 days ago
181.6 MB
mongo          latest        a3bf96cf65e     2 weeks ago
402 MB
web            latest        f5792fc30aaa    2 weeks ago
267.9 MB
firstweb       latest        0e52e236368a    2 weeks ago
267.6 MB
ubuntu         latest        104bec311bcd    3 weeks ago
129 MB
jenkins        latest        ff6f0851ef57    5 weeks ago
714.1 MB
demo@ubuntudemo:~$ _
```

## Container Logging

Logging is also available at the container level. So in our example, let's spin up an Ubuntu container first. We can do it by using the following command.

```
docker run -it ubuntu /bin/bash
```

```
demo@ubuntudemo:~$ sudo docker run -it ubuntu /bin/bash
root@6bf91271fcdd:/# demo@ubuntudemo:~$
demo@ubuntudemo:~$
```

Now, we can use the **docker log command** to see the logs of the container.

### Syntax

```
docker logs containerID
```

### Example

On our Docker Host, let's issue the following command. Before that, you can issue some commands whilst in the container.

```
docker logs 6bfb1271fcdd
```

### Output

```
demo@ubuntudemo:~$ sudo docker logs 6bfb1271fcdd
root@6bfb1271fcdd:#
root@6bfb1271fcdd:~# ifconfig
bash: ifconfig: command not found
root@6bfb1271fcdd:~# ls
bin dev home lib64 opt proc run sys tmp var
boot etc lib media opt root skin sys usr
demo@ubuntudemo:~$
```

From the output, you can see that the commands executed in the container are shown in the logs.

# KUBERNETES

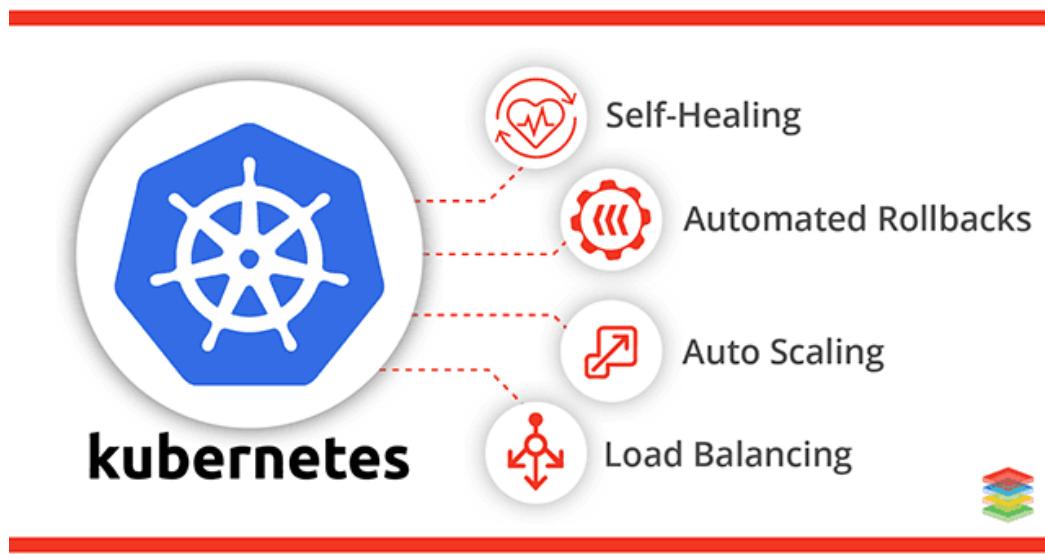
### What is Kubernetes?

- Kubernetes or k8s is an open-source platform that automates Linux container operations. In other words, you can cluster together groups of hosts running Linux containers.



# kubernetes

- Kubernetes keeps track of your container applications that are deployed into the cloud. It restarts orphaned containers, shuts down containers when they're not being used, and automatically provisions resources like memory, storage, and CPU when necessary.
- Kubernetes was originally developed and designed by engineers at Google. Google was one of the early contributors to Linux container technology and has talked publicly about how everything at Google runs in containers. (This is the technology behind Google's cloud services.)



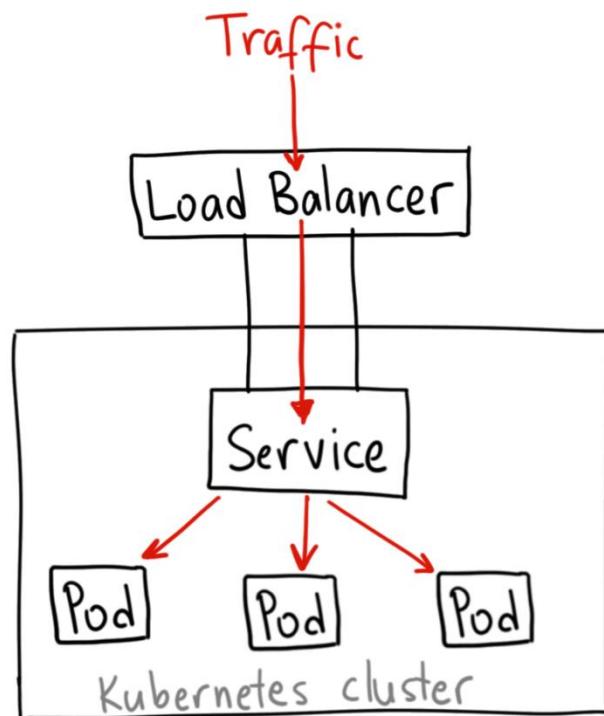
Before starting with Kubernetes, there are few necessary terms one should be aware of. These terms are directly connected to the operations we perform in the Kubernetes.

**Container Orchestration:** Container orchestration automates the deployment, management, scaling, and networking of containers. Enterprises that need to deploy and manage hundreds or thousands of Linux containers and hosts can benefit from container orchestration.

Container orchestration can be used in any environment where you use containers. It can help you to deploy the same application across different environments without needing to redesign it.

**Kubernetes Load Balancer:** The most basic type of load balancing in Kubernetes is actually load distribution, which is easy to implement at the dispatch level. Kubernetes uses two methods of load distribution, both of them operating through a feature called kube-proxy, which manages the virtual IPs used by services.

The default mode for kube-proxy is called iptables, which allows fairly sophisticated rule-based IP management. The native method for load distribution in iptables mode is random selection—an incoming request goes to a randomly chosen pod within a service. The older (and former default) kube-proxy mode is user space, which uses round-robin load distribution.



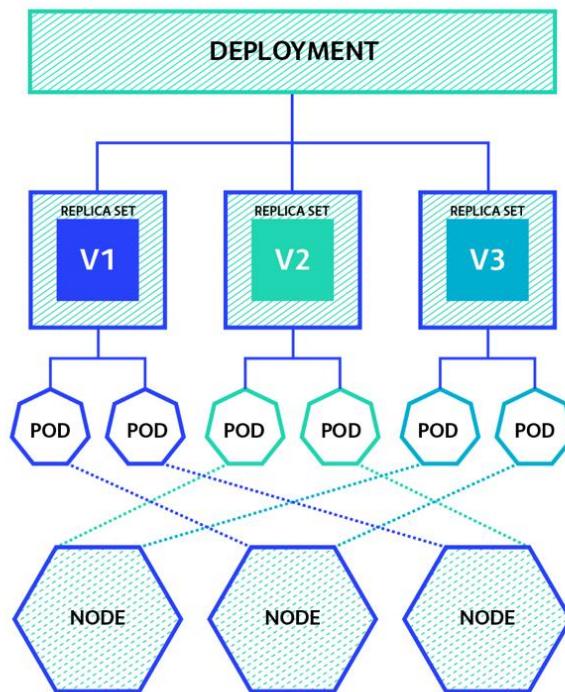
**Multinode Kubernetes Architecture:** It allows the deployment, orchestration, and scaling of container applications and micro-services across multiple hosts. ...

Understanding the basic Kubernetes concepts and multi-node deployment architecture will make the installation and management much easier.

It allows the deployment, orchestration, and scaling of container applications and micro-services across multiple hosts. Understanding the basic Kubernetes concepts and multi-node deployment architecture will make the installation and management much easier.

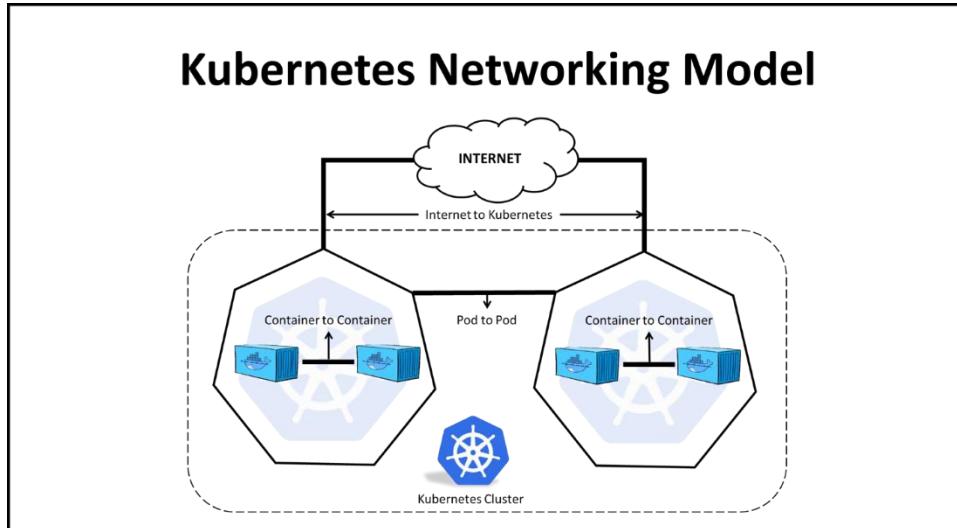
**Master Slave Architecture:** Kubernetes is really a master-slave type of architecture with certain components (master components) calling the shots in the cluster, and other components (node components) executing application workloads (containers) as decided by the master components. The master components manage the state of the cluster.

**Kubernetes Deployment:** A Kubernetes deployment is a resource object in Kubernetes that provides declarative updates to applications. A deployment allows you to describe an application's life cycle, such as which images to use for the app, the number of pods there should be, and the way in which they should be updated.



**Kubernetes Service:** A service can be defined as a logical set of pods. It can be defined as an abstraction on the top of the pod which provides a single IP address and DNS name by which pods can be accessed. With Service, it is very easy to manage load balancing configuration. It helps pods to scale very easily.

**Kubernetes Networking:** Kubernetes networking model dictates that Pods must be reachable by their IP address across Nodes. That is, the IP address of a Pod is always visible to other Pods in the network, and each Pod views its own IP address as the same as how other Pods see it.



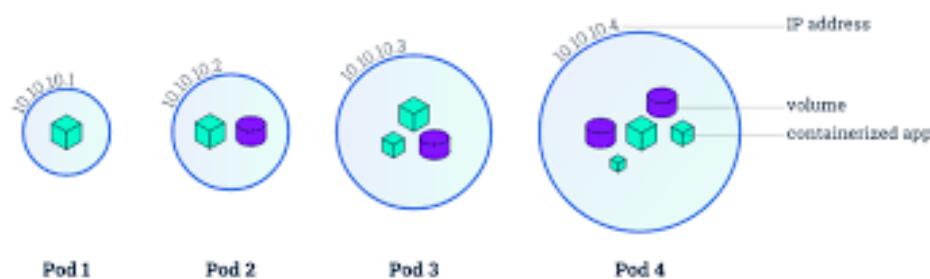
**Kubernetes Networking Drivers:** Kubernetes uses a pluggable support for network "drivers" via so called Container Network Interface (CNI) - started in CoreOS to abstract networks for the container runtime

**Kubernetes Pod:** A pod is a collection of containers and its storage inside a node of a Kubernetes cluster. It is possible to create a pod with multiple containers inside it. For example, keeping a database container and data container in the same pod.

#### Types of Pod

There are two types of Pods –

- Single container pod
- Multi container pod



**Kubernetes Side Car:** A sidecar is a utility container in the Pod and its purpose is to support the main container. It is important to note that standalone sidecar does not serve any purpose, it must be paired with one or more main containers.

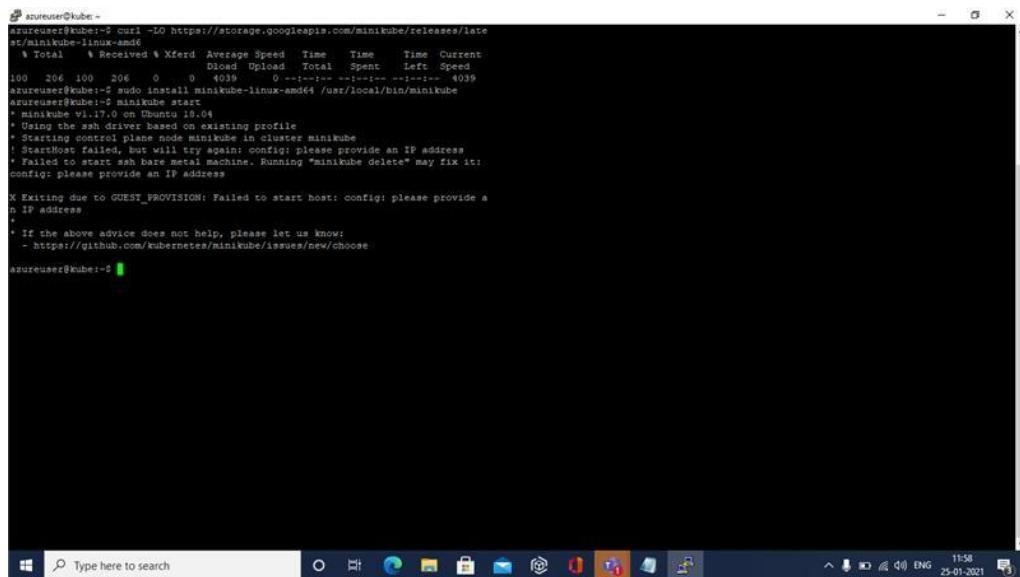
### **What is the role of Kubelet?**

The kubelet is responsible for maintaining a set of pods, which are composed of one or more containers, on a local system. Within a Kubernetes cluster, the kubelet functions as a local agent that watches for pod specs via the Kubernetes API server.

Before starting the Kubernetes, there should be a tool called "Minikube" in the system, the user is working on.

**Minikube:** Minikube is a tool that lets you run Kubernetes locally. Minikube runs a single-node Kubernetes cluster on your personal computer (including Windows, macOS and Linux PCs) so that you can try out Kubernetes, or for daily development work.

\*\*There should be Docker already installed in the Virtual Machine where you would be running the Minikube and Kubernetes operations, otherwise you would get the error as shown below (Refer the Docker Installation in the Docker Section).



```
azureuser@kube:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 | sudo install minikube-linux-amd64 /usr/local/bin/minikube
azureuser@kube:~$ minikube start
* minikube v1.17.0 on Ubuntu 18.04
* Using the ssh driver based on existing profile
* Starting control plane node minikube in cluster minikube
* This command will take a few minutes to complete. Please bear with us...
* Failed to start ssh bare metal machine. Running "minikube delete" may fix it:
config: please provide an IP address

* Existing due to GUEST_PROVISION: Failed to start host config: please provide a
  IP address
*
* If the above advice does not help, please let us know:
  - https://github.com/kubernetes/minikube/issues/new/choose

azureuser@kube:~$
```

## **MINIKUBE INSTALLATION COMMANDS:**

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

```
minikube start
```

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl
```

```
chmod +x ./kubectl
```

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

```
kubectl version -o json
```

```
kubectl config view
```

```
kubectl cluster-info
```

```
kubectl get nodes
```

```
kubectl get pod
```

```
minikube status
```

```
minikube addons list
```

```
kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4
```

```
kubectl expose deployment hello-minikube --type=NodePort --port=8080
```

```
kubectl get services hello-minikube
```

```
curl http://192.168.49.2:31518 ( 31518 : port number wouldn't be same for everyone. Check before entering the port number)
```

```
kubectl get services
```

```
kubectl create -h
```

```
#kubectl create deployment deploymentname --image=myimage
```

```
kubectl create deployment webapp-deployment --image=nginx
```

**#Will pull nginx image from Docker Hub, Name & Image is the minimum mandatory arguments to create deployment**

**kubectl get pod**

**kubectl get replicaset**

**kubectl edit deployment webapp-deployment**

**=> Will open an auto generated configuration file in Yaml format**

**kubectl logs containerIDOrName**

**=> containerIDOrName is the value we get from result of kubectl get pod**

**kubectl describe pod containerIDOrName**

**kubectl exec -it containerIDOrName -- bin/bash**

**kubectl describe deployments**

**kubectl delete deployment deploymentname**

**kubectl create deployment my-dep --image=nginx --replicas=3**

**kubectl scale deployment my-dep --current-replicas=3 --replicas=4**

**Configuration Files will be created with Deployment specific details**

**to create deployment with configuration, apply command is used**

**kubectl apply -f config\_file.yaml**

**=> Apply can be used for both create & Update**

**kubectl delete -f config\_file.yaml**

### **Creating a Deployment in Kubernetes:**

**kubectl create deployment NAME --image=image -- [args...]**

**Example:** `kubectl create deployment my-dep --image=nginx --replicas=3`

Also present in the above installation commands.

**ReplicaSet:** A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time. As such, it is often used to guarantee the availability of a specified number of identical Pods.

**Image: NGINX:** NGINX is open-source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability.

### **Deleting a Deployment in Kubernetes:**

`kubectl delete deployment deploymentname`

Ex: `kubectl delete deployment my-dep`

### **Creating a pod in Kubernetes:**

**POD:** A Pod is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers.

**`kubectl run mypod --image=nginx --port=80 --generator=run-pod/v1`**

This creates a pod in the Kubernetes and for displaying a pod

**`kubectl get pods`**

This displays all the pods present in the deployment.

### **Creating a Service in Kubernetes:**

**`kubectl expose deployment nginx --type=NodePort --port=80`**

In Kubernetes, a Service is an abstraction which defines a logical set of Pods and a policy by which to access them (sometimes this pattern is called a micro-service). The set of Pods targeted by a Service is usually determined by a selector.

Services:

kubectl expose --type=NodePort => Will expose App to the host & hence accessible from the outside

kubectl expose --type=ClusterIP => Will expose App to Virtual IP within the cluster & hence accessible from the inside the app cluster only

kubectl expose --type=LoadBalancer => Will expose App to Cloud Load Balancer(ELB, etc)

- Default range of Ports Auto Assigned during Expose command is 30000 to 32767
- This can be changed with --service-node-port-range in kubeapi-server configuration in master nodes.

**YAML:** YAML is a human-readable data-serialization language. It is commonly used for configuration files and in applications where data is being stored or transmitted. YAML targets many of the same communications applications

YAML uses three dashes ("---") to separate directives from document content.

```
object1.yaml x
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: hello-world
          image: hello-world:latest
          ports:
            - containerPort: 80
```

## **Yaml Based Configurations & Deployment:**

```
kubectl apply -f https://k8s.io/examples/application/deployment.yaml
```

### **Yaml Based Deployment:**

```
vi pod.yaml
```

Below is the YAML File Format

---

```
apiVersion: v1
kind: Pod
metadata:
  name: rss-site
  labels:
    app: web
spec:
  containers:
    - name: front-end
      image: nginx
      ports:
        - containerPort: 80
    - name: rss-reader
      image: nickchase/rss-php-nginx:v1
      ports:
        - containerPort: 88
```

---

```
kubectl create -f pod.yaml
```

```
kubectl get pods
```

```
kubectl get pods --show-labels
```

```
kubectl describe pod rss-site
```

```
kubectl get pods
```

## deployment.yaml

---

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rss-site
  labels:
    app: web
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: front-end
          image: nginx
      ports:
        - containerPort: 80
```

---

```
kubectl create -f deployment.yaml  
kubectl describe deployment rss-site  
kubectl get pods
```

### **Labels in Kubernetes:**

**Labels are key/value pairs that are attached to objects, such as pods. Labels are intended to be used to specify identifying attributes of objects that are meaningful and relevant to users, but do not directly imply semantics to the core system. Labels can be used to organize and to select subsets of objects.**

#### **Labels are part of metadata, labels are not unique**

Labels are tags: key & value pairs

Example:

metadata:

labels:

```
name=webapp
```

```
kubectl label nodes localhost cloudprovider=azure
```

```
kubectl label nodes localhost cloudprovider=aws
```

### **Label Selector:**

Match a label and do some operation

Example select label app=webapp and expose port 80 in a service configuration

nodeSelector:

```
cloudprovider: azure
```

```
kubectl get nodes --show-labels  
kubectl create -f nodeselector.yml  
kubectl get pods  
kubectl describe pod helloworld-deployment-7749869674-p9psx  
kubectl label nodes minikube cloudprovider=azure  
kubectl get nodes --show-labels  
kubectl describe pod helloworld-deployment-7749869674-p9psx
```

### **Health Checks:**

Health checks, or probes as they are called in Kubernetes, are carried out by the kubelet to determine when to restart a container (for liveness Probe ) and used by services and deployments to determine if a pod should receive traffic (for readinessProbe ).

**Liveness probe:** The kubelet uses liveness probes to know when to restart a container. For example, liveness probes could catch a deadlock, where an application is running, but unable to make progress. Restarting a container in such a state can help to make the application more available despite bugs.

**Readiness probes:** Readiness probes are designed to let Kubernetes know when your app is ready to serve traffic. Kubernetes makes sure the readiness probe passes before allowing a service to send traffic to the pod. If a readiness probe starts to fail, Kubernetes stops sending traffic to the pod until it passes.

Periodic Checks on a Pod to verify if the Pod is healthy & running properly without being hung and getting errors.

Options for Health check verification:

1. Run a command periodically
2. Accessibility of a URL
  - Always configure Health checks for production deployments

### **Creating a Health Check Probe:**

```
vi healthcheck.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: helloworld-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: helloworld
  template:
    metadata:
      labels:
        app: helloworld
  spec:
    containers:
      - name: k8s-demo
        image: nginx
    ports:
      - name: nginx-port
        containerPort: 80
    livenessProbe:
      httpGet:
        path: /
        port: nginx-port
    initialDelaySeconds: 15
    timeoutSeconds: 30
```

---

```
kubectl create -f healthcheck.yml
```

(Before creating a healthcheck yml file, create a YAML File and enter the YAML content)

```
kubectl get pods
```

```
kubectl describe pod helloworld-deployment-5f9469cd54-4pj
```

```
kubectl get deployment
```

```
kubectl edit deployment helloworld-deployment
```

Esc: wq

### **Readiness Probe:**

Done at the Pod startup only

```
kubectl create -f liveness-readiness.yml
```

```
watch -n1 kubectl get pods
```

Pod Status:

1. Pending
2. Succeeded
3. Running
4. Exited
5. Unknown
6. Failed

```
kubectl get pods -n kube-system
```

```
kubectl describe pod kube-apiserver-minikube -n kube-system
```

Pod Condition:

PodScheduled

Ready

Initialized

Unschedulable

ContainersReady

```
kubectl get pod kube-apiserver-minikube -n kube-system -o yaml
```

containerStatuses:

- containerID:

docker://1735e49ddb0212c006f5b43433a271f0f455de4f0f1cb1a42c6c3cf3a67db5bc

image: k8s.gcr.io/kube-apiserver:v1.19.4

imageID: docker-pullable://k8s.gcr.io/kube-apiserver@sha256:1d42d9b81ee77529c96080a1bb96d16e1988076400d3cf9e03c1ab2ec3df3374

lastState:

terminated:

containerID: docker://f6a72c45e595fe56dd27073174398bde7d7e904bb842901502a5e103989e4dfc

exitCode: 0

finishedAt: "2020-11-23T05:22:31Z"

reason: Completed

startedAt: "2020-11-23T02:43:02Z"

name: kube-apiserver

ready: true

restartCount: 1

```
started: true  
state:  
running:  
startedAt: "2020-11-24T02:12:16Z"
```

**Secrets:** Kubernetes Secrets let you store and manage sensitive information, such as passwords, OAuth tokens, and ssh keys. ... A Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key. Such information might otherwise be put in a Pod specification or in an image

Managing Safely the username, passwords which is Native to Kubernetes

### **Secrets Passing:**

1. As Environment Variables
2. As Secrets file & has to be shared on volumes mounted on Pods. Folder mounted on Volume will have the secrets files
3. Read from new files within the container
4. Pull from private images

```
echo -n "username" >> "./username.txt"  
echo -n "Password" >> "./password.txt"  
  
kubectl create secret generic db-user-pass --from-file=./username.txt --from-file=./password.txt  
  
ssh-keygen
```

```
kubectl create secret generic ssh-db-secret --from-file=ssh-privatekey=.ssh/id_rsa
```

Creating Secrets from Volumes

```
echo -n "root" | base64  
echo -n "password" | base64
```

### **Creating a secrets YAML file:**

```
vi secrets.yml
```

---

```
apiVersion: v1
kind: Secret
metadata:
  name: db-secrets
type: Opaque
data:
  username: cm9vdA==
  password: cGFzc3dvcmQ=
```

---

```
kubectl create -f secrets.yml
```

Create a secrets-volumes YAML File:

```
vi secrets-volumes.yml
```

---

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: secrets-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: helloworld
  template:
    metadata:
```

```
labels:  
  app: helloworld  
  
spec:  
  containers:  
    - name: nginx-webserver  
      image: nginx  
      ports:  
        - name: nginx-port  
          containerPort: 80  
      volumeMounts:  
        - name: cred-volume  
          mountPath: /etc/creds  
          readOnly: true  
  volumes:  
    - name: cred-volume  
      secret:  
        secretName: db-secrets
```

---

```
kubectl create -f secrets-volumes.yml
```

```
kubectl get pods
```

```
kubectl describe pod secrets-deployment-668587f4f-65467
```

Mounts:

/etc/creds from cred-volume (ro)

/var/run/secrets/kubernetes.io/serviceaccount from default-token-shwhs (ro)

```
kubectl exec -it secrets-deployment-668587f4f-65467 -- /bin/bash
```

```
ls /etc/creds  
cat /etc/creds/password  
kubectl delete deployment secrets-deployment
```

### **Deploying WordPress with Secrets:**

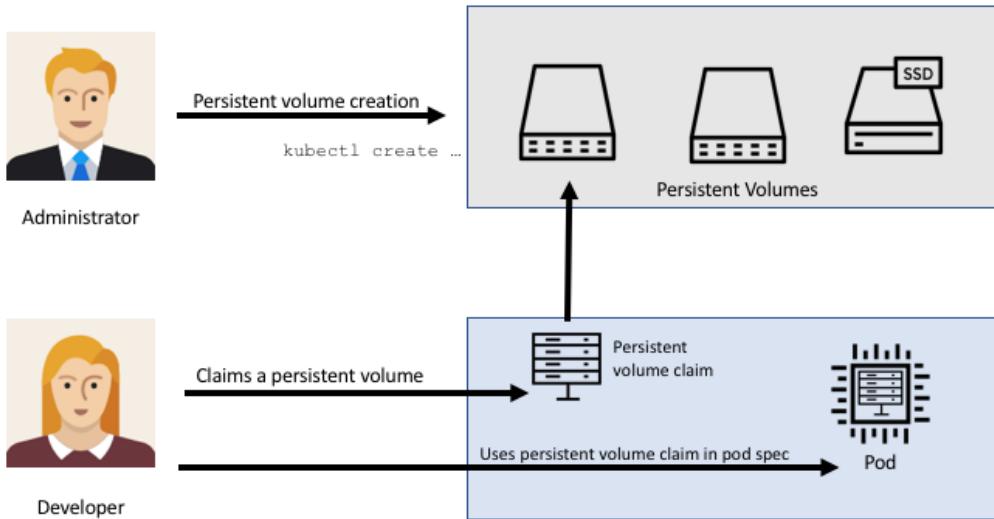
```
cd wordpress  
kubectl create -f wordpress-secrets.yml  
kubectl create -f wordpress-deployment.yml  
kubectl create -f wordpress-service.yml  
minikube service wordpress-service --url
```

The service will take a few minutes to not give error as the worpress containers take some time to initialize

```
curl http://192.168.49.2:31001
```

### **Kubernetes Volumes:**

In Kubernetes, a volume can be thought of as a directory which is accessible to the containers in a pod. We have different types of volumes in Kubernetes and the type defines how the volume is created and its content. It supports any or all the containers deployed inside the pod of Kubernetes.



The concept of volume was present with the Docker; however the only issue was that the volume was very much limited to a particular pod. As soon as the life of a pod ended, the volume was also lost.

The volumes that are created through Kubernetes is not limited to any container. It supports any or all the containers deployed inside the pod of Kubernetes. A key advantage of Kubernetes volume is, it supports different kind of storage wherein the pod can use multiple of them at the same time.

### **Types of Kubernetes Volume:**

Here is a list of some popular Kubernetes Volumes –

**emptyDir** – It is a type of volume that is created when a Pod is first assigned to a Node. It remains active as long as the Pod is running on that node. The volume is initially empty and the containers in the pod can read and write the files in the emptyDir volume. Once the Pod is removed from the node, the data in the emptyDir is erased.

**hostPath** – This type of volume mounts a file or directory from the host node's filesystem into your pod.

**gcePersistentDisk** – This type of volume mounts a Google Compute Engine (GCE) Persistent Disk into your Pod. The data in a gcePersistentDisk remains intact when the Pod is removed from the node.

**awsElasticBlockStore** – This type of volume mounts an Amazon Web Services (AWS) Elastic Block Store into your Pod. Just like gcePersistentDisk, the data in an awsElasticBlockStore remains intact when the Pod is removed from the node.

**nfs** – An nfs volume allows an existing NFS (Network File System) to be mounted into your pod. The data in an nfs volume is not erased when the Pod is removed from the node. The volume is only unmounted.

**iscsi** – An iscsi volume allows an existing iSCSI (SCSI over IP) volume to be mounted into your pod.

**flocker** – It is an open-source clustered container data volume manager. It is used for managing data volumes. A flocker volume allows a Flocker dataset to be mounted into a pod. If the dataset does not exist in Flocker, then you first need to create it by using the Flocker API.

### **Persistent Volume and Persistent Volume Claim**

Persistent Volume (PV) – It's a piece of network storage that has been provisioned by the administrator. It's a resource in the cluster which is independent of any individual pod that uses the PV.

### Creating Persistent Volume

```
kind: PersistentVolume -----> 1
apiVersion: v1
metadata:
  name: pv0001 -----> 2
  labels:
    type: local
spec:
  capacity: -----> 3
    storage: 10Gi -----> 4
  accessModes:
    - ReadWriteOnce -----> 5
  hostPath:
    path: "/tmp/data01" -----> 6
```

In the above code, we have defined –

- **kind: PersistentVolume** → We have defined the kind as PersistentVolume which tells kubernetes that the yaml file being used is to create the Persistent Volume.
- **name: pv0001** → Name of PersistentVolume that we are creating.
- **capacity:** → This spec will define the capacity of PV that we are trying to create.
- **storage: 10Gi** → This tells the underlying infrastructure that we are trying to claim 10Gi space on the defined path.
- **ReadWriteOnce** → This tells the access rights of the volume that we are creating.
- **path: "/tmp/data01"** → This definition tells the machine that we are trying to create volume under this path on the underlying infrastructure.

Persistent Volume Claim (PVC) – The storage requested by Kubernetes for its pods is known as PVC. The user does not need to know the underlying provisioning. The claims must be created in the same namespace where the pod is created.

### Creating a Volume in Kubernetes:

#We should create a yaml file named "sc" and then applying. The content of the YAML file should be according to the Volumes.

```
kubectl apply -f sc.yml
```

```
kubectl get pv
```

```
kubectl apply -f ss.yml
```

```
kubectl get pvc
```

```
kubectl exec -it helloworld-deployment-84d6ddb9b9-lt7f4 -- /bin/bash
```

**Describing PV** — `kubectl describe pv pv0001`

**Flannel Network plugin in Kubernetes:** Flannel is a virtual network that gives a subnet to each host for use with container runtimes. Platforms like Google's Kubernetes assume that each container (pod) has a unique, routable IP inside the cluster. The advantage of this model is that it reduces the complexity of doing port mapping.

Port mapping redirects a communication request from one address and port number combination to another while the packets are traversing a network gateway.

### ★ Definitions of Various Kubernetes Tool:

**Kubernetes Pod Security:** A Pod Security Policy is a cluster-level resource that controls security sensitive aspects of the pod specification. The PodSecurityPolicy objects define a set of conditions that a pod must run with in order to be accepted into the system, as well as defaults for the related fields.

By default, data in Kubernetes secrets is stored in Base64 encoding, which is practically the same as plaintext

### Kubernetes Deployment Vs Pod:

**Pods** - runs one or more closely related containers

**Services** - sets up networking in a Kubernetes cluster

**Deployment** - Maintains a set of identical pods, ensuring that they have the correct config and that the right number of them exist.

#### Pods:

Runs a single set of containers

Good for one-off dev purposes

Rarely used directly in production

#### Deployment:

Runs a set of identical pods

Monitors the state of each pod, updating as necessary

Good for dev

Good for production

**Microservices Architecture:** Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are:

1. Highly maintainable and testable
2. Loosely coupled
3. Independently deployable

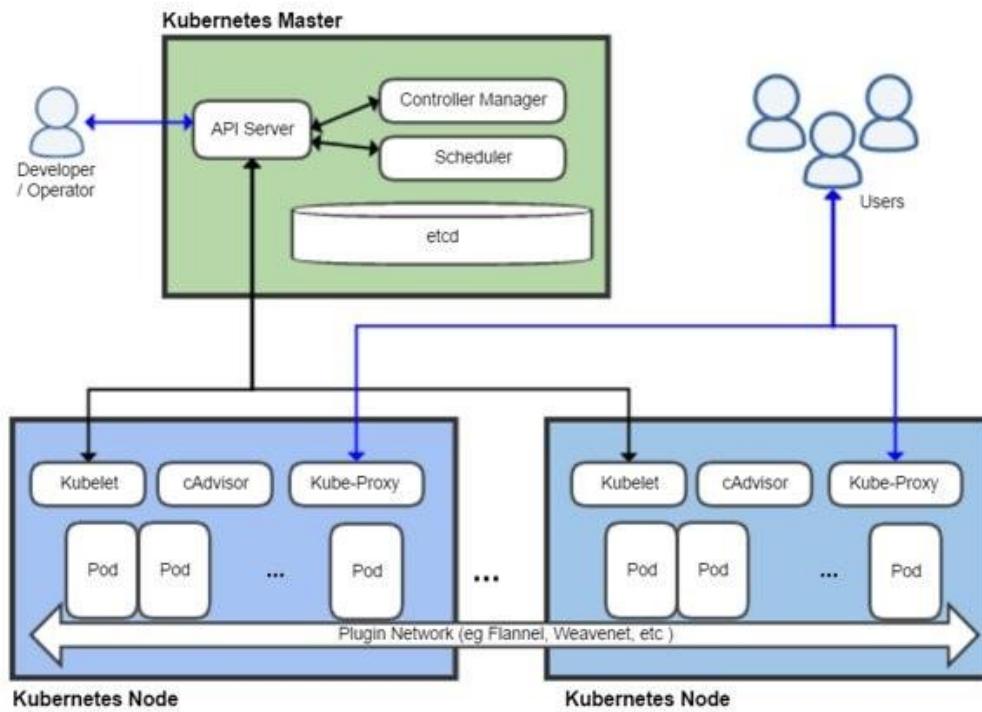
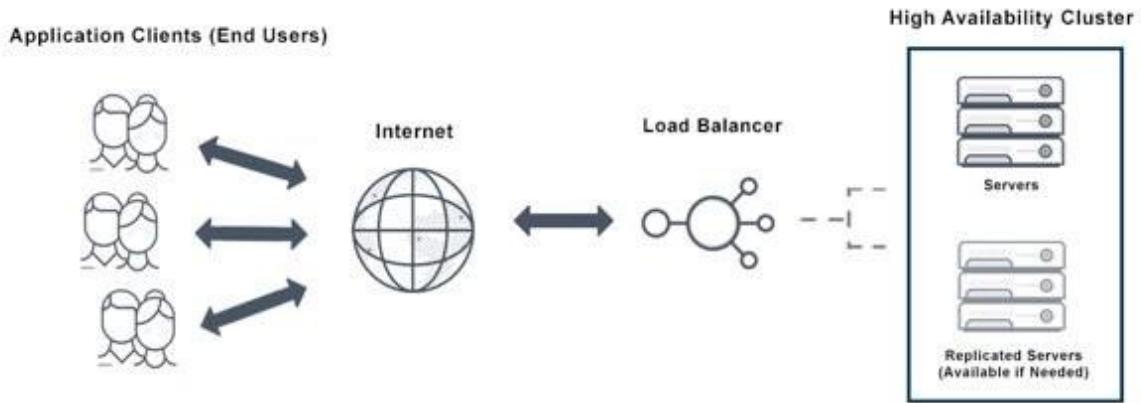
- The microservice architecture pattern language is a collection of patterns for applying the microservice architecture. It has two goals:
  - The pattern language enables you to decide whether microservices are a good fit for your application.
  - The pattern language enables you to use the microservice architecture successfully.

**Kubernetes etcd:** Etcd is an open-source distributed key-value store used to hold and manage the critical information that distributed systems need to keep running.

Kubernetes is a distributed system, so it needs a distributed data store like etcd. etcd lets any of the nodes in the Kubernetes cluster read and write data.

etcd is a consistent and highly-available key value store used as Kubernetes' backing store for all cluster data. If your Kubernetes cluster uses etcd as its backing store, make sure you have a backup plan for those data.

**Kubernetes High availability:** Kubernetes High-Availability is about setting up Kubernetes, along with its supporting components in a way that there is no single point of failure. A single master cluster can easily fail, while a multi-master cluster uses multiple master nodes, each of which has access to same worker nodes.



**Taints:** Taints are used to repel Pods from specific nodes.

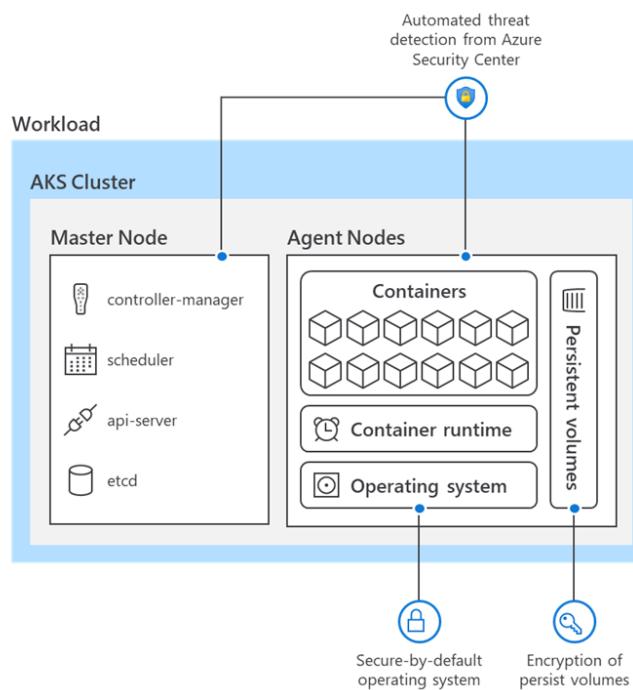
**Tolerations:** Tolerations are applied to pods, and allow (but do not require) the pods to schedule onto nodes with matching taints.

Taints and tolerations work together to ensure that pods are not scheduled onto inappropriate nodes.

## Azure Kubernetes Services(AKS)

Azure Kubernetes Service (AKS) makes it simple to deploy a managed Kubernetes cluster in Azure. AKS reduces the complexity and operational overhead of managing Kubernetes by offloading much of that responsibility to Azure. As a hosted Kubernetes service, Azure handles critical tasks like health monitoring and maintenance for you. The Kubernetes masters are managed by Azure. You only manage and maintain the agent nodes. As a managed Kubernetes service, AKS is free - you only pay for the agent nodes within your clusters, not for the masters.

You can create an AKS cluster in the Azure portal, with the Azure CLI, or template driven deployment options such as Resource Manager templates and Terraform. When you deploy an AKS cluster, the Kubernetes master and all nodes are deployed and configured for you. Additional features such as advanced networking, Azure Active Directory integration, and monitoring can also be configured during the deployment process. Windows Server containers are supported in AKS.

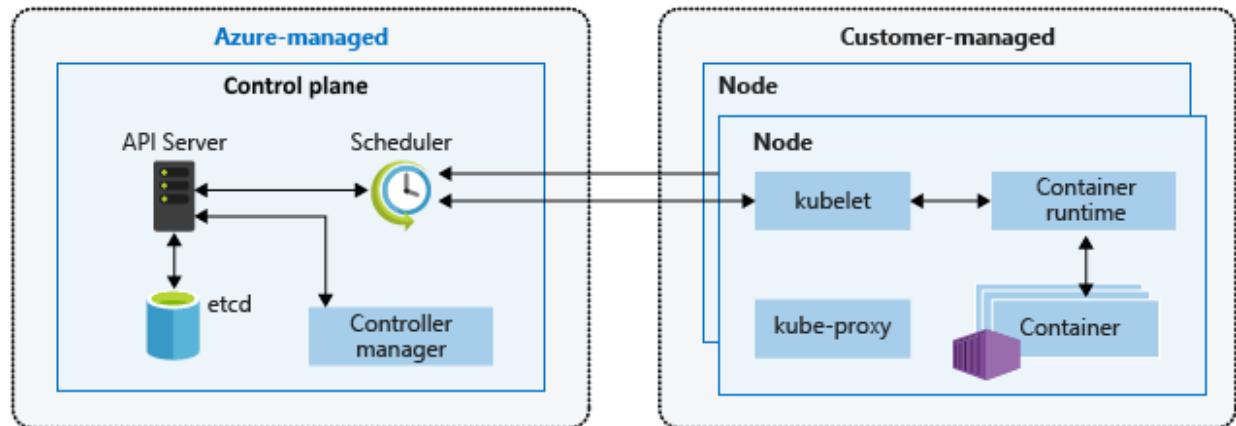


## Azure Kubernetes Services Concepts

## Kubernetes cluster architecture

A Kubernetes cluster is divided into two components:

- ★ *Control plane* nodes provide the core Kubernetes services and orchestration of application workloads.
- ★ *Nodes* run your application workloads.



### Control plane

When you create an AKS cluster, a control plane is automatically created and configured. This control plane is provided as a managed Azure resource abstracted from the user. There's no cost for the control plane, only the nodes that are part of the AKS cluster. The control plane and its resources reside only on the region where you created the cluster.

The control plane includes the following core Kubernetes components:

- ★ *kube-apiserver* - The API server is how the underlying Kubernetes APIs are exposed. This component provides the interaction for management tools, such as `kubectl` or the Kubernetes dashboard.
- ★ *etcd* - To maintain the state of your Kubernetes cluster and configuration, the highly available *etcd* is a key value store within Kubernetes.
- ★ *kube-scheduler* - When you create or scale applications, the Scheduler determines what nodes can run the workload and starts them.
- ★ *kube-controller-manager* - The Controller Manager oversees a number of smaller Controllers that perform actions such as replicating pods and handling node operations.

AKS provides a single-tenant control plane, with a dedicated API server, Scheduler, etc. You define the number and size of the nodes, and the Azure platform configures the secure communication between the control plane and nodes. Interaction with the

control plane occurs through Kubernetes APIs, such as `kubectl` or the Kubernetes dashboard.

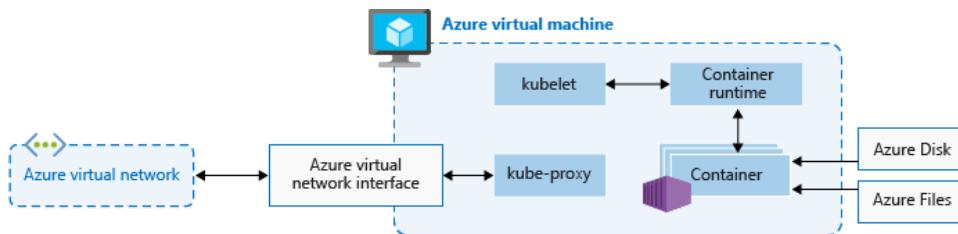
This managed control plane means you don't need to configure components like a highly available *etcd* store, but it also means you can't access the control plane directly. Upgrades to Kubernetes are orchestrated through the Azure CLI or Azure portal, which upgrades the control plane and then the nodes. To troubleshoot possible issues, you can review the control plane logs through Azure Monitor logs.

If you need to configure the control plane in a particular way or need direct access to it, you can deploy your own Kubernetes cluster using [aks-engine](#).

## Nodes and node pools

To run your applications and supporting services, you need a Kubernetes *node*. An AKS cluster has one or more nodes, which is an Azure virtual machine (VM) that runs the Kubernetes node components and container runtime:

- ★ The *kubelet* is the Kubernetes agent that processes the orchestration requests from the control plane and scheduling of running the requested containers.
- ★ Virtual networking is handled by the *kube-proxy* on each node. The proxy routes network traffic and manages IP addressing for services and pods.
- ★ The *container runtime* is the component that allows containerized applications to run and interact with additional resources such as the virtual network and storage. In AKS, Moby is used as the container runtime.



The Azure VM size for your nodes defines how many CPUs, how much memory, and the size and type of storage available (such as high-performance SSD or regular HDD). If you anticipate a need for applications that require large amounts of CPU and memory or high-performance storage, plan the node size accordingly. You can also scale out the number of nodes in your AKS cluster to meet demand.

In AKS, the VM image for the nodes in your cluster is currently based on Ubuntu Linux or Windows Server 2019. When you create an AKS cluster or scale out the number of nodes, the Azure platform creates the requested number of VMs and configures them. There's no manual configuration for you to perform. Agent nodes are billed as standard

virtual machines, so any discounts you have on the VM size you're using (including [Azure reservations](#)) are automatically applied.

If you need to use a different host OS, container runtime, or include custom packages, you can deploy your own Kubernetes cluster using [aks-engine](#). The upstream aks-engine releases features and provides configuration options before they are officially supported in AKS clusters. For example, if you wish to use a container runtime other than Moby, you can use aks-engine to configure and deploy a Kubernetes cluster that meets your current needs.

### Resource reservations

Node resources are utilized by AKS to make the node function as part of your cluster. This usage can create a discrepancy between your node's total resources and the resources allocatable when used in AKS. This information is important to note when setting requests and limits for user deployed pods.

To find a node's allocatable resources, run:

Kubectl Copy

```
kubectl describe node [NODE_NAME]
```

To maintain node performance and functionality, resources are reserved on each node by AKS. As a node grows larger in resources, the resource reservation grows due to a higher amount of user deployed pods needing management.

#### Note

Using AKS add-ons such as Container Insights (OMS) will consume additional node resources.

Two types of resources are reserved:

- ★ **CPU** - Reserved CPU is dependent on node type and cluster configuration, which may cause less allocatable CPU due to running additional features
- ★ TABLE 1

CPU cores on host	1	2	4	8	16	32	64
	60	100	140	180	260	420	740

### Memory

- ★ - Memory utilized by AKS includes the sum of two values.

- The kubelet daemon is installed on all Kubernetes agent nodes to manage container creation and termination. By default on AKS, this daemon has the following eviction rule:  $memory.available < 750Mi$ , which means a node must always have at least 750 Mi allocatable at all times. When a host is below that threshold of available memory, the kubelet will terminate one of the running pods to free memory on the host machine and protect it. This action is triggered once available memory decreases beyond the 750Mi threshold.
- The second value is a regressive rate of memory reservations for the kubelet daemon to properly function (kube-reserved).
  - 25% of the first 4 GB of memory
  - 20% of the next 4 GB of memory (up to 8 GB)
  - 10% of the next 8 GB of memory (up to 16 GB)
  - 6% of the next 112 GB of memory (up to 128 GB)
  - 2% of any memory above 128 GB

The above rules for memory and CPU allocation are used to keep agent nodes healthy, including some hosting system pods that are critical to cluster health. These allocation rules also cause the node to report less allocatable memory and CPU than it normally would if it were not part of a Kubernetes cluster. The above resource reservations can't be changed.

For example, if a node offers 7 GB, it will report 34% of memory not allocatable including the 750Mi hard eviction threshold.

$$0.75 + (0.25*4) + (0.20*3) = 0.75GB + 1GB + 0.6GB = 2.35GB / 7GB = 33.57\% \text{ reserved}$$

In addition to reservations for Kubernetes itself, the underlying node OS also reserves an amount of CPU and memory resources to maintain OS functions.

## **Node pools**

Nodes of the same configuration are grouped together into *node pools*. A Kubernetes cluster contains one or more node pools. The initial number of nodes and size are defined when you create an AKS cluster, which creates a *default node pool*. This default node pool in AKS contains the underlying VMs that run your agent nodes.

### **Note**

To ensure your cluster operates reliably, you should run at least 2 (two) nodes in the default node pool.

When you scale or upgrade an AKS cluster, the action is performed against the default node pool. You can also choose to scale or upgrade a specific node pool. For upgrade operations, running containers are scheduled on other nodes in the node pool until all the nodes are successfully upgraded.

## **Node selectors**

In an AKS cluster that contains multiple node pools, you may need to tell the Kubernetes Scheduler which node pool to use for a given resource. For example, ingress controllers shouldn't run on Windows Server nodes. Node selectors let you define various parameters, such as the node OS, to control where a pod should be scheduled.

## **Pods**

Kubernetes uses *pods* to run an instance of your application. A pod represents a single instance of your application. Pods typically have a 1:1 mapping with a container, although there are advanced scenarios where a pod may contain multiple containers. These multi-container pods are scheduled together on the same node, and allow containers to share related resources.

When you create a pod, you can define *resource requests* to request a certain amount of CPU or memory resources. The Kubernetes Scheduler tries to schedule the pods to run on a node with available resources to meet the request. You can also specify maximum resource limits that prevent a given pod from consuming too much compute resource from the underlying node. A best practice is to include resource limits for all pods to help the Kubernetes Scheduler understand which resources are needed and permitted.

For more information, see [Kubernetes pods](#) and [Kubernetes pod lifecycle](#).

A pod is a logical resource, but the container(s) are where the application workloads run. Pods are typically ephemeral, disposable resources, and individually scheduled pods miss some of the high availability and redundancy features Kubernetes provides. Instead, pods are deployed and managed by Kubernetes *Controllers*, such as the Deployment Controller.

## **Deployments and YAML manifests**

A *deployment* represents one or more identical pods, managed by the Kubernetes Deployment Controller. A deployment defines the number of *replicas* (pods) to create, and the Kubernetes Scheduler ensures that if pods or nodes encounter problems, additional pods are scheduled on healthy nodes.

You can update deployments to change the configuration of pods, container image used, or attached storage. The Deployment Controller drains and terminates a given

number of replicas, creates replicas from the new deployment definition, and continues the process until all replicas in the deployment are updated.

Most stateless applications in AKS should use the deployment model rather than scheduling individual pods. Kubernetes can monitor the health and status of deployments to ensure that the required number of replicas run within the cluster. When you only schedule individual pods, the pods aren't restarted if they encounter a problem, and aren't rescheduled on healthy nodes if their current node encounters a problem.

If an application requires a quorum of instances to always be available for management decisions to be made, you don't want an update process to disrupt that ability. *Pod Disruption Budgets* can be used to define how many replicas in a deployment can be taken down during an update or node upgrade. For example, if you have *five* (5) replicas in your deployment, you can define a pod disruption of *4* to only permit one replica from being deleted/rescheduled at a time. As with pod resource limits, a best practice is to define pod disruption budgets on applications that require a minimum number of replicas to always be present.

Deployments are typically created and managed with `kubectl create` or `kubectl apply`. To create a deployment, you define a manifest file in the YAML (YAML Ain't Markup Language) format. The following example creates a basic deployment of the NGINX web server. The deployment specifies *three* (3) replicas to be created, and requires port *80* to be open on the container. Resource requests and limits are also defined for CPU and memory.

## Azure CLI

The Azure command-line interface (Azure CLI) is a set of commands used to create and manage Azure resources. The Azure CLI is available across Azure services and is designed to get you working quickly with Azure, with an emphasis on automation.

### **Key Characteristics**

Azure CLI capabilities make it easy to work with different programming languages and software environments. For example, Azure CLI:

- ★ Is available to install in Windows, macOS, and Linux environments.
- ★ Can also be run in Docker and Azure Cloud Shell.
- ★ Offers command-line flexibility when managing an Azure solution.

- ★ Supports long-running operations.
- ★ Has the ability to use one subscription for all commands, or vary subscriptions per command.
- ★ Allows for querying of command-line results with query output returned in your format of choice.
- ★ Has the flexibility to work with multiple clouds.
- ★ Provides configurable settings for logging, data collection, and default argument values.
- ★ Is deployed with Resource Manager deployment templates.

In this quickstart, you deploy an Azure Kubernetes Service (AKS) cluster using the Azure CLI. AKS is a managed Kubernetes service that lets you quickly deploy and manage clusters. A multi-container application that includes a web front end and a Redis instance is run in the cluster. You then see how to monitor the health of the cluster and pods that run your application.

## **Creating a Voting application using AKS in the Azure CLI**

### **Create a resource group**

An Azure resource group is a logical group in which Azure resources are deployed and managed. When you create a resource group, you are asked to specify a location. This location is where resource group metadata is stored, it is also where your resources run in Azure if you don't specify another region during resource creation. Create a resource group using the [az group create](#) command.

The following example creates a resource group named *myResourceGroup* in the *eastus* location.

In the Azure CLI

```
az group create --name myResourceGroup --location eastus
```

### **Create AKS cluster**

```
az provider show -n Microsoft.OperationsManagement -o table
```

```
az provider show -n Microsoft.OperationalInsights -o table
```

Now enable the resource group and node count of the cluster in the shell

```
az aks create --resource-group myResourceGroup --name myAKSCluster --node-count  
1 --enable-addons monitoring --generate-ssh-keys
```

## **Connect to the cluster**

```
az aks install-cli
```

To configure kubectl to connect to your Kubernetes cluster, use the [az aks get-credentials](#) command. This command downloads credentials and configures the Kubernetes CLI

```
az aks get-credentials --resource-group myResourceGroup --name myAKSCluster
```

## **Run the application**

Create a yaml file named azure-vote.yaml in the cluster to write the service for the voting application the yaml file

### **Azure-vote.yaml**

[Quickstart: Deploy an AKS cluster by using Azure CLI - Azure Kubernetes Service | Microsoft Docs](#)

## **Deploy the application**

```
kubectl apply -f azure-vote.yaml
```

## **Test the application**

```
kubectl get service azure-vote-front --watch
```

**AZURE DEVOPS**

Azure DevOps is a Software as a service (SaaS) platform from Microsoft that provides an end-to-end DevOps toolchain for developing and deploying software. It also integrates with most leading tools on the market and is a great option for orchestrating a DevOps toolchain. At DevOpsGroup, we have lots of customers who have found Azure DevOps fits their needs irrespective of their language, platform.

# Introducing Azure DevOps



Azure  
Boards



Azure  
Repos



Azure  
Pipelines



Azure  
Test Plans



Azure  
Artifacts

Plan, track, and discuss work across teams, deliver value to your users faster.

Unlimited cloud-hosted private Git repos. Collaborative pull requests, advanced file management, and more.

CI/CD that works with any language, platform, and cloud. Connect to GitHub or any Git provider and deploy continuously to any cloud.

The test management and exploratory testing toolkit that lets you ship with confidence.

Create, host, and share packages. Easily add artifacts to CI/CD pipelines.

Azure DevOps comprises a range of services covering the full development life-cycle. At the time of writing these are:

- Azure Boards: agile planning, work item tracking, visualisation and reporting tool.
- Azure Pipelines: a language, platform and cloud agnostic CI/CD platform with support for containers or Kubernetes.
- Azure Repos: provides cloud-hosted private git repos.
- Azure Artifacts: provides integrated package management with support for Maven, npm, Python and NuGet package feeds from public or private sources.
- Azure Test Plans: provides an integrated planned and exploratory testing solution.

To get started with azure devops first create an account in microsoft services as free account:

<https://azure.microsoft.com/en-in/services/devops/>

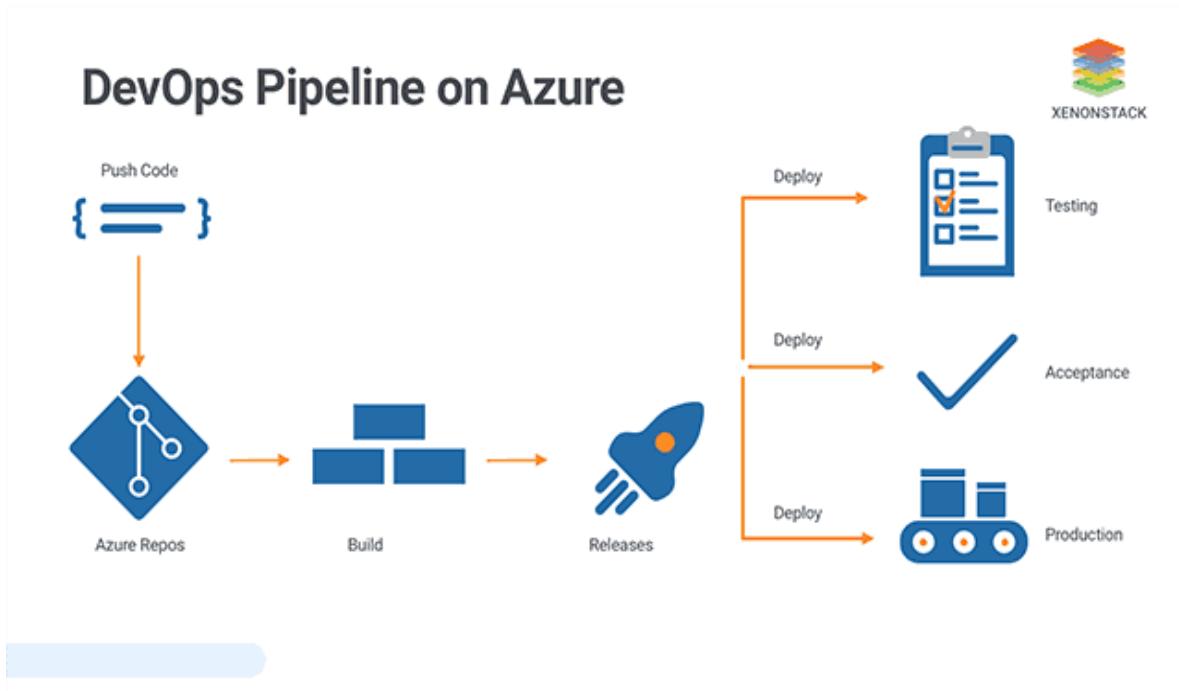
- create a free account with github signup and get started
- Give a name to your project and make it has public or private

- **Public** visibility permits users anywhere on the internet to see your board. That word public means exactly that! This is ideal for open-source projects that may need collaborators that are not part of a unified group that requires authentication.
- **Private** visibility is just that, it allows you to lock your Azure DevOps and only permit those you choose to have access. This is great for your personal projects or smaller projects that do not have a large team within your team.
- Then it will appear as follows:

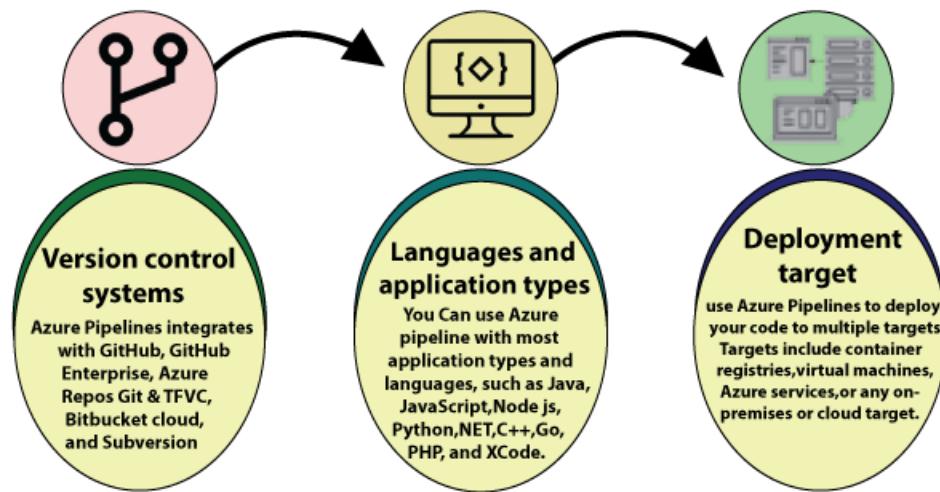
- We can now perform different tasks like creation of pipelines, boards, sprints etc

### ➤ **PIPELINE IN AZURE DEVOPS:**

**Azure Pipeline** is a cloud service that we can use to build and test our code project automatically. The Azure pipeline has a lot of capabilities such as continuous integration and continuous delivery to regularly and consistently test and build our code and ship to any target.



There are three key distinct advantages of using Azure DevOps pipelines.



**Version control system:** Azure Pipelines integrates with GitHub, GitHub Enterprise, Azure Repos Git & TFVC, Bitbucket Cloud, and Subversion.

**Language and application types:** We can use Azure Pipeline with most application types and languages, such as Java, JavaScript, Node.js, Python, .NET, C++, Go, PHP, and XCode.

**Deployment target:** We can use Azure Pipelines to deploy our code to multiple targets. Targets include - container registries, virtual machines, Azure services, or any on-premises or cloud target.

### Azure DevOps Pipeline concepts

1. **Pipeline:** It is a workflow that defines how our test, build, and deployment steps are run.
2. **Stage:** It is a logical boundary in the pipeline. It can be used to mark the separation of concerns. Each stage contains one or more jobs.
3. **Job:** A stage can contain one or more jobs. Each job runs on an agent. It represents an execution boundary of a set of steps.
4. **Step:** It is the smallest building block of a pipeline. It can either be a script or a task. A task is simply an already created script offered as a convenience to you.
5. **Agent and Agent pools:** An agent is an installable software that runs one job at a time. Instead of managing each agent individually, you organize agents into agent pools.
6. **Artifact:** It is a collection of files or packages published by a run. The Artifact is made available to subsequent tasks, such as distribution or deployment.
7. **Trigger:** It is something that is set up to tell the pipeline when to run. We can configure a pipeline to run upon a push to the repository, at scheduled times, etc.
8. **Environment:** It is a collection of resources, where you deploy your application. It contains one or more virtual machines, containers, web apps, etc.
9. **Checks:** Checks define a set of validations required before a deployment can be performed.
10. **Runs:** It represents a single execution of a pipeline and collects the logs associated with running the steps and the results of running tests.

### CREATION OF PIPELINE IN AZURE DEVOPS

STEP1: -open the github account (the account created at the time of azure devops account) and sign up with the details provided at the creation time.

Step2: - Now click on marketplace at the top and type azure pipelines in the search bar as appears below:

The screenshot shows the GitHub Marketplace search results for "azure pipelines". The search bar at the top contains the query "azure pipelines". Below the search bar, there are filters for "Types" (set to "Actions"), "Apps", and "Categories" (including API management, Chat, Code quality, Code review, Continuous integration, Dependency management, Deployment, and IDEs). The search results show four items:

- Azure Pipelines** (By Microsoft) - Continuously build, test, and deploy to any platform and cloud.
- Azure Pipelines Action** (By Azure) - Trigger a run in Azure pipelines.
- PasteICRUZE** (By salihyilmaz) - Trigger a run in Azure pipelines.
- CPods Falcon** (By CPods Falcon) - CPods Falcon is DevOps offering for easy ML Ops.

Below the results are "Previous" and "Next" navigation buttons.

Step3:- Right click on the azure pipelines (as a rocket sign) and then click on install

Step4:-And then click the (complete order begin installation) and then after the installation again provide the account details and get started.

Step 5:- copy the url and open (<https://github.com/matthewisabel/calc-stage-sample>) then click york

Step 6:-Go into the Azure DevOps project and click on pipelines. After that, click on the New pipeline button.

Step 7:- Then it will ask you (where is your code) click on github and then click on the file in the below and click fork.

Step8:-now to (configure your pipeline) click on node.js and then save and run the code.

Step9:-After the completion (job) option will appear in the below click and let the process get check and you should get the green ticks (says that everything is correct in the code).

Step10:-Now your pipeline is created and will appear as in the below figure:

The screenshot shows the Azure DevOps Pipelines dashboard. On the left, there is a sidebar with navigation links: Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and a link to "Private editions". The main area is titled "Pipelines" and shows a list of "Recently run pipelines". The list includes the following entries:

Pipeline	Last run	
kavithadocker.calc-stage-sample (4)	#20210205.1 • Set up CI with Azure Pipelines Individual CI for master	3m ago 23s
kavithadocker.calc-stage-sample (3)	#20210203.2 • testcases Manually triggered for master	Wednesday 26s
kavithadocker.calc-stage-sample (2)	#20210203.1 • Set up CI with Azure Pipelines Individual CI for master	Wednesday 23s
kavithadocker.calc-stage-sample	#20210203.2 • Update arithmeticController.js Manually triggered for master	Wednesday 26s

If your code is not correct and if u get the red tick while checking the job then steps to be followed are:

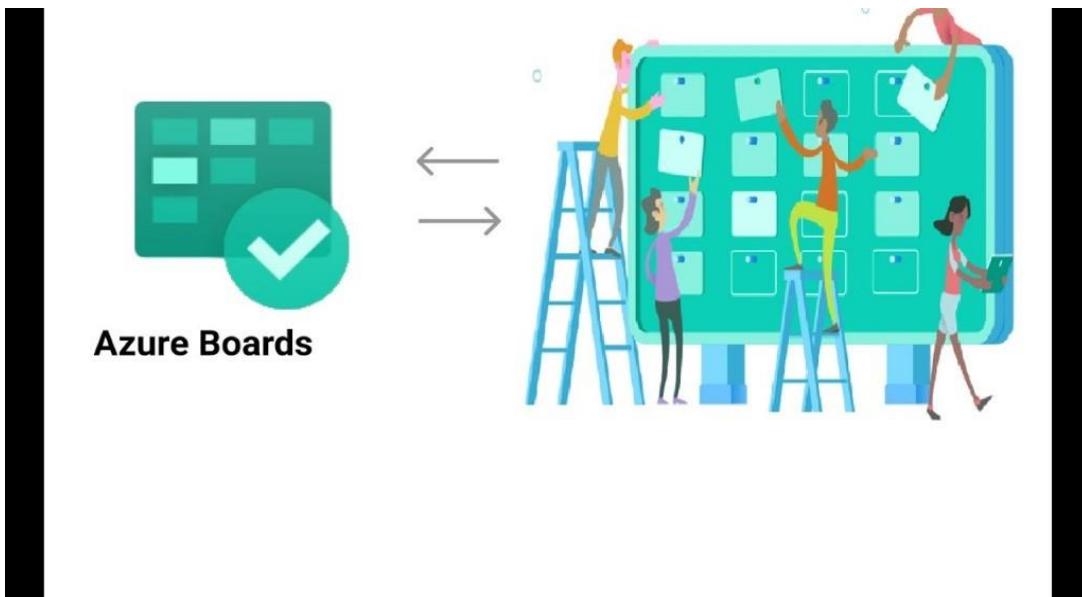
Step1:-click on the red tick and it will take you to the github account there check the code again and if any correction has to be done then

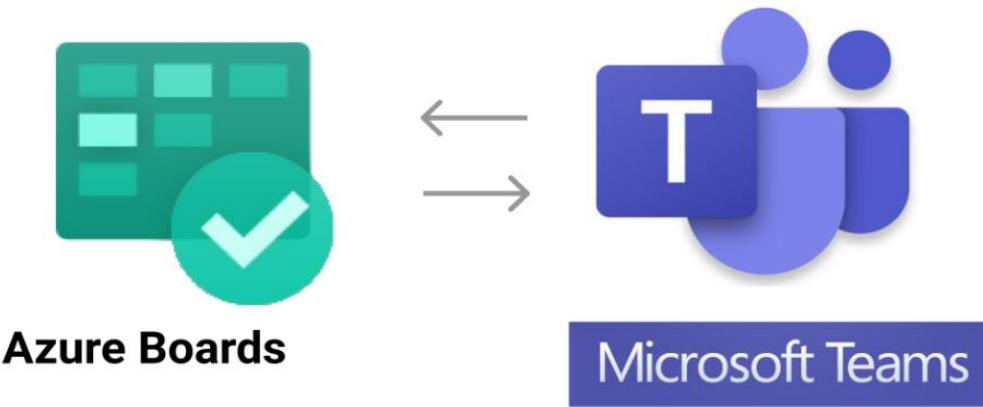
Step2:-click on azure-pipelines.yml file and edit the file by right click on the pencil option on the left of the page and after correction is completed

Step3:- Below the page click on the commit changes(you can name the change).

Step 4:- In this way we can make or correct the changes into our files if required.

- **BOARDS IN AZURE DEVOPS:-**





You'll see "Boards" on the left side of your Azure DevOps project. Click the "Boards" icon and you'll see the section where your work items are held.

Work items are the things you actually want to accomplish that are represented by identification numbers that Azure DevOps assigns. How you group these are completely up to your project management style.

ID	Title	Assigned To	State	Area Path
1	new	Unassigned	To Do	azureddevops

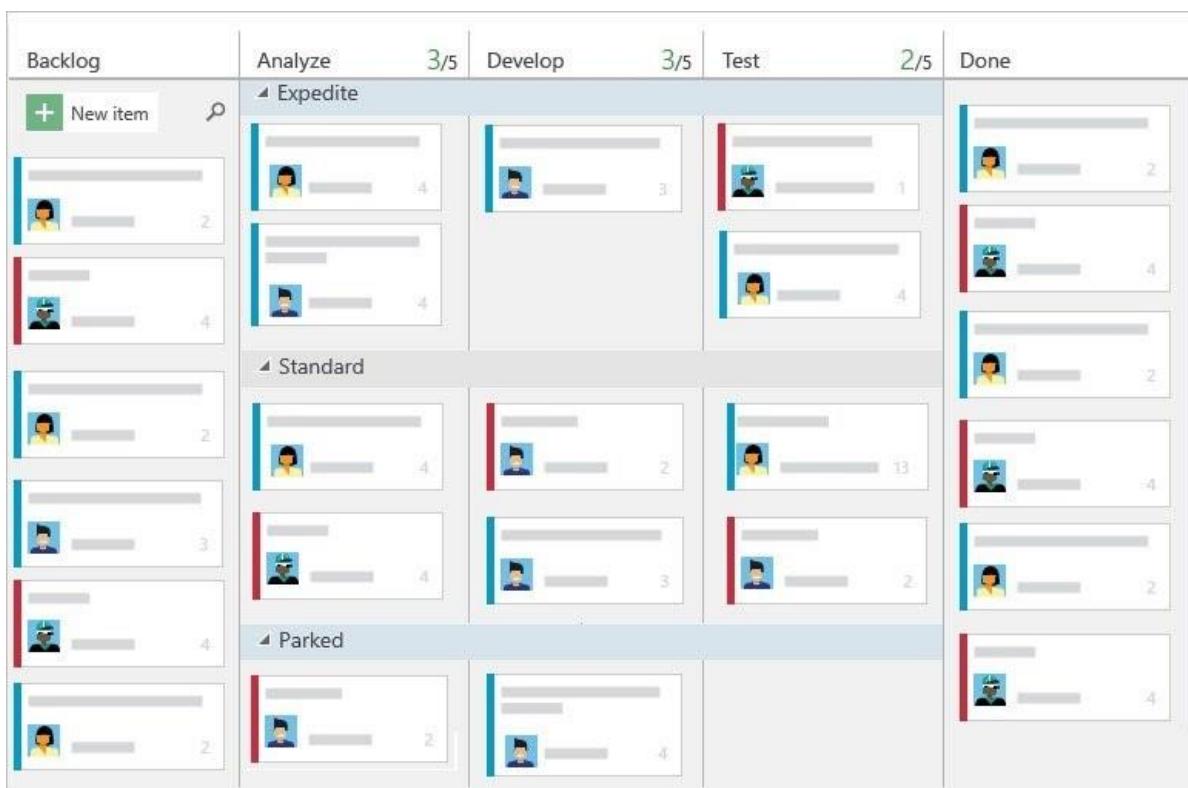
Here's an example of an Agile process from the Azure DevOps documentation ([Track work with user stories, issues, bugs, features, and epics](#)) that shows how to group Epics and Features for your work items:

You can see all the options here to add labels, give a state of the request, add related

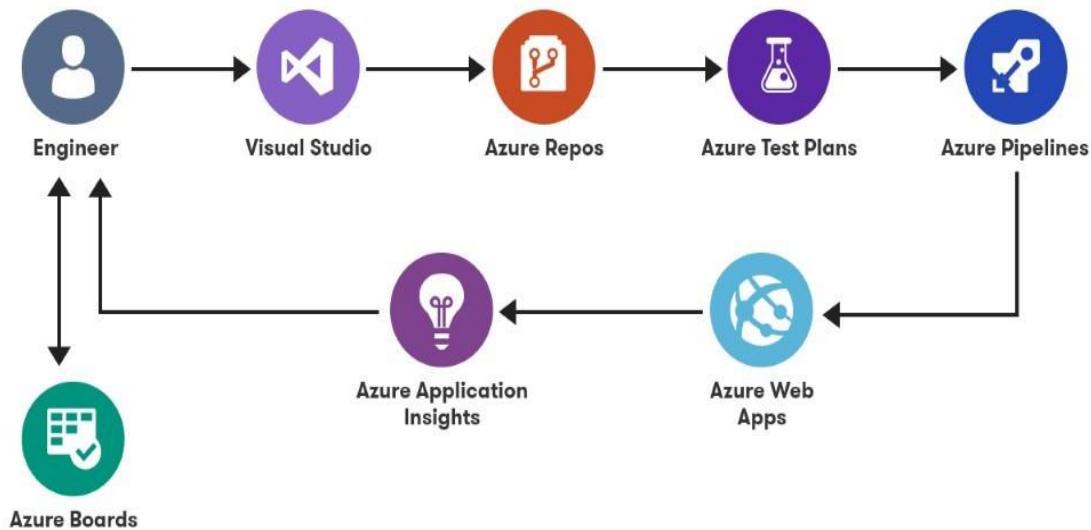
work, and any details you may need to record on the work item. Teams can customize these fields to ensure specific data is captured.

This work-item I've created for this shows the main criteria required for planning a new part of the project. This includes a unique work-item ID number (for some you may call these ticket numbers), the subject of the work-item, who it's assigned to, the state, and of course any other details we'll need to complete the task.

This example shows you a number of different lanes created for different parts of the project that contains different stages of development (Analyze, Develop, Test, and Done):



Azure DevOps provides you with options beyond just [Continuous Integration and Continuous Delivery](#). [Azure DevOps](#) provides you with [Azure Boards](#) in order to help provide a place for you to begin tracking your work with a “drag-and-drop” interface right in your browser. [Azure Boards](#) allows you to filter to individual users, export data into calendars, plan sprints and even will let you query for your work items.



**Azure Devops:** Azure DevOps is a Software as a service (SaaS) platform from Microsoft that provides an end-to-end DevOps toolchain for developing and deploying software. It also integrates with most leading tools on the market and is a great option for orchestrating a DevOps toolchain.

It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches.

You can use all the services included with Azure DevOps, or

Azure Boards. Plan, track, and discuss work across your teams.

Azure Pipelines. Continuously build, test, and deploy to any platform and cloud.

Azure Repos. Get unlimited, cloud-hosted private Git repositories for your project.

**Azure Boards:** Azure Boards is a service for managing the work for your software projects. Teams need tools that flex and grow. Azure Boards does just that, bringing you a rich set of capabilities including native support for Scrum and Kanban, customizable dashboards, and integrated reporting.

Kanban is a method for managing the creation of products with an emphasis on continual delivery while not overburdening the development team.

**Azure Repos:** Azure DevOps repos are a set of repositories that allow you to version control and manage your project code. It helps to work and coordinate code changes across your team. It will allow you to monitor code, solutions, builds, commits, pushes, PR's (Pull requests) and branching information about projects.

Azure Repos is a set of version control tools that you can use to manage your code. ... Use version control to save your work and coordinate code changes across your team. Even if you're just a single developer, version control helps you stay organized as you fix bugs and develop new features.

**Azure Pipelines:** Azure Pipeline is a cloud service that we can use to build and test our code project automatically. The Azure pipeline has a lot of capabilities such as continuous integration and continuous delivery to regularly and consistently test and builds our code and ship to any target.

Stages are the major divisions in your release pipeline: "run functional tests", "deploy to pre-production", and "deploy to production" are good examples of release stages.

- **Azure Test Plans:** Azure DevOps Test Plan provides all the tools you need to successfully test your applications. Create and run manual test plans, generate automated tests and collect feedback from users.

**Azure Artifacts:** Azure Test Plans also provides a browser extension for exploratory testing and gathering feedback from stakeholders

## SQL

**SSMS:** SQL server management studio

**SQL Server Management Studio (SSMS)** is an IDE that provides a graphical interface for connecting and working with MS SQL server. It was launched with Microsoft SQL Server 2005 and is used for configuring, managing, and administering all components within Microsoft SQL Server.

### Installation:

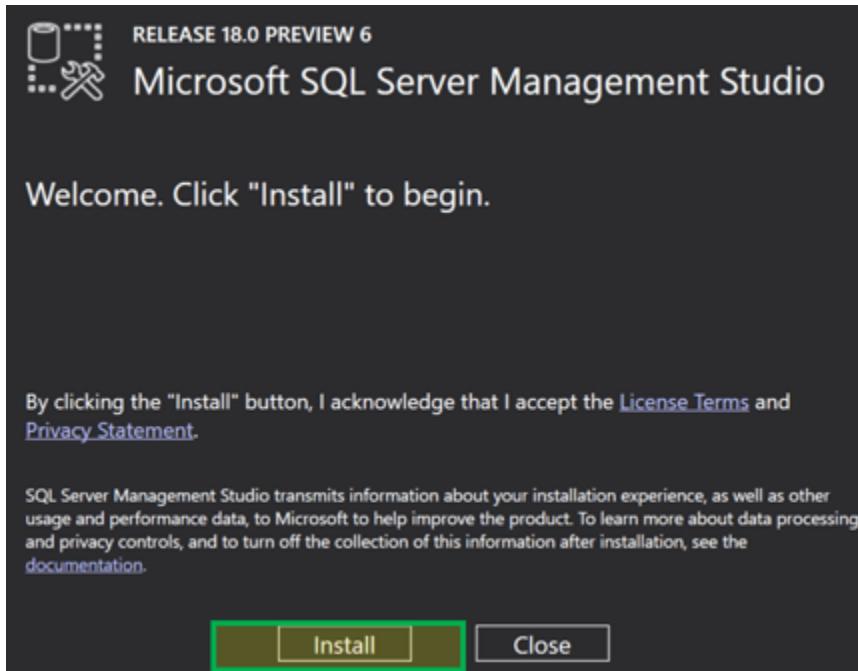
1. **Select Custom**
2. **Windows option**
3. **License Agreement**
4. **Login: sa and password (using mixed mode authentication.... sql express)**
5. **Restore database ->device->browse->**
6. **SSMS installation**

## **Download and Install SQL Server Management Studio:**

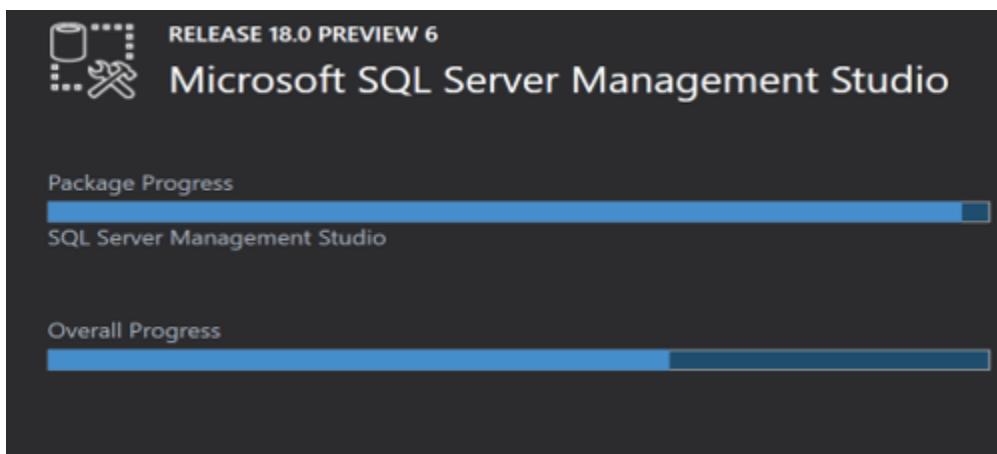
**Step 1)** Go to the link and click on Download SQL Server management Studio 18.0 (preview 6) for SSMS Download

**Step 2)** Once downloaded we will get a .exe file named as "**SSMS-Setup-ENU.exe.**" Double click on it

**Step 3)** Below window will appear. Click on '**Install**' button to install SQL Server Management Studio (SSMS) on your system



**Step 4)** Installation will begin. Below screen will show **Packages progress** and **Overall Progress**.



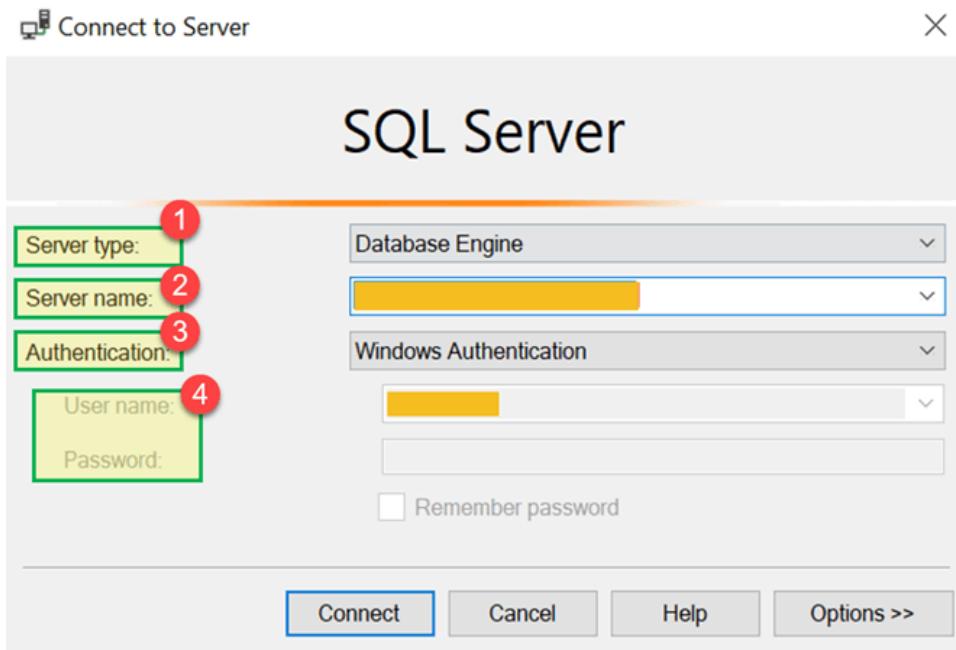
**Step 5)** Once **Completed**, SSMS Setup will show the below screen with "Setup Completed" message.



#### **How to access "Management Studio."**

Now, we're ready to open "Management Studio. Go To Start Menu>Programs>Microsoft SQL Server Tools 18> Microsoft SSMS 18.

Below '**Connect to Server**' screen will appear. Server Name defaults to the name selected while installing MS SQL server.



**Server type:** This is an option to select one out of four available MS SQL services option. We will be working on 'Database Engine' for creating and working with Database. Other Server type includes Analysis, Reporting & Integration Services.

**Server name:** This is Server's name where MS SQL Server is installed and need to establish the connection with that server. Generally, we use the server name as "Machine name\Instance." Here Instance is the name given to SQL Server instance while SQL server installation.

**Authentication:** This is defaulted to "Windows Authentication" if we use "Windows Authentication" during SQL Server Installation. Else, if we select 'Mixed Mode (Windows Authentication & Windows Authentication)' then Authentication will be defaulted to "SQL Server Installation."

User name\Password: If Authentication is selected other than "Windows Authentication" like "SQL server Installation" then these two fields will be required.

Click on 'Connect.' Now you will be connected to 'Data Management Studio.'

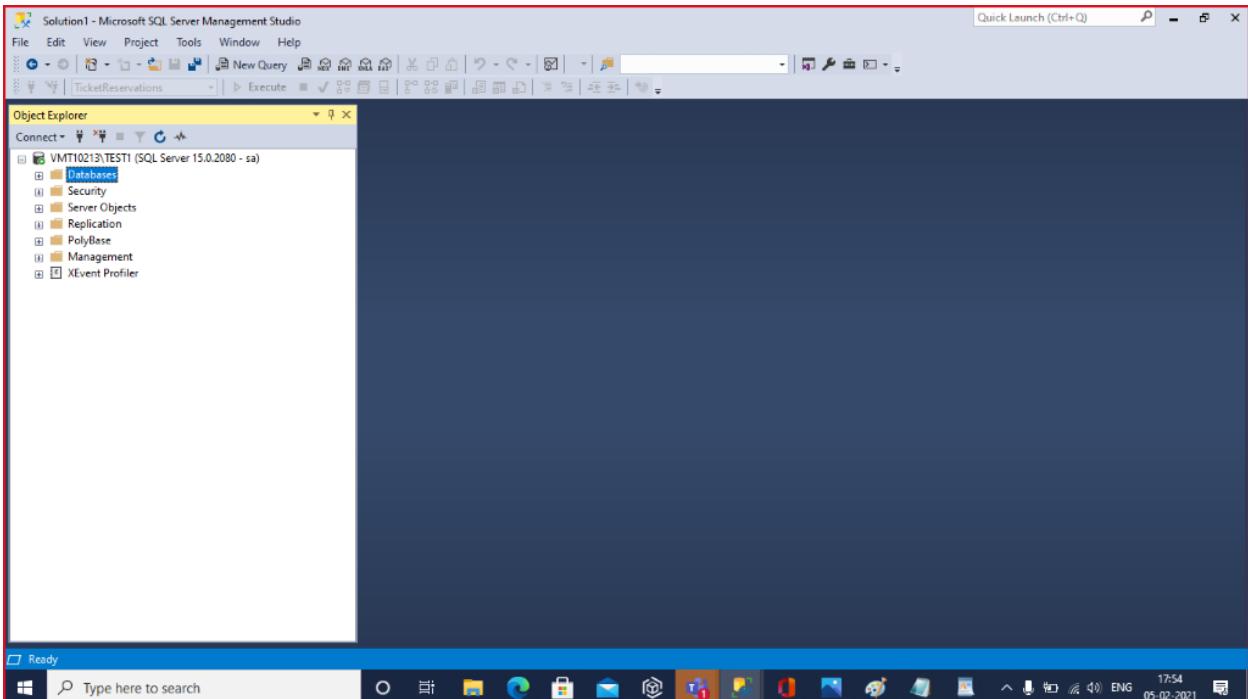
#### **Access "Management studio" using Command line.**

Alternatively, we can also open MS SQL Management Studio using Window Command line. You need to have full path of ssms.exe. Below is default location and file name:

#### **Path:**

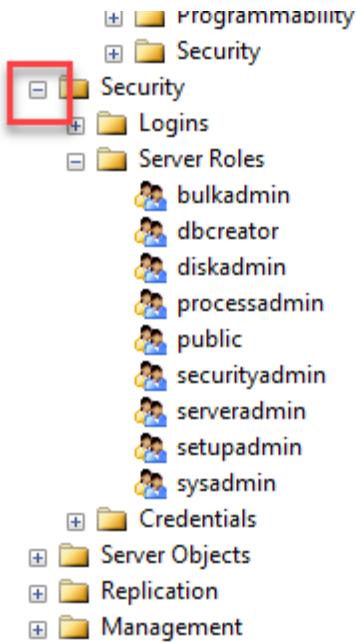
C:\Program Files (x86)\Microsoft SQL Server Management Studio 18\Common7\IDEExe  
name: ssms.exe

'Connect to Server' screen will appear



### Object Explorer:

The Object Explorer provides a tree view of the database objects contained in the server. This section shows all the Databases, Security, Server Object for quick reference. To view the components of each object, just click the + icon located to the left of the object which will expand it.



## **2) Databases Selection Dropdown**

This dropdown allows the user to select the Database in which we will be running our queries.

## **3) Query Editor**

Here we can write all our queries. MS SQL server provides interactive suggestions for tables, columns, etc. for easy queries creations and much more.

## **4) Execute button**

This button will finally execute the query and return the results.

### **SSMS Tips and Issues:**

- Management Studio is a standalone product. It does not correspond to any specific version of SQL Server. For example, we can use SSMS Version 18 with SQL Server 2017, SQL Server 2016 as well.
- Large codes reduce readability. Use comments for better Readability. Put "--" in front of any line to comment it out.
- Group comment: We can comment out the group of line by selecting them all by /\* \*/
- Table creation

Create table tablename (parameters datatypes)

- Insertion

`Insert into tablename values(parameters_values)`

- adding multiple column

`alter table tablename add columnname datatype`

- Update

`update tablename`

`Set columnname =value where conditions`

- Deleting.

`delete from tablename where conditions`

- Selection

`select * from tablename`

Select parameterlist from tablename                    optional-----(where condition)

### **Different constraints:**

1)primary key- key that is unique for each record. used to recognize each column. It wont accept null values .It can be given by NOT NULL

`(empid int identity(1,1) primary key NOTNULL ,empname varchar(10))`

2)foreign key – key that links two table together (referencing key) .It accepts only one null values

3)unique key – two different rows must not have same value.it accepts null values also

4)check ident – it is used to check the current identity value for specified table.

`dbcc checkident('student')`

`dbcc checkident('student',reseed=6)`

If we want to use manual identity we have to use dbcc  
`checkident(presentident,newident)`

5)default- if no value is specified default value is added

`Alter table tablename add default(columnname)`

6)not null – if this is mentioned then it should be entered with value definitely.(insert ,update)

7)index- two types char index ,pat index

Select charindex(' ','ab cd') ans: 3 (findout characters)

Select patindex((%[f-z]%,abcd12cgh') ans: 8

- Primary key creates clustered index and other key creates Non clustered index.
- If table is absent then it display intelliscence value(not exist) and it also specifies error if it is on .
- Set identity insert tablename on --- with this we can directly insert insert.
- In index we have char index,pat index.pat index(search for characters mentioned in range returns the first appear value) ex(%[f-z]%,abcd12fgh ---7)
- Date ,time ,place— getdate(),gettime(),getplace()
- dbcc checkident('employee')
- Dbcc checkident('employee',reseed=max value)-----( continuation--it will give value from next number itself)
- Select \* from databasename.schema name.tablename
- Eg: select \*advworks.[hr].[employee]
- Comment for single line (--)
- Comment for multiple line (\\*, \*\`)
- Cast(from one data type to other type)
- Convert(one data type to other but using date and time format)

#### **Difference between BLANK and NULL:**

Blank is a string of zero characters. NULL is nothing. In the below query output you can see that length of blank gives 0 and length of NULL as NULL which is nothing.

```
;WITH CTE AS ( SELECT " AS BLANK, NULL AS [NULL] )
```

```
SELECT LEN(BLANK),LEN([NULL]) FROM CTE
```

#### **Difference between Truncate and Delete:**

- Truncate is a DDL command, whereas Delete is a DML command. This is because, suppose a table has identity column and if we Truncate table and insert a record its identity property is reset i.e. count starts from beginning. It means the table structure is

reset. Where as in case of delete if we delete and insert a record the identity column value will be one more than the value last deleted.

- Truncate and Delete both is logged. Where Delete logs the whole record deleted but Truncate logs pointer to the data page. Because of this Truncate is faster and Delete is slower.
- Truncate is bulk operation whereas Delete is Row by Row operation.
- In Truncate we cannot specify where condition, where as in Delete we can specify where condition.

**Commit:** temporary saving the transaction

**Rollback:** it can retrieve till the last commit happened

Savepoint is like checkpoint (if any data lost the transactions upto the savepoint data can be retrieved)

Desc: descending order

Asc: ascending order

Rename:

Alter table tablename

rename oldcolumn newcolumn datatype

### **Exercises:**

Create two different tables, employee, your choice table

1) Insert all rows with same name for at least 2 names

inserting names with same name

---create table for employee

```
create table employee (empid int identity(1,1) primary key, empname varchar(50), phone
```

```
decimal,zip int)
```

```
insert into employee values('sravani',7683479040,123456)
```

```
select * from employee
```

```
create table student(stdid int identity(1,1) primary key, stdname varchar(50), phone decimal, zip int)
```

```
insert into student values('sravani', 1234566543, 123456)
```

```
select * from student
```

```
select * from employee
```

## **2) Add multiple columns with different data types and see differences.**

```
create table student add loc varchar (50), email varchar(50), dept nvarchar (50)
```

```
update student set loc='hyd', email='sravani@gmail.com', dept='ece' where stdname='sravani'
```

```
Select datalength(columnname) from tablename where condition
```

```
select * from student
```

## **3) Define table with primary key and identity column**

```
create table student(stdid int identity(1,1) primary key, stdname varchar(50), phone decimal, zip int)
```

## **4) After inserting 10 rows, delete last five rows**

```
delete from student where stdid >= 5
```

## **5) Check inserting new rows and verify identity values is proper or not**

```
insert into student values
```

```
('pranitha', 1234567890, 123456, 'hyd', 'pranitha@gmail.com', 'ece')
```

```
select * from student
```

```
insert into student values('a', 4534534532, 123456, 'hyd', 'a@gmail.com', 'ece')
```

```
insert into student values('b', 4534534532, 123456, 'hyd', 'b@gmail.com', 'ece')
```

```
insert into student values('c', 4534534532, 123456, 'hyd', 'c@gmail.com', 'ece')
```

```
insert into student values('d', 4534534532, 123456, 'hyd', 'd@gmail.com', 'ece')
```

```
select * from student
```

## **6) Resolve identity values property issue after deleting rows.**

```
delete from student where stdid = 6
```

```
insert into student values ('g',5678900332,123456,'hyd','g@gmail.com','ece')

dbcc checkident('student')

dbcc checkident('student','reseed',6)
```

- **Date part – eg:day(20210122)**

#### Syntax:getdate()

- SELECT SUBSTRING('abcde',1, 3); -- 'abc'
- SELECT LEFT('abcde', 2); -- 'ab'
- SELECT RIGHT('abcde', 3); -- 'cde'
- SELECT CHARINDEX('k','Itzik Ben-Gan'); -- 6
- SELECT PATINDEX('%[f-z]%', 'abcd123efgh'); -- 5
- SELECT LEN('xyzkjklj'); -- 3

#### Length of the character

- SELECT DATALENGTH('\n xyz'); -- 6  
( size of the character eg :char 2 bytes)
- SELECT REPLACE('.1.2.3.', '.', '/'); -- '/1/2/3/'
- SELECT REPLICATE('0', 5); --00000
- SELECT STUFF('x,y,z', 1, 2, 'VALUEM-'); -- 'VALUEM-y,z'

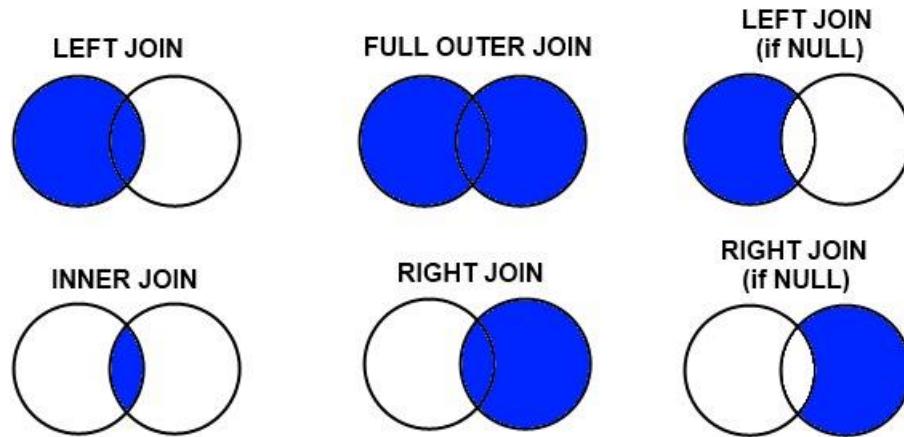
#### Searching the character and replacing till the startvalue to particular position

- SELECT UPPER('aBcD'); -- 'ABCD'
- SELECT LOWER('aBcD'); -- 'abcd'
- SELECT RTRIM(LTRIM(' xyz ')); -- 'xyz'
- SELECT LTRIM(' xyz '); -- 'xyz '
- SELECT FORMAT(1759, '000000000'); -- '0000001759'

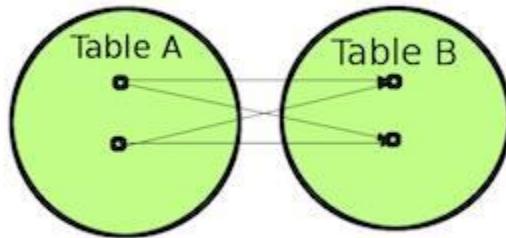
#### Joints: to retrieve data from two or more tables

1. Inner join: common between two tables
2. Left join: shows common and left content table value

3. Right join: shows common and right content table value
4. Full join: combination of two tables (left and right join combination)



5. Cross join: one table row with all rows in other table (cartesian product)



6. Self join: table with itself eg:a \*a

In ,not in , on ,between ,and ,unique,or,NOTNULL, are used and relational operators also used(where conditions)

**Ranking: it gives ranking to rows based on particular value mentioned.**

**Types of ranking:**

1. Rank(): skipping of rank numbers done
2. Row\_number(): it is given different rank
3. Dense\_rank(): no skipping of ranks
4. Ntile(): dividing equally into the parts mentioned in parenthesis.

Syntaxes:

```
ROW_NUMBER() over (partition by [BusinessEntityID] order by rate desc) as RowNumber,  
rank() over (partition by [BusinessEntityID] order by rate desc) as RankNumber,  
dense_rank() over (partition by [BusinessEntityID] order by rate desc) as DenseRankNumber,  
Ntile(6) over (partition by [BusinessEntityID] order by rate desc) as NtileNumber
```

Example:

```
select * from ( SELECT [BusinessEntityID] ,[RateChangeDate] ,[Rate] ,[PayFrequency]  
,[ModifiedDate], ROW_NUMBER() over (partition by [BusinessEntityID] order by rate  
desc) as RowNumber, rank() over (partition by [BusinessEntityID] order by rate desc) as  
RankNumber, dense_rank() over (partition by [BusinessEntityID] order by rate desc) as  
DenseRankNumber, Ntile(6) over (partition by [BusinessEntityID] order by rate desc) as  
NtileNumber FROM [AdventureWorks2016].[HumanResources].[EmployeePayHistory] )  
as D where RowNumber = 1
```

- Partition by clause: divided and generates based on columns
- Over, orderby has to be used in the syntax.

Without cte:

```
select * from  
(  
SELECT [BusinessEntityID] ,[RateChangeDate] ,[Rate] ,[PayFrequency] ,[ModifiedDate],  
ROW_NUMBER() over (partition by [BusinessEntityID] order by rate desc) as  
RowNumber,  
rank() over (partition by [BusinessEntityID] order by rate desc) as RankNumber,  
dense_rank() over (partition by [BusinessEntityID] order by rate desc) as  
DenseRankNumber,  
Ntile(6) over (partition by [BusinessEntityID] order by rate desc) as NtileNumber  
FROM [AdventureWorks2016].[HumanResources].[EmployeePayHistory]
```

) as D

where RowNumber=1

With cte: virtual function

Cte=common table expression

with CTE\_1 as

(

SELECT [BusinessEntityID]

, [RateChangeDate]

, [Rate]

, [PayFrequency]

, [ModifiedDate],

ROW\_NUMBER() over (partition by [BusinessEntityID] order by rate desc) as

RowNumber,

rank() over (partition by [BusinessEntityID] order by rate desc) as RankNumber,

dense\_rank() over (partition by [BusinessEntityID] order by rate desc) as

DenseRankNumber,

Ntile(6) over (partition by [BusinessEntityID] order by rate desc) as NtileNumber

FROM [AdventureWorks2017].[HumanResources].[EmployeePayHistory]

)

select \* from CTE\_1 where RowNumber = 1

### **Functions:**

- Set of sql statement used to perform specified tasks like data searching, calculations. The content present in function is reused .
- It has to be executed with **select** statement.
- It takes input in parameters and return values

Types of functions:

1. Built in functions: predefined and cannot be modified.

2. User defined functions:

Example: `create function ufn_findsalary()`

Types of User defined or inline functions

1) scalar valued: returns only single value

Syntax:

**1) CREATE FUNCTION to display card type of given credit card no. E.g.  
400000000022 for this credit card, display result as VISA**

go

```
create function ufn_findcardtype (@cardnumber decimal)
```

```
returns varchar(20)
```

As

```
begin
```

```
declare @cardtype varchar(20)
```

```
SELECT @cardtype=CardType FROM credit_card_List where @cardnumber between  
StartRange and EndRange
```

```
return @cardtype
```

```
end
```

```
Go
```

```
select dbo.ufn_findcardtype(400000000022)
```

2) table valued: returns table data type

**create table valued function to display card ranges and card type based on  
parameter with credit card no**

**e.g.** E.g. 400000000022 for this credit card, display result as 400000000000  
499999999999 VISA

```
create function ufn_findtable(@cardnumber decimal)
```

```
returns @table1 table(startrange decimal ,EndRange decimal,CardType varchar(20)) as  
begin
```

```
insert into @table1
```

```
select startrange,EndRange,CardType from credit_card_List ---selection order
```

```
where @cardnumber between starrange and EndRange  
return  
end  
go
```

**Create user-defined (table-valued function) which finds the name of a vendor using vendorid with-out parameter**

```
alter function ufn_vendorname ()  
returns @table table(vendorid int,vendorname varchar(50))  
as  
begin  
declare @vendorname varchar(50)  
insert into @table select vendorid,vendorname from mcnvendors  
return  
end  
Go  
select * from dbo.ufn_vendorname()
```

**Create user-defined (table-valued function) which finds the name of a vendor using vendorid with parameter**

```
alter function ufn_vendorname (@vendorid int)  
returns @table table(vendorid int,vendorname varchar(50))  
as begin  
declare @vendorname varchar(50)  
insert into @table select vendorid,vendorname from mcnvendors where  
@vendorid=vendorid return  
End
```

```
go  
select * from dbo.ufn_vendorname(22)
```

3)scalar and tabled function:

**Create user-defined (Scalar-valued function) which finds the name of a vendor using vendorid**

```
create function ufn_findname(@vendorid decimal)  
returns varchar(20)  
as  
begin  
declare @vendorname varchar(20)  
select @vendorname = vendorname from mcnvendors  
where @vendorid = vendorid  
return @vendorname  
end  
go  
select * from dbo.mcnvendors  
where vendorname=dbo.uf Aggregate n_findname(20)
```

**View: data retrieving from linked tables or any single table as per our convience**

1. Complexity reduces for multiple joints
2. Increases Data level security

Create view vw\_testview as

```
Select * from testdemo1
```

```
Select * from vw_testview
```

```
Alter table testdemo1
```

```
Drop column phone
```

```
Alter view vw_testview with schemabinding as
```

**Two types of views:**

1)standard view or normal view: data is not stored in the disk and view is noticed while executing

2) index view :data is stored in disk and form new index view .it has to use unique clustered index so that we can create non clustered index

### **Schema binding:**

If we create a view with schema binding option it will lock the tables referred by a view and restrict the changes that may change the table schema.(no alter command). While creating schema binding view, we can't mention "SELECT \* FROM TABLENAME" with the query. We have to mention the entire column name for reference

**Group by :** Group BY is used to summarize the table data. As the name implies it will group the set of similar data in a column. Grouping may be based on only one column or multiple columns.

SELECT column1, column2

FROM table\_name WHERE [ conditions ]

GROUP BY column1, column2

ORDER BY column1, column2

Types:

1)single column grouping

2)multiple column grouping

### **Order by:**

The ORDER BY command is used to sort the result set in ascending or descending order.

The ORDER BY command sorts the result set in ascending order by default. To sort the records in descending order, use the DESC keyword.

### **Pivoting:**

It provides an easy mechanism in Sql Server to transform rows into column

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer pane on the left displays the database structure for 'VM10126.TEST1'. The main query window contains the following T-SQL code:

```
WITH PivotData AS
(
    SELECT
        custid, -- grouping column
        shipperid, -- spreading column
        freight -- aggregation column
    FROM Sales.Orders order by custid
)
SELECT custid, [1], [2], [3]
FROM PivotData
PIVOT(SUM(freight) FOR shipperid IN ([1],[2],[3])) AS P
order by custid;

select * from [Adventureworks2019].[HumanResources].[Employee]

with PivotData as (
SELECT
    [MaritalStatus]
    ,[Gender]
    ,YEAR([HireDate]) as HireYear, count(*) as TotalCounts
    FROM [Adventureworks2019].[HumanResources].[Employee]
    group by [MaritalStatus],YEAR([HireDate])
    ,[Gender]
    ,YEAR([HireDate])
)
select HireYear, [M].[S] into #tmp
from PivotData
pivot (sum(TotalCounts) for [MaritalStatus] in ([M].[S])) as P

select HireYear, isnull([M],'0') as M, isnull([S],'0') as [S] from #tmp

with PivotData as (
SELECT
    [Gender]
```

2) find employees married/unmarried

With pivotdata as

```

SELECT
[MaritalStatus]
-- ,[Gender]
,Year ([HireDate]) as HireYear, count (*) as TotalCounts
FROM [AdventureWorks2017].[HumanResources].[Employee]
group by [MaritalStatus],Year ([HireDate])
-- ,[Gender]
,Year ([HireDate)))
select HireYear, [M],[S] into #tmp1
from PivotData
pivot (sum(TotalCounts) for [MaritalStatus] in ([M],[S])) as P
select HireYear, isnull([M],'0') as M,isnull([S],'0') as [S] from #tmp1

```

6 ▾

HireYear	M	S
2006	1	0
2007	2	4
2008	37	37
2009	73	75
2010	21	17
2011	8	8
2012	3	1
2013	1	2

## Unions:

```

select country ,region,city
from sales.customers
union
select country,region,city
from hr.Employees

```

## Union all:

```

query4.sql - VM....lSQL2012 (sa (/!))*
select country ,region,city
from sales.customers
union all
select country,region,city
from hr.Employees

```

### **Intersect:**

```
select country ,region,city  
from sales.customers  
interesect  
select country,region,city  
from hr.Employees
```

### **Except:**

```
select country ,region,city  
from sales.customers  
except  
select country,region,city  
from hr.Employees
```

### **Offset and fetch:**

```
-- skip 50 rows, fetch next 25 rows  
SELECT country,region,city  
FROM Sales.customers  
ORDER BY country  
OFFSET 20 ROWS FETCH NEXT 10 ROWS ONLY
```

country	region	city
Denmark	NULL	Århus
Finland	NULL	Oulu
Finland	NULL	Helsinki
France	NULL	Lille
France	NULL	Nantes
France	NULL	Nantes
France	NULL	Marseille
France	NULL	Strasbourg
France	NULL	Versailles
France	NULL	Toulouse

### **Stored procedures:**

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.

So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

It should have exec statement .

```
select * from [AdventureWorks2017].[HumanResources].[Employee] as employeetable
select * into employeetable1 from [AdventureWorks2017].[HumanResources].[Employee]

go
create procedure newprocedure3 @businessEntityID int, @Birthdate date, @Hiredate date ,@vacationhours smallint
as
begin
update employeetable1
set BirthDate=@Birthdate,HireDate=@Hiredate ,vacationHours=@vacationhours
where BusinessEntityID= @businessEntityID
end

exec newprocedure3 2,'1958-08-01','2000-08-01',2
select * from employeetable1
```

```
Query9.sql - VM...T1.master (sa (57))*  ▶ ×
select * from [AdventureWorks2017].[HumanResources].[Employee] as employeetable
select * into employeetable1 from [AdventureWorks2017].[HumanResources].[Employee]

go
create procedure newprocedure4 @businessEntityID int, @Birthdate date, @jobtitle varchar(20) ,@vacationhours smallint
as
begin
update employeetable1
set BirthDate=@Birthdate,JobTitle=@jobtitle ,vacationHours=@vacationhours
where BusinessEntityID= @businessEntityID
end

exec newprocedure4 2,'1958-08-01','salesofficier',2
select * from employeetable1
```

### Declaring function inside a procedure:

```
alter proc usp_updatecardtypes as
```

```
declare @cardnum decimal
declare @start int, @end int
```

```
select @start=min(CreditCardID),@end=max(CreditCardID) from NewCarddatatest1
```

```
while (@start <= @end)
begin
select @cardnum = CardNumber from NewCarddatatest1 where CreditCardID = @start
update NewCarddatatest1
set Cardtype = dbo.fn_findcardtype (@cardnum)
where CardNumber = @cardnum
and CreditCardID = @start

set @start = @start + 1
```

```
end

select * from NewCarddatatest1

usp_updatecardtypes

declare @cardnum decimal = 33336091960108

update NewCarddatatest1
set Cardtype = dbo.fn_findcardtype (@cardnum)
where CardNumber = @cardnum

select * from NewCarddatatest1
where CardNumber = 33336091960108
```

- **declaring Procedure in function is not possible:**

#### **Using While statement example:**

```
Q: print hello world 100 times

Declare @start int

Declare @end int

Declare@sql varchar(max)

Set@start=1 ,@end=100

While(@start<=@end)

Begin

Set sql=' Hello world' +convert(varchar(10), @start);

Print@sql

Set @start=@start+1

End
```

- While statement is best than cursor because cursor takes more storage and memory.
- Cast,convert same function
- Cast(@start as varchar(10))

- Try\_cast can be used to not get error when printing while cast is used.

### **Authentications:**

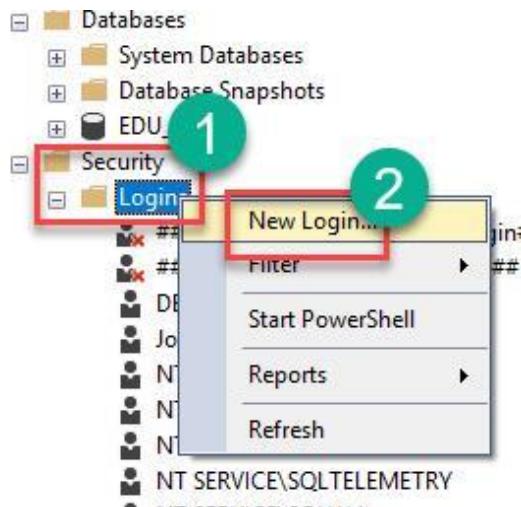
1)windows Authentication

2)sql server authentication

### **SQL server login and permission:**

**create a login, Navigate to Security > Logins**

To create a login, Navigate to Security > Logins



1.Login Name

2.Select SQL Server authentication

3.Enter Password

### **Server roles:**

**1.Bulk admin** – Bulk Creations of databases. Members of the **bulkadmin** fixed server role can run the BULK INSERT statement.

**2.DB creator-** The **diskadmin** fixed server role is used for managing disk files. Members of the **dbcreator** fixed server role can create, alter, drop, and restore any database.

**3.Disk admin:** The **diskadmin** fixed server role is used for managing disk files.

**4.Process admin:** Members of the **processadmin** fixed server role can end processes that are running in an instance of SQL Server.

**5.Security admin:** Members of the **securityadmin** fixed server role manage logins and their properties. They can GRANT, DENY, and REVOKE server-level permissions. They can also GRANT, DENY, and REVOKE database-level permissions if they have access to a database. Additionally, they can reset passwords for SQL Server logins.

IMPORTANT: The ability to grant access to the Database Engine and to configure user permissions allows the security admin to assign most server permissions. The **securityadmin** role should be treated as equivalent to the **sysadmin** role.

**6.Server admin:** Members of the **serveradmin** fixed server role can change server-wide configuration options and shut down the server.

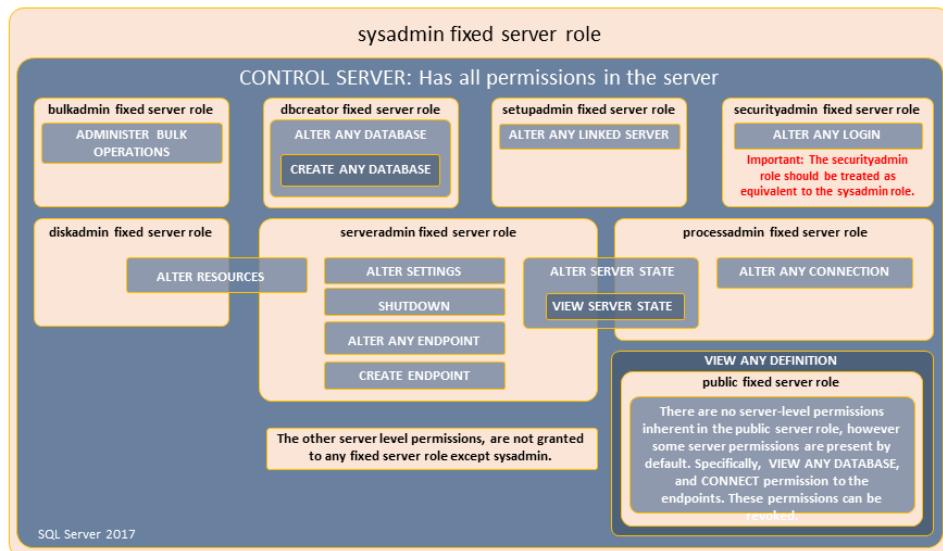
**7.Setup admin:** Members of the **setupadmin** fixed server role can add and remove linked servers by using Transact-SQL statements. (**sysadmin** membership is needed when using Management Studio.)

**8.Sys admin:** This is high role and every activity is monitored and this access is given only to the db admins. Members of the **sysadmin** fixed server role can perform any activity in the server.

Public: Every SQL Server login belongs to the **public** server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to public on that object. Only assign public permissions on any object when you want the object to be available to all users. You cannot change membership in public.

Note: **public** is implemented differently than other roles, and permissions can be granted, denied, or revoked from the public fixed server roles.

## SERVER LEVEL ROLES AND PERMISSIONS: 9 fixed server roles, 34 server permissions



## Permissions:

Every SQL Server securable has associated permissions that can be granted to a principal. Permissions in the Database Engine are managed at the server level assigned to logins and server roles, and at the database level assigned to database users and database roles.

Once you understand the permissions, apply server level permissions to logins and database level permissions users with the [GRANT](#), [REVOKE](#), and [DENY](#) statements. For Example:

```
GRANT SELECT ON OBJECT::HumanResources.Employee TO Larry;
```

```
REVOKE SELECT ON OBJECT::HumanResources.Employee TO Larry;
```

## Importing and Exporting of Data:

**Example:** From sales.order table

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer pane displays a tree view of the database structure for the 'AdventureWorksLT' database, including databases, tables, and system objects. On the right, a query window titled 'SQLQuery26.sql - V..Works2019 (ss [53])' contains a T-SQL SELECT statement. The statement retrieves data from the 'person' table, specifically columns related to names and titles. The results grid shows 10 rows of data, each representing a person's information such as BusinessEntityID, Title, FirstName, MiddleName, LastName, Suffix, JobTitle, PhoneNumber, and PhoneNumberType.

BusinessEntityID	Title	FirstName	MiddleName	LastName	Suffix	JobTitle	PhoneNumber	PhoneNumType
1	274	Stephen	Y	Jang		North American Sales Manager	230-555-0197	Cell
2	275	Michael	G	Bythe		Sales Representative	257-555-0154	Cell
3	276	Linda	C	Mitchell		Sales Representative	803-555-0116	Work
4	277	Jillian		Canson		Sales Representative	517-555-0117	Work
5	278	Genet	R	Verges		Sales Representative	922-555-0165	Work
6	279	Tsvi	Michael	Peter		Sales Representative	604-555-0112	Work
7	200	Panels	O	Annen-Wolfe		Sales Representative	340-555-0193	Cell
8	201	Shu	K	Ito		Sales Representative	300-555-0120	Cell
9	202	José	Edvaldo	Santiva		Sales Representative	105-555-0169	Work

From sales.person credit table:

ass

The screenshot shows the SQL Server Management Studio interface. On the left, there's a tree view of database objects under 'Graph Tables'. In the center, a query window displays a T-SQL script for selecting top 1000 rows from the 'customer' table. To the right, the 'Results' tab shows the output of the query, which is a table with columns: BusinessEntityID, CreditCardID, and ModifiedDate. The data includes rows for various customer IDs like 293, 295, 297, etc., with their respective credit card IDs and modification dates.

	BusinessEntityID	CreditCardID	ModifiedDate
1	293	17038	2013-07-31 00:00:00
2	295	15369	2011-08-01 00:00:00
3	297	8010	2011-08-01 00:00:00
4	299	5316	2013-07-31 00:00:00
5	301	6653	2011-05-31 00:00:00
6	303	9010	2012-07-31 00:00:00
7	305	8937	2011-05-31 00:00:00
8	307	16124	2012-10-30 00:00:00
9	309	9478	2011-07-01 00:00:00

Assignment:

- 1.Create login with name dbuser with complex password
- 2) Assign Schema level permission for Sales schema in AdventureWorks db for select , insert, update , delete.
- 3) Restrict this dbuser to delete data of only one table [Sales].[SalesPerson] from sales Schema (Hint -You need to deny here on this table)
- 4) Grant select access to other schema few tables which are listed below.

**BCP :**

The bulk copy program utility (bcp) bulk copies data between an instance of Microsoft SQL Server and a data file in a user-specified format. The bcp utility can be used to import large numbers of new rows into SQL Server tables or to export data out of tables into data files. Except when used with the query out option, the utility requires no knowledge of Transact-SQL. To import data into a table, you must either use a format

file created for that table or understand the structure of the table and the types of data that are valid for its columns.

### **BCP out command:**

```
BCP [AdventureWorks2019].[Sales].[Customer] OUT Desktop\customer1.txt -S  
"VMT10126\TEST1" -U sa -P abcd@1234 -b 5000 -C RAW -n
```

#### BCP commands execution for sales.vendor table

```
BCP [AdventureWorks2019].[Sales].[Customer] OUT Desktop\customer1.txt -S  
"VMT10126\TEST1" -U sa -P abcd@1234 -b 5000 -C RAW -n
```

```
C:\Users\neha>BCP [AdventureWorks2019].[Sales].[Customer] OUT Desktop\customer1.txt -S "VMT10126\TEST1" -U sa -P abcd@1234 -b 5000 -C RAW -n  
Starting copy...  
1000 rows successfully bulk-copied to host-file. Total received: 1000  
1000 rows successfully bulk-copied to host-file. Total received: 2000  
1000 rows successfully bulk-copied to host-file. Total received: 3000  
1000 rows successfully bulk-copied to host-file. Total received: 4000  
1000 rows successfully bulk-copied to host-file. Total received: 5000  
1000 rows successfully bulk-copied to host-file. Total received: 6000  
1000 rows successfully bulk-copied to host-file. Total received: 7000  
1000 rows successfully bulk-copied to host-file. Total received: 8000  
1000 rows successfully bulk-copied to host-file. Total received: 9000  
1000 rows successfully bulk-copied to host-file. Total received: 10000  
1000 rows successfully bulk-copied to host-file. Total received: 11000  
1000 rows successfully bulk-copied to host-file. Total received: 12000  
1000 rows successfully bulk-copied to host-file. Total received: 13000  
1000 rows successfully bulk-copied to host-file. Total received: 14000  
1000 rows successfully bulk-copied to host-file. Total received: 15000  
1000 rows successfully bulk-copied to host-file. Total received: 16000  
1000 rows successfully bulk-copied to host-file. Total received: 17000  
1000 rows successfully bulk-copied to host-file. Total received: 18000  
1000 rows successfully bulk-copied to host-file. Total received: 19000
```

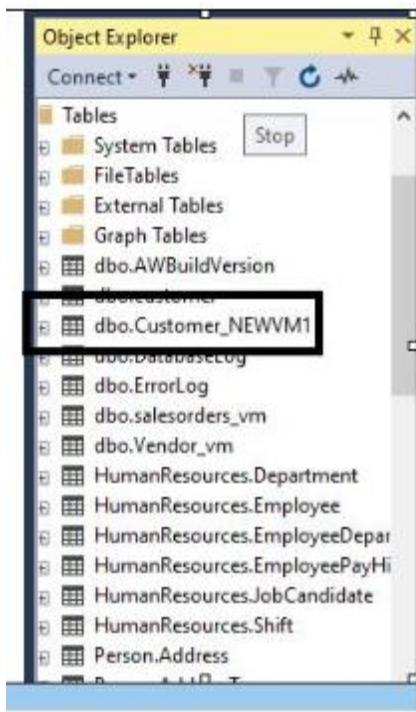
### **BCP In command:**

```
BCP [AdventureWorks2019].[dbo].[Customer_NEWM1] in Desktop\customer1.txt -S  
"VMT10126\TEST1" -U sa -P abcd@1234 -b 5000 -C RAW -n
```

```
BCP [AdventureWorks2019].[dbo].[Customer_NEWM1] in Desktop\customer1.txt -S  
"VMT10126\TEST1" -U sa -P abcd@1234 -b 5000 -C RAW -n
```

```
C:\Users\neha>BCP [AdventureWorks2019].[dbo].[Customer_NEWM1] in Desktop\customer1.txt -S "VMT10126\TEST1" -U sa -P abcd@1234 -b 5000 -C RAW -n  
Starting copy...  
5000 rows sent to SQL Server. Total sent: 5000  
5000 rows sent to SQL Server. Total sent: 10000  
5000 rows sent to SQL Server. Total sent: 15000
```

After executing the above two commands table is added to the ssms



## **Backups:**

SQL Backup helps us to improve the availability and service of the SQL database continuously to the users. If you are using the SQL database in the production environment for your clients then you need to make sure that your database is always working fine and is available 24 \* 7. For the availability of the database, you must often keep the backup of your client's database. In case if the database is corrupted or is crashed or lost then you should be able to restore the data in the database.

## **Recovery Methods:**

### 1. Full Recovery Model:

With the full recovery model, SQL Server preserves the transaction log until you back it up. This allows you to design a disaster recovery plan that includes a combination of full and differential database backups in conjunction with transaction log backups.

You have the most flexibility restoring databases using the full recovery model when a database failure happens. In addition to preserving data modifications stored in the

transaction log, the full recovery model allows you to restore a database to a specific point in time.

Advantage: no work is lost due to a lost or damaged data file. It can recover to an arbitrary point in time.

Disadvantage: if the log is damaged, changes since the most recent log backup must be redone.

## 2.Bulk logged recovery model:

The bulk-logged recovery model is a special-purpose model that works in a similar manner to the full recovery model. The only difference is in the way it handles bulk data modification operations. The bulk-logged model records these operations in the transaction log using a technique known as minimal logging. This saves significantly on processing time but prevents you from using the point-in-time restore option.

Advantage: permits high-performance bulk copy operations, minimal log space is used by bulk operations.

Disadvantage: if the log is damaged, or bulk operations occurred since the most recent log backup, changes since that last backup must be redone.

Full Recovery and Bulk-Logged Recovery models provide the greatest protection for data. These models rely on the transaction log to provide full recoverability and to prevent work loss in the broadest range of failure scenarios. The Bulk-Logged model provides higher performance and lower log space consumption for certain large-scale operations.

When you choose different MS SQL Server backup strategies, you will have specific limitations to recover SQL Server backups.

## 3.Simple Recovery model:

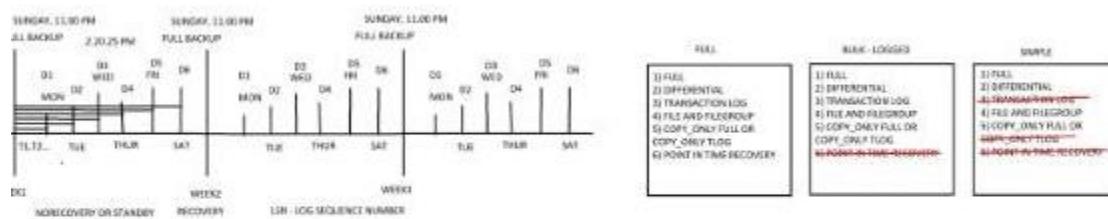
When you choose the simple recovery model, SQL Server maintains only a minimal amount of information in the transaction log. SQL Server truncates the transaction log each time the database reaches a transaction checkpoint, leaving no log entries for disaster recovery purposes.

In databases using the simple recovery model, you may restore full or differential backups only. It is not possible to restore such a database to a given point in time, you may only restore it to the exact time when a full or differential backup occurred. Therefore, you will automatically lose any data modifications made between the time of the most recent full/differential backup and the time of the failure.

Simple Recovery requires the least administration. It is easier to manage than the Full or Bulk-Logged models, but at the expense of higher data loss exposure if a data file is damaged. Simple Recovery is not an appropriate choice for production systems where loss of recent changes is unacceptable. When using Simple Recovery, the backup interval should be long enough to keep the backup overhead from affecting production work, yet short enough to prevent the loss of significant amounts of data.

**Advantage:** permits high-performance bulk copy operations. Reclaims log space to keep space requirements small.

**Disadvantage:** changes since the most recent database or differential backup must be redone.



Types of recovery:

- Full backup
- Transaction log
- Differential backup

### **Full backup:**

In this type of backup strategy, all the data of the database is backed up in the disk or memory location that we have specified in this query. As whole data from the database is being backed up the process of backup may take some time especially when there is huge data to be backed up. Also, the restoration process becomes time-consuming.

But once the backup is created using a full backup strategy, at the time of necessity the database can be restored to the state when it was being backed up. In SQL, we use the BACKUP DATABASE command to create a full backup of the database with which we also have to mention the name of the database that we wish to back up and the disk location where the backup needs to be created.

```
create table USA_FullBackup_1(srno int)
```

```
backup database USA
```

```
to disk = 'C:\sql\USA_FULL_2MARCH2018.bak'
```

```
with stats=10
```

```
RESTORE FILELISTONLY FROM disk = 'C:\sql\USA_FULL_2MARCH2018.bak' -- to check  
data files and log files present in
```

```
RESTORE FILELISTONLY FROM disk =  
'C:\sql\USA_FULL_2MARCH2018.bak' -- to check data files and log files present in  
backup
```

```
RESTORE verifyonly FROM disk = 'C:\sql\USA_FULL_2MARCH2018.bak' -- to check  
backup set is valid or not
```

```
RESTORE headeronly FROM disk = 'C:\sql\USA_FULL_2MARCH2018.bak' -- to check type  
or backup, servername, lsn of backups.
```

```
RESTORE database test2
```

```
FROM disk = 'C:\sql\USA_FULL_2MARCH2018.bak'
```

```
with stats=10,replace,norecovery,
```

```
move 'USA' to 'C:\sql\USA_New.mdf',
```

```
move 'USA_log' to 'C:\sql\USA_New_log.ldf' backup
```

### **Differential backup:**

Backing up the whole database every time can be very time consuming and unnecessary the old data is hardly been modified. In such cases, we can only backup the data that has recently been added and modified in the database from the time when the last backup was created. This saves a sufficient amount of resources and time that must have consumed and required for backing up the old that has already been backup in the past. While performing differential backup only a small amount of data is being

backup up between the time interval of last backup and current one provided if the number of transactions that are made in that time duration as considerably small.

In case if a huge amount of data is being generated each day then differential backups can also become very time-consuming. Also, note that the restoration process doesn't get affected and still consumes the same time as consumed while restoring full backup as in both the case whole database is being restored. The command used to perform the differential backup in SQL is the same BACKUP DATABASE along with an extra clause of the specification that is DIFFERENTIAL along with which the name of database and disk location needs to be specified.

```
create table USA_DIFF1_2MARCH2018_2(srno int)
backup database USA
to disk = 'C:\sql\test2_DIFF1_2MARCH2018_2.bak'
with stats=10,DIFFERENTIAL

RESTORE FILELISTONLY FROM disk = 'C:\sql\USA_DIFF1_2MARCH2018_1.bak' -- to
check data files and log files present in backup
RESTORE verifyonly FROM disk = 'C:\sql\USA_DIFF1_2MARCH2018_1.bak' -- to check
backup set is valid or not
RESTORE headeronly FROM disk = 'C:\sql\USA_DIFF1_2MARCH2018_1.bak' -- to check
type or backup, servername, lsn of backups.

RESTORE database test2
FROM disk = 'C:\sql\USA_DIFF1_2MARCH2018_1.bak'
with stats=10,replace,norecovery
```

### **Transactional backup:**

When bulk-logged or full recovery models are being used in the SQL database then we can use the transactional log backups. In this type of backup, we can take the backup of all the logs of the transactions that have been made on the database so that the data can be recovered at any specific point of time where we want as all the history of the modifications on the database is stored in the logs of transactional log backup.

This type of backup consists of all the logs that have not been backed up in the previous transactional log backup. The nature of the backup of the transaction log is incremental. When an accident occurs that results in the data loss and you want to restore the database to the specific point of time that is just before the time of the accident then

you will have to restore latest differential backup data and all the transaction logs whose modifications you wish to desire to be restored.

```
create table USA_TRNLOG1_2MARCH2018_1(srno int)
backup LOG USA
to disk = 'C:\sql\USA_TRNLOG1_2MARCH2018_1.bak'
with stats=10
```

```
RESTORE FILELISTONLY FROM disk = 'C:\sql\USA_TRNLOG1_2MARCH2018_1.bak' -- to
check data files and log files present in backup
RESTORE verifyonly FROM disk = 'C:\sql\USA_TRNLOG1_2MARCH2018_1.bak' -- to
check backup set is valid or not
RESTORE headeronly FROM disk = 'C:\sql\USA_TRNLOG1_2MARCH2018_1.bak' -- to
check type or backup, servername, lsn of backups.
```

```
RESTORE LOG test2
FROM disk = 'C:\sql\USA_TRNLOG1_2MARCH2018_1.bak'
with stats=10,replace
```

### **Assignment:**

Q:Backup and restoring from database1 to database2:

Restore of database from test1 to testrestore

TEST

SQL Server Object Explorer

MT10213\TEST1 (SQL Server 15.0.2080 - sa)

- Databases
  - System Databases
  - Database Snapshots
  - AdventureWorks2017
  - AdventureWorks2019
  - microsoft
  - TEST
  - Testrestore**
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - External Tables
    - Graph Tables
    - dbo.Test\_diffbackup\_1
    - dbo.Test\_fullbackup\_1
    - dbo.Test\_transbackup\_1
  - Views
  - External Resources
  - Synonyms
  - Programmability
  - Service Broker
  - Storage
  - Security
- TSQL2012
- Security
- Server Objects
- Replication
- PolyBase

SQLQuery11.sql - V...\EST1.TEST (sa (63)) \* SQLQuery10.sql - V...\EST1.TEST (sa (63)) \* SQLQuery6.sql - VM...\EST1.TEST (sa (62))

```

CREATE TABLE Test_transbackup_1(srno int)
BACKUP LOG Test
TO DISK = 'C:\sql\backups\transback_1.bak'
WITH STATS=10

RESTORE FILELISTONLY FROM disk = 'C:\sql\backups\transback_1.bak' -- to check data files and log files present
RESTORE VERIFYONLY FROM disk = 'C:\sql\backups\transback_1.bak' -- to check backup set is valid or not
RESTORE HEADERONLY FROM disk = 'C:\sql\backups\transback_1.bak'

RESTORE LOG Testrestore
FROM disk = 'C:\sql\backups\transback_1.bak'
WITH STATS=10, REPLACE
  
```

Messages

90 % 4

19 percent processed.  
28 percent processed.  
58 percent processed.  
77 percent processed.  
96 percent processed.  
100 percent processed.  
Processed 0 pages for database 'Testrestore', file 'TEST' on file 1.  
Processed 42 pages for database 'Testrestore', file 'TEST\_log' on file 1.  
RESTORE LOG successfully processed 42 pages in 0.007 seconds (46.107 MB/sec).

Completion time: 2021-01-28T15:18:21.6804821+05:30

90 % 4

Query executed successfully.

VMT10213\TEST1 (15.0 RTM)

Ln 14 Col 22 Ch 22 INS

## From testrestore to test4

MT10213\TEST1 (SQL Server 15.0.2080 - sa)

SQL Server Object Explorer

MT10213\TEST1 (SQL Server 15.0.2080 - sa)

- Databases
  - System Databases
  - Database Snapshots
  - AdventureWorks2017
  - AdventureWorks2019
  - microsoft
  - TEST
  - Test4**
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - External Tables
    - Graph Tables
    - dbo.Test\_diffbackup\_1
    - dbo.Test\_fullbackup\_1
    - dbo.Test\_transbackup\_1
  - Views
  - External Resources
  - Synonyms
  - Programmability
  - Service Broker
  - Storage
  - Security
- Testrestore
- TSQL2012
- Security
- Server Objects
- Replication

backup database Testrestore
 to disk = 'C:\sql\backups\fullback\_1.bak'
 with stats=10

 RESTORE FILELISTONLY FROM disk = 'C:\sql\backups\fullback\_1.bak' -- to check data files and log files present
 RESTORE VERIFYONLY FROM disk = 'C:\sql\backups\fullback\_1.bak' -- to check backup set is valid or not
 RESTORE HEADERONLY FROM disk = 'C:\sql\backups\fullback\_1.bak'

 RESTORE database Test4
 FROM disk = 'C:\sql\backups\fullback\_1.bak'
 WITH STATS=10, REPLACE, NORECOVERY,
 MOVE 'TEST' TO 'C:\sql\backups\test4.mdf',
 MOVE 'TEST\_log' TO 'C:\sql\backups\test4.ldk'

 RESTORE DATABASE Test4 WITH RECOVERY

Messages

90 % 4

11 percent processed.  
21 percent processed.  
30 percent processed.  
40 percent processed.  
51 percent processed.  
61 percent processed.  
70 percent processed.  
80 percent processed.  
91 percent processed.  
100 percent processed.  
Processed 352 pages for database 'Test4', file 'TEST' on file 1.  
Processed 1 pages for database 'Test4', file 'TEST\_log' on file 1.  
RESTORE DATABASE successfully processed 353 pages in 0.012 seconds (229.258 MB/sec).

Completion time: 2021-01-28T16:28:17.0944838+05:30

90 % 4

Query executed successfully.

VMT10213\TEST1 (15.0 RTM)

Ln 14 Col 22 Ch 22 INS

## Backup restoring(copy\_only):

COPY\_ONLY backup is a special type of SQL Server Backup which is independent of the sequence of conventional SQL Server backups. Normally whenever you take a backup it will affect how later backups are restored. However, when you use COPY\_ONLY backup feature of SQL Server one can take a backup of the database without affecting the overall backup and restore procedures for the database.

One can use COPY\_ONLY backup if there is a need to refresh the Development/QA environment with the latest backup of your Production database. In such a scenario you could run a FULL backup of the database WITH COPY\_ONLY clause to take a backup without affecting the backup chain.

The transaction log is never truncated after a COPY\_ONLY backup. A COPY\_ONLY log backup preserves the existing log archive point and, therefore, does not affect the sequencing of regular log backups.

Types of COPY\_ONLY backups are:-

- Using COPY\_ONLY feature one can create FULL backup for any user databases irrespective of the recovery model used by the database. i.e., it supports FULL, BULK-LOGGED and SIMPLE recovery model. However, restoring a COPY\_ONLY backup is same as restoring a normal FULL Backup.
- Using COPY\_ONLY feature one can create DIFFERENTIAL backup for any user databases irrespective of the recovery model used by the database. i.e., it supports FULL, BULK-LOGGED and SIMPLE recovery model.
- Using COPY\_ONLY feature one can create TRANSACTIONAL LOG backup for any user database which is in FULL or BULK-LOGGED recovery model.

```
Create table VMTest_full_copy_only_backup1 (srno int)
```

```
backup database VMTest1  
to disk = 'C:\sql\Backups\VMTest_fullbackup1.bak'  
with stats=10, copy_only
```

### **Indexes:**

1. It is a database object which is used to speed up the querying process by providing query access to rows in the data table.
2. When we create an index on any column SQL Server internally maintains a separate table called index table so that when any user trying to retrieve the data from the

existing table depends on index table SQL Server directly go to the table and retrieve required data very quickly.

3.In a table, we can use a maximum of 250 indexes. The index type refers to the way the index is stored internally by SQL Server.

The SQL Server uses indexes of a table provided the select or update or delete statement contained the "WHERE" clause and moreover the where condition column must be an indexed column. If the select statement contains an "ORDER BY" clause then also the indexes can be used.

Note: When SQL Server is searching for information under the database first it verifies the best execution plan for retrieving the data and uses that plan which can be either a full-page scan and index scan.

The syntax for creating Index in SQL Server:

```
CREATE [UNIQUE] [CLUSTERED/ NON-CLUSTERED] INDEX <INDEX NAME> ON <TABLE NAME> (<COLUMN LIST>)
```

To see the index: sp\_helpindex Employee

To drop an index: Drop index Employee.IX\_Employee\_Id

- Whenever the database engine wants to retrieve the data from a database table it will adopt two different mechanisms for searching the data

1.Table scan

2.Index Scan/Seek

- Table Scan in SQL Server:

In Table Scan, the SQL Server Search Engine will search for the required information sequentially one by one from the start to the last record of the table. If the table having more number of rows, then it will take more time for searching the required data, so it is a time-consuming process.

```

select * from VMTest1

select
    [PersonID]
    ,[StoreID]
  from VMTest1
  where [CustomerID]=100

```

Table Scan  
Scan rows from a table.

Physical Operation	Table Scan
Logical Operation	Table Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated I/O Cost	0.0083887
Estimated Operator Cost	0.0095672 (100%)
Estimated CPU Cost	0.0011785
Estimated Subtree Cost	0.0095672
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	1000
Estimated Row Size	19 B
Ordered	False
Node ID	0

Predicate  
[AdventureWorks2016].[dbo].[VMTest1].[CustomerID]  
=CONVERT\_IMPLICIT(int,[@1],0)

Object  
[AdventureWorks2016].[dbo].[VMTest1]

Output List  
[AdventureWorks2016].[dbo].[VMTest1].PersonID,  
[AdventureWorks2016].[dbo].[VMTest1].StoreID

Query executed successfully.

### Index Scan/Seek in SQL Server:

In Index Scan, the SQL Server Search Engine uses B-Tree structure to search the required data which can reduce the disk I/O operation saving the time, as a result, we will get better performance.

SQLQuery6.sql - VM...orks2016 (sa (56))\*

SQLQuery5.sql - VM...orks2016 (sa (53))\*

SQLQuery4.sql - VM...orks2

```

select * from VMTest123

select
    [PersonID]
    ,[StoreID]
    from VMTest123
    where [CustomerID]=100

alter table VMTest123
add constraint pk_customerid
    primary key clustered ([CustomerID])

alter table VMTest123
drop constraint pk_customerid

create clustered index ix_clu_customerid_VMTest123
    on VMTest123 ([CustomerID])
drop index nk_customerid_vmt

```

00 %

Messages Execution plan

Query 1: Query cost (relative)

```

select [PersonID] , [StoreID]

```

SELECT Cost: 0 %

Clustered Index [VMTest123].[ix\_clu\_customerid\_VMTest123] Cost:

Clustered Index Seek (Clustered)  
Scanning a particular range of rows from a clustered index.

Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	0.0032831 (100%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0032831
Estimated CPU Cost	0.0001581
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	1
Estimated Row Size	15 B
Ordered	True
Node ID	0

**Object**  
[AdventureWorks2016].[dbo].[VMTest123].[ix\_clu\_customerid\_VMTest123]

**Output List**  
[AdventureWorks2016].[dbo].[VMTest123].PersonID,  
[AdventureWorks2016].[dbo].[VMTest123].StoreID

**Seek Predicates**  
Seek Keys[1]: Prefix: [AdventureWorks2016].[dbo].[VMTest123].CustomerID = Scalar Operator(CONVERT\_IMPLICIT(int,[@1],0))

Query executed successfully.

## Types of indexes

Indexes are divided into two types such as

1.Clustered index

2.Non- clustered index

### Clustered index:

The Clustered Index in SQL Server defines the order in which the data is physically stored in a table. That is the leaf node store the actual data. As the leaf nodes store the actual data a table can have only one clustered index.

The Clustered Index by default created when you created the primary key constraint for that table. That means the primary key column creates a clustered index by default.

When a table has a clustered index then the table is called a clustered table. If a table has no clustered index its data rows are stored in an unordered structure.

```
alter table VMTest1234
```

```
add constraint pk_BusinessEntityID_VMTest1234 primary key ([BusinessEntityID])
```

```
create      clustered      index      ix_clu_BusinessEntityID_VMTest1234      on  
VMTest1234(BusinessEntityID)
```

### **Non- Clustered index:**

In the Non-Clustered Index, the arrangement of data in the index page or table will be different from the arrangement of data in the actual table. The data is stored in one place and the index is stored in another place. Moreover, the index will have pointers to the storage location of the actual data.

```
create      nonclustered      index      ix_nc_FirstName  
on          VMTest1234      ([FirstName])  
include ([PersonType],[NameStyle])
```

```

drop index ix_clu_customerid_VMTest123 on VMTest123

select [PersonID]
      ,[StoreID]
  from VMtest123
 where [TerritoryID]= 6

create nonclustered index ix_nc
on VMTest123 ([TerritoryID])
include ([PersonID],[StoreID])

sp_help VMTest123

```

Index Seek (NonClustered)  
Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	0.0034327 (100%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0034327
Estimated CPU Cost	0.0003077
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	137
Estimated Number of Rows to be Read	137
Estimated Row Size	15 B
Ordered	True
Node ID	0

Object  
[AdventureWorks2016].[dbo].[VMTest123].[ix\_nc\_TerritoryID]  
Output List  
[AdventureWorks2016].[dbo].[VMTest123].PersonID,  
[AdventureWorks2016].[dbo].[VMTest123].StoreID  
Seek Predicates  
Seek Keys[1]: Prefix: [AdventureWorks2016].[dbo].[VMTest123].TerritoryID = Scalar Operator(CONVERT\_IMPLICIT(int,[@1],0))

Query executed successfully.

## Assignment:

### Backups and restore:

Q:create one db name with VMTest and create respective tables amd 3 sets of backups.restore backup set full backup(1) and restore diff backup(5)

1)full\_backup1

create table Full\_backup1(srno int)

backup database VMTEST1

```
to disk = 'C:\sql\backup1\full_backup12.bak'  
with stats=10
```

RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\full\_backup12.bak' -- to check data files and log files present in backup

RESTORE verifyonly FROM disk = 'C:\sql\backup1\full\_backup12.bak' -- to check backup set is valid or not

RESTORE headeronly FROM disk = 'C:\sql\backup1\full\_backup12.bak' -- to check type or backup, servername, lsn of backups.

```
RESTORE database VM_test1
```

```
FROM disk = 'C:\sql\backup1\full_backup12.bak'
```

```
with stats=10,replace,norecovery,
```

```
move 'VMTEST1' to 'C:\sql\backup1\full_backup12_restore123.mdf',
```

```
move 'VMTEST1_log' to 'C:\sql\backup1\full_backup12log_restore123.ldk'
```

```
2)diff_backup1
```

```
create table diff_backup1(srno int)
```

```
backup database VMTEST1
```

```
to disk = 'C:\sql\backup1\diff_backup12.bak'
```

```
with stats=10,DIFFERENTIAL
```

RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\diff\_backup12.bak' -- to check data files and log files present in backup

RESTORE verifyonly FROM disk = 'C:\sql\backup1\diff\_backup12.bak' -- to check backup set is valid or not

RESTORE headeronly FROM disk = 'C:\sql\backup1\diff\_backup12.bak'

```
RESTORE database VM_test1
```

```
FROM disk = 'C:\sql\backup1\diff_backup12.bak'
```

```
with stats=10,replace,norecovery
```

```
3)tlog_backup1
```

```
create table Tlog_backup1(srno int)
```

```
backup LOG VMTEST1
```

```
to disk = 'C:\sql\backup1\Tlog_backup12.bak'
```

```
with stats=10
```

```
RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\Tlog_backup12.bak' -- to check  
data files and log files present in backup
```

```
RESTORE verifyonly FROM disk = 'C:\sql\backup1\Tlog_backup12.bak' -- to check  
backup set is valid or not
```

```
RESTORE headeronly FROM disk = 'C:\sql\backup1\Tlog_backup12.bak'
```

```
RESTORE log VM_test1
```

```
FROM disk = 'C:\sql\backup1\Tlog_backup12.bak'
```

```
with stats=10,replace
```

```
4)full_copy_only_backup2
```

```
create table Full_copy_only_backup2(srno int)
```

```
backup database VMTEST1
```

```
to disk = 'C:\sql\backup1\full_backup22.bak'
```

```
with stats=10,copy_only
```

```
RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\full_backup22.bak' -- to check  
data files and log files present in backup
```

```
RESTORE verifyonly FROM disk = 'C:\sql\backup1\full_backup22.bak' -- to check backup  
set is valid or not
```

RESTORE headeronly FROM disk = 'C:\sql\backup1\full\_backup22.bak' -- to check type or backup, servername, lsn of backups.

```
RESTORE database VM_test1
FROM disk = 'C:\sql\backup1\full_backup22.bak'
with stats=10,replace,norecovery,
move 'VMTEST1' to 'C:\sql\backup1\vm_test1.mdf',
move 'VMTEST1_log' to 'C:\sql\backup1\vm_test1_log.ldk'
5)diff_backup2
create table diff_backup2(srno int)
backup database VMTEST1
to disk = 'C:\sql\backup1\diff_backup22.bak'
with stats=10,DiFFERENTIAL
```

RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\diff\_backup22.bak' -- to check data files and log files present in backup

RESTORE verifyonly FROM disk = 'C:\sql\backup1\diff\_backup22.bak' -- to check backup set is valid or not

RESTORE headeronly FROM disk = 'C:\sql\backup1\diff\_backup22.bak'

```
RESTORE database VM_test1
FROM disk = 'C:\sql\backup1\diff_backup22.bak'
with stats=10,replace,norecovery
6)tlog_backup2
create table Tlog_backup2(srno int)
backup LOG VMTEST1
to disk = 'C:\sql\backup1\Tlog_backup22.bak'
```

with stats=10

RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\Tlog\_backup22.bak' -- to check data files and log files present in backup

RESTORE verifyonly FROM disk = 'C:\sql\backup1\Tlog\_backup22.bak' -- to check backup set is valid or not

RESTORE headeronly FROM disk = 'C:\sql\backup1\Tlog\_backup22.bak'

RESTORE database VM\_test1

FROM disk = 'C:\sql\backup1\Tlog\_backup22.bak'

with stats=10,replace

7)full\_backup3

create table Full\_backup3(srno int)

backup database VMTEST1

to disk = 'C:\sql\backup1\full\_backup3.bak'

with stats=10

RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\full\_backup3.bak' -- to check data files and log files present in backup

RESTORE verifyonly FROM disk = 'C:\sql\backup1\full\_backup3.bak' -- to check backup set is valid or not

RESTORE headeronly FROM disk = 'C:\sql\backup1\full\_backup3.bak' -- to check type or backup,servername, lsn of backups.

RESTORE database VM\_test1

FROM disk = 'C:\sql\backup1\full\_backup3.bak'

with stats=10,replace,norecovery,

```
move 'VMTEST1' to 'C:\sql\backup1\full_backup12_restore3.mdf',
move 'VMTEST1_log' to 'C:\sql\backup1\full_backup12log_restore3.ldk'
8)diff_backup3
create table diff_backup3(srno int)
backup database VMTEST1
to disk = 'C:\sql\backup1\diff_backup3.bak'
with stats=10,DIFFERENTIAL
RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\diff_backup3.bak' -- to check data
files and log files present in backup
RESTORE verifyonly FROM disk = 'C:\sql\backup1\diff_backup3.bak' -- to check backup
set is valid or not
RESTORE headeronly FROM disk = 'C:\sql\backup1\diff_backup3.bak'
```

```
RESTORE database VM_test1
FROM disk = 'C:\sql\backup1\diff_backup3.bak'
with stats=10,replace,norecovery
9)tlog_backup3
create table Tlog_backup3(srno int)
backup LOG VMTEST1
to disk = 'C:\sql\backup1\Tlog_backup3.bak'
with stats=10
```

```
RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\Tlog_backup3.bak' -- to check
data files and log files present in backup
RESTORE verifyonly FROM disk = 'C:\sql\backup1\Tlog_backup3.bak' -- to check backup
set is valid or not
RESTORE headeronly FROM disk = 'C:\sql\backup1\Tlog_backup3.bak'
```

RESTORE log VM\_test1

FROM disk = 'C:\sql\backup1\Tlog\_backup3.bak'

with stats=10,replace

```
RESTORE log VMTEST1
FROM disk = 'C:\sql\backup1\Tlog_backup3.bak'
with stats=10,replace
```

```
RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\Tlog_backup22.bak' -- to check data files and log files present in backup
RESTORE verifyonly FROM disk = 'C:\sql\backup1\Tlog_backup22.bak' -- to check backup set is valid or not
RESTORE headeronly FROM disk = 'C:\sql\backup1\Tlog_backup22.bak'
```

```
RESTORE database VM_test1
FROM disk = 'C:\sql\backup1\Tlog_backup22.bak'
with stats=10,replace
```

100 percent processed.  
Processed 0 pages for database 'VM\_test1', file 'VMTEST1' on file 1.  
Processed 4 pages for database 'VM\_test1', file 'VMTEST1\_log' on file 1.  
RESTORE LOG successfully processed 4 pages in 0.005 seconds (5.664 MB/sec).

Completion time: 2021-01-29T19:46:06.7082688+05:00

Query executed successfully.

```
backup LOG VMTEST1
to disk = 'C:\sql\backup1\Tlog_backup2.bek'
with stats=10
```

```
RESTORE FILELISTONLY FROM disk = 'C:\sql\backup1\Tlog_backup22.bak' -- to check data files and log files present in backup
RESTORE verifyonly FROM disk = 'C:\sql\backup1\Tlog_backup22.bak' -- to check backup set is valid or not
RESTORE headeronly FROM disk = 'C:\sql\backup1\Tlog_backup22.bak'
```

```
RESTORE database VM_test1
FROM disk = 'C:\sql\backup1\Tlog_backup22.bak'
with stats=10,replace
```

100 percent processed.  
Processed 0 pages for database 'VM\_test1', file 'VMTEST1' on file 1.  
Processed 4 pages for database 'VM\_test1', file 'VMTEST1\_log' on file 1.  
RESTORE LOG successfully processed 4 pages in 0.005 seconds (5.664 MB/sec).

Completion time: 2021-01-29T19:46:06.7082688+05:00

## Indexes

1)for sales.customer table

```
SELECT [CustomerID]
```

```
,[PersonID]
```

```
,[StoreID]
```

```
,[TerritoryID]
```

```
,[AccountNumber]
```

```
,[rowguid]
```

```
,[ModifiedDate]
```

```
into vmtest1
```

```
FROM [AdventureWorks2019].[Sales].[Customer]
```

```
select * from dbo.vmtest1
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database 'AdventureWorks2019' selected. The right pane shows the results of a query:

```
select * from dbo.vmtest1
```

The results grid shows the following data:

	StoreID	int	no	4	10	U	yes	(n/a)	(n/a)	NULL
3	TerritoryID	int	no	4	10	0	yes	(n/a)	(n/a)	NULL
4	AccountNu...	var_	no	10			no	no	no	SQL_Latin1_General_CI_AS
5	rowguid	uni_	no	16			no	(n/a)	(n/a)	NULL
6	ModifiedDate	dat_	no	8			no	(n/a)	(n/a)	NULL
7										

Below the results grid, there are tabs for 'Identity', 'Seed', 'Increment', and 'Not For Replication'. The 'Identity' tab is selected, showing 'CustomerID' as the identity column with a seed of 1 and an increment of 0.

```
select [PersonID]
```

```
,[StoreID]
```

from vmtest1

where [CustomerID]= 100

The screenshot shows the SQL Server Management Studio interface with two panes. The left pane displays the database structure of 'T1 (SQL Server 15.0.2080 - sa)' with various system and user tables listed. The right pane shows the results of a query execution plan for the following SQL statement:

```
select * from dbo.vmtest1
select [PersonID]
,[StoreID]
from vmtest1
where [CustomerID]= 100
```

The execution plan details a "Table Scan" operation. Key statistics shown include:

Physical Operation	Table Scan
Logical Operation	Table Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated I/O Cost	0.117278
Estimated Operator Cost	0.139158 (100%)
Estimated CPU Cost	0.0218805
Estimated Subtree Cost	0.139158
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	1
Estimated Number of Rows to be Read	19820
Estimated Row Size	19 B
Ordered	False
Node ID	0

The predicate for the scan is: [AdventureWorks2019].[dbo].[vmtest1].[CustomerID] = CONVERT\_IMPLICIT(int,[@1],0). The output list includes [AdventureWorks2019].[dbo].[vmtest1].

At the bottom of the right pane, a message states: "Query executed successfully".

alter table vmtest1

add constraint pk\_customerid\_vmtest1 primary key ([CustomerID])

```

--TEST1 (SQL Server 15.0.2080 - sa)
--drop table vmtest1
--create table vmtest1([CustomerID] int,[PersonID] int,[StoreID] int)
--insert into vmtest1 values(100,1,1)
--alter table vmtest1 add constraint pk_customerid_vmt1 primary key ([CustomerID])

```

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'TEST1' is selected. In the main query window, a script is run to drop the table if it exists, create it with three columns, insert a single row with CustomerID 100, and then add a primary key constraint named 'pk\_customerid\_vmt1' on the CustomerID column. Below the script, the 'Results' tab displays the execution plan for the 'ALTER TABLE' statement. The plan shows four operations: a 'Compute Scalar' for the constraint definition, a 'Sort' operation, a 'Table Scan' for the primary key index, and an 'Insert' operation for the data row. The execution plan table includes columns for StmtText, StmtId, NodeId, Parent, PhysicalOp, LogicalOp, Argument, DefinedValues, EstimateRows, EstimateCPU, and AvgRowSize.

alter table vmtest1

drop constraint pk\_customerid\_vmt1

create clustered index ix\_clu\_customerid\_vmt1 on vmtest1(CustomerID)

```

--TEST1 (SQL Server 15.0.2080 - sa)
--drop table vmtest1
--create table vmtest1([CustomerID] int,[PersonID] int,[StoreID] int)
--insert into vmtest1 values(100,1,1)
--sp_help vmtest1
--alter table vmtest1 add constraint pk_customerid_vmt1 primary key ([CustomerID])
--alter table vmtest1 drop constraint pk_customerid_vmt1
--create clustered index ix_clu_customerid_vmt1 on vmtest1(CustomerID)

```

The screenshot shows the SQL Server Management Studio interface. In the Object Explorer, the database 'TEST1' is selected. In the main query window, a script is run to drop the table if it exists, create it with three columns, insert a single row with CustomerID 100, and then drop the primary key constraint 'pk\_customerid\_vmt1'. Finally, a clustered index 'ix\_clu\_customerid\_vmt1' is created on the CustomerID column. Below the script, the 'Results' tab displays the execution plan for the 'CREATE CLUSTERED INDEX' statement. The plan shows two operations: a 'No RowGuidCol column defined' operation and a 'Data\_located\_on\_filegroup' operation for the index 'ix\_clu\_customerid\_vmt1' located on the filegroup 'HR1'. The execution plan table includes columns for StmtText, StmtId, NodeId, Parent, PhysicalOp, LogicalOp, Argument, DefinedValues, EstimateRows, EstimateCPU, and AvgRowSize.

create nonclustered index ix\_nc\_TerritoryID

on vmtest1 ([TerritoryID])

include ([PersonID],[StoreID])

```

SQLQuery1.sql - VM_mworks2019 (sa (bo)) [ ] X SQLQuery1.sql - VM_mworks2019 (sa (bo))
drop index ix_clu_customerid_vmtest1 on vmtest1

select [PersonID]
      ,[StoreID]
  from vmtest1
 where [TerritoryID]= 6

create nonclustered index ix_nc_TerritoryID
on vmtest1 ([TerritoryID])
include ([PersonID],[StoreID])

```

The screenshot shows a SQL Server Management Studio interface. On the left, the Object Explorer displays a list of databases, tables, and other objects. In the center, a query window contains T-SQL code for dropping an index and creating a new one. Below the query window, an execution plan window is open, showing the results of the query's execution.

	rowguid	uni...	no	16	no	(n/a)	(n/a)	NULL
6	rowguid	uni...	no	16	no	(n/a)	(n/a)	NULL
7	ModifiedDate	dat...	no	8	no	(n/a)	(n/a)	NULL

	Identity	Seed	Increment	Not For Replication
1	CustomerID	1	1	0

	RowGuidCol
1	No rowguidcol column defined

	Data_located_on_filegroup
1	HR1

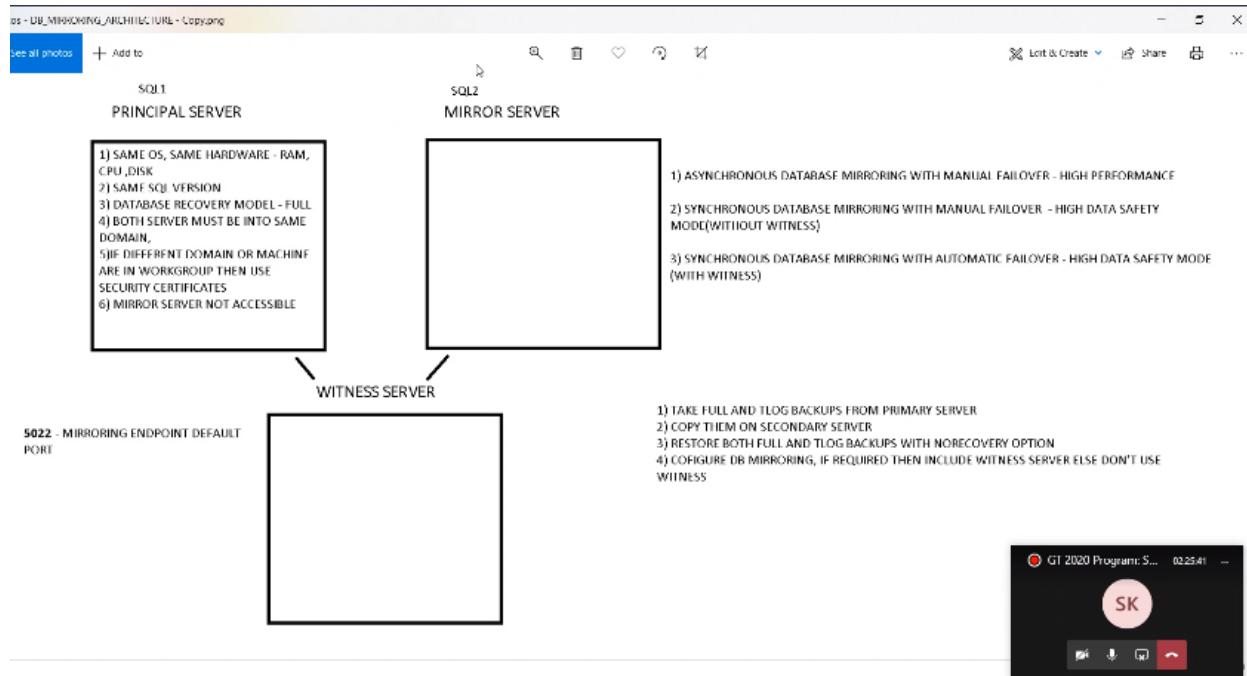
	index_name	index_description	index_keys
1	ix_nc_TerritoryID	nonclustered located on HR1	TerritoryID

## High Availability:

- 1.database mirroring
- 2.Log shipping
- 3.Replication
- 4.Clustering
- 5.Always on

## Database Mirroring :

- 1.primary
- 2.secondary



### Prerequisites:

- Principal and mirror server should have same configured like same ram ,same server and should be on same domain etc..
- Witness server used to monitor principal and mirror server
- In this both mirror and witness are sql servers.
- If principal server goes down the witness server identifies and saves replica to mirror server
- Asynchronous database mirroring with manual failover.it wont check whether data files sent. received or not.
- In Synchronous database mirroring it checks for files sent and after their reception it send further files.
- Synchronous database mirroring with manual failover
- synchronous database mirroring with automatic failover
- Database mirroring accepts only full recovery model.in that it accepts full and transactional backups and it should be sent over to multiple backups.
- In mirroring primary key is not required

### SQL server mirroring:

SQL Server database mirroring is a disaster recovery and high availability technique that involves two SQL Server instances on the same or different machines. One SQL Server instance acts as a primary instance called the **principal**, while the other is a mirrored

instance called the **mirror**. In special cases, there can be a third SQL Server instance that acts as a witness

### **Installation of VMWare:**

#### **Steps to Install VMware**

Below are the detailed steps for installing VMware Workstation.

**Step 1.** To download and install the VMware product visit the official website of VMware.

<https://www.vmware.com/in.html>

Workstation player is for running a second OS on Windows or Linux PC with free for personal use.

We have chosen VMware Workstation Pro for installation demo purpose. VMware Workstation is available with the latest version as 15.

**Step 2.** Click on Free Product Trials & Demo >> Workstation Pro. You will be redirected to the download page. (Similarly, you can select any product which you want to install.)

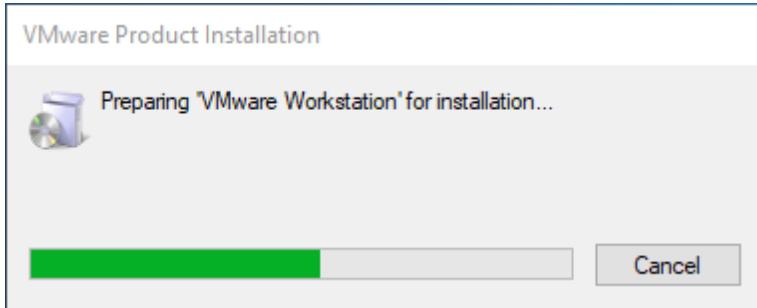
Try VMware Workstation Pro



Click on Download Now according to your Operating System. We have chosen Workstation 15 Pro for Windows.

While downloading make sure you have proper internet connection as the file may have a large size.

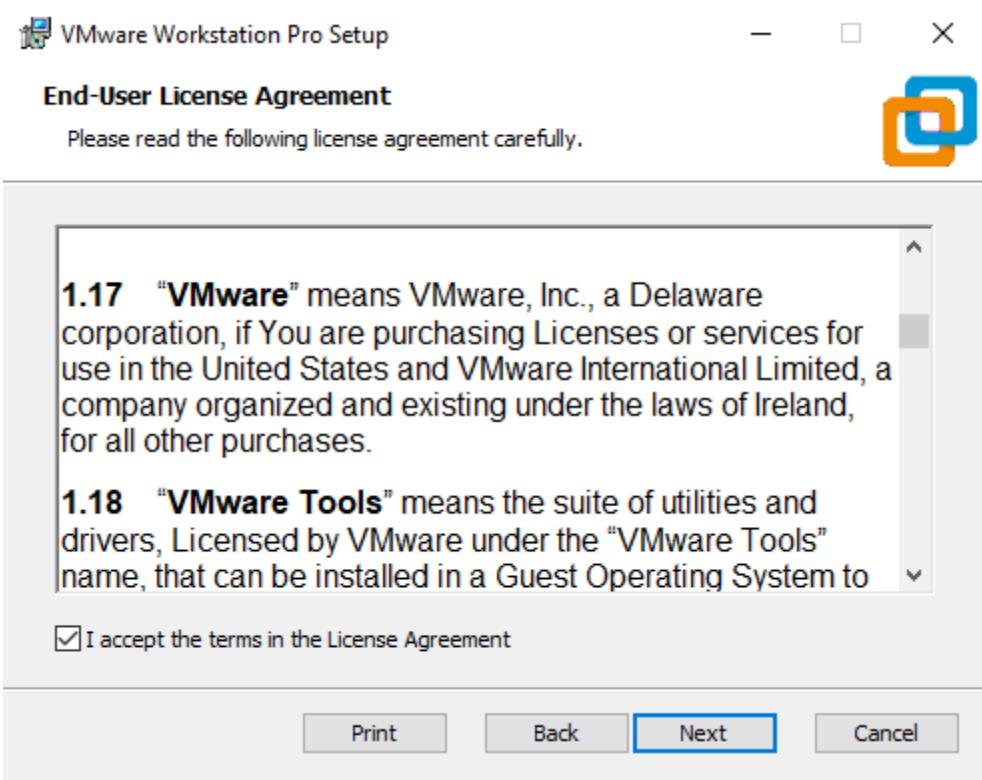
**Step 3.** Once the download is complete, run the .exe to install VMware Workstation. Popup will appear.



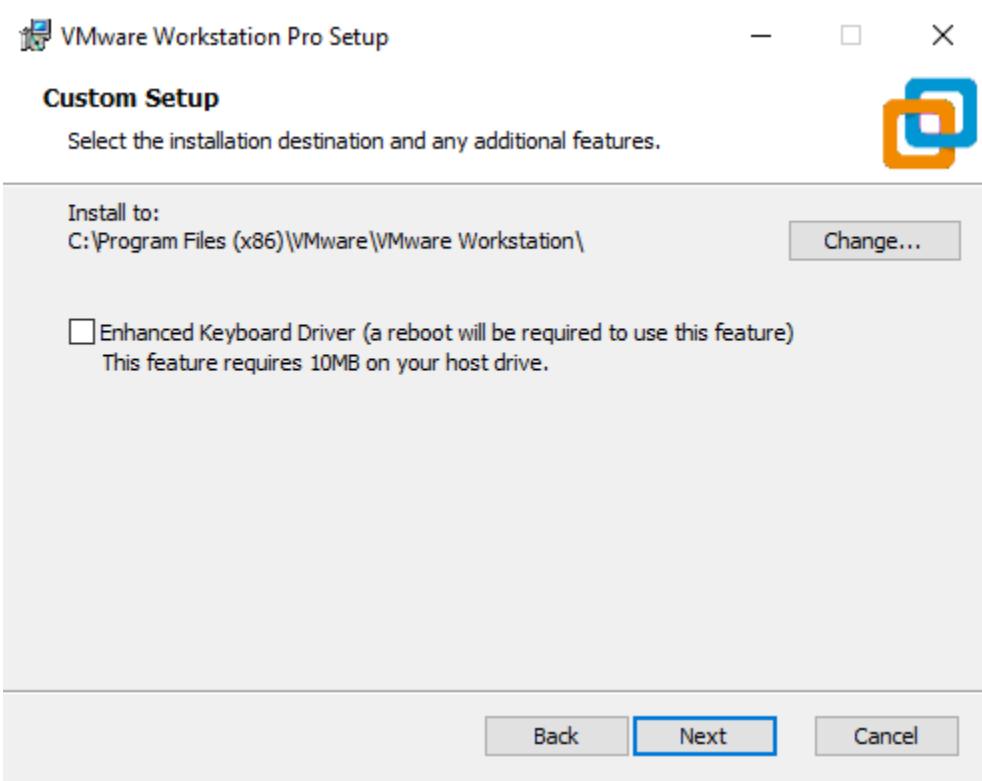
**Step 4.** Once Initialization gets completes, Click on Next.



**Step 5.** Accept the terms and click Next

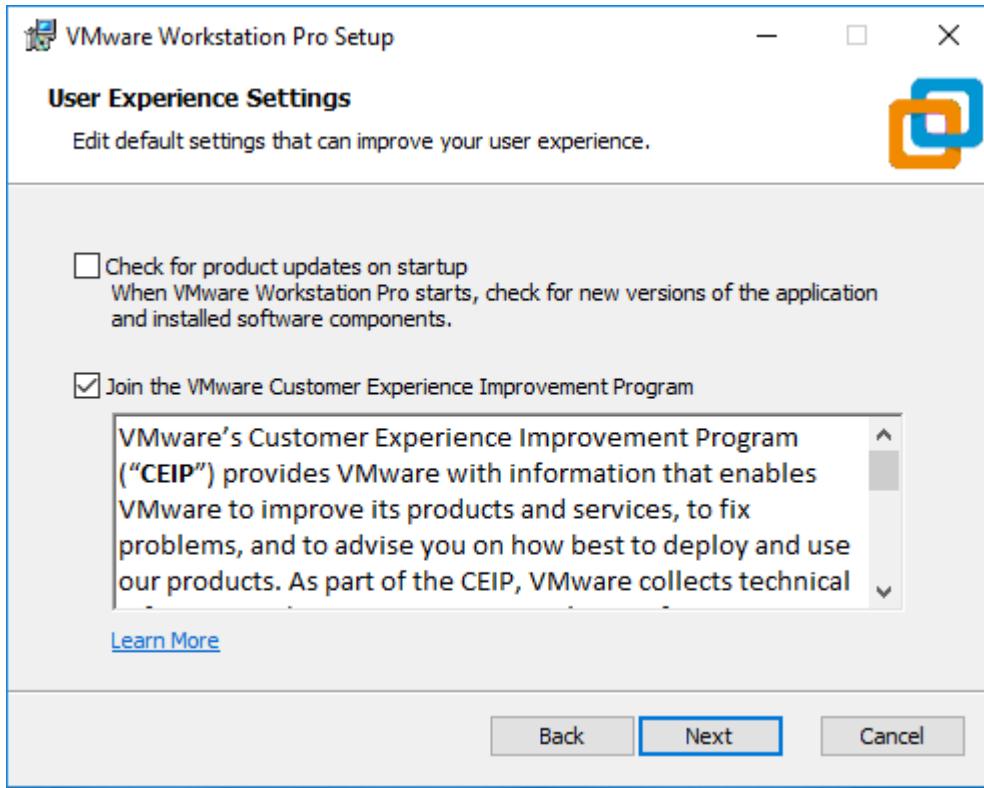


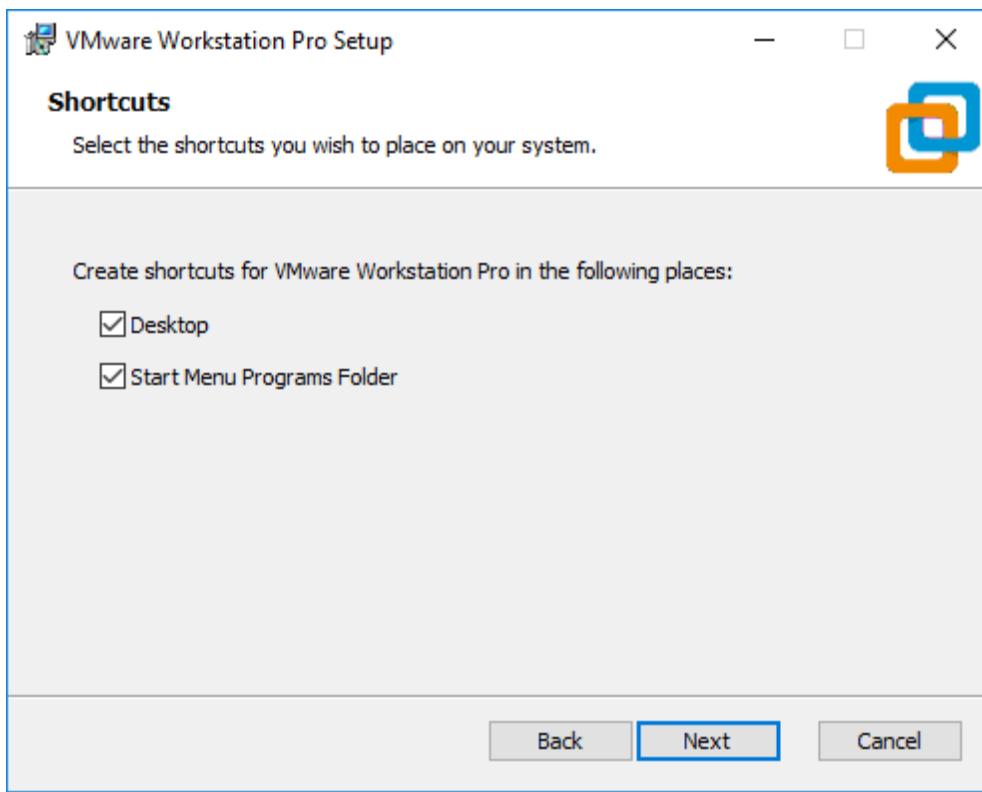
**Step 6.** In the next screen, It will ask for some additional features, it is not mandatory to check this box. Click on Next.



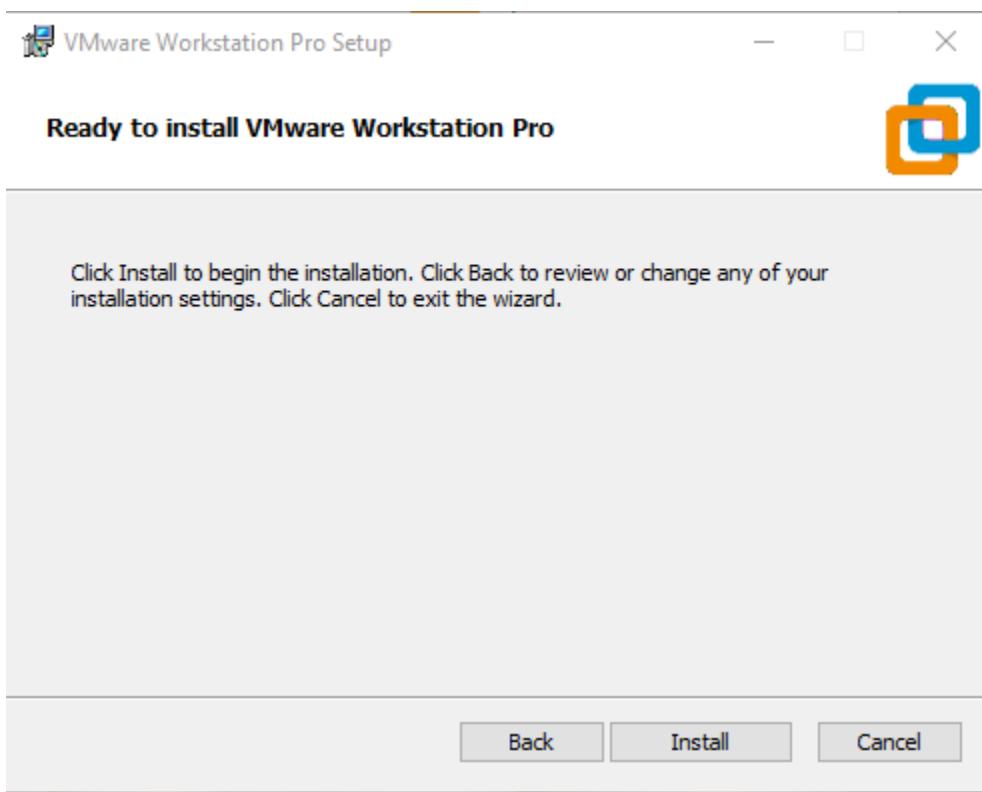
**Step 7.** On the next screen, some checkboxes are populated, Check them as per your requirement.

Click on Next.

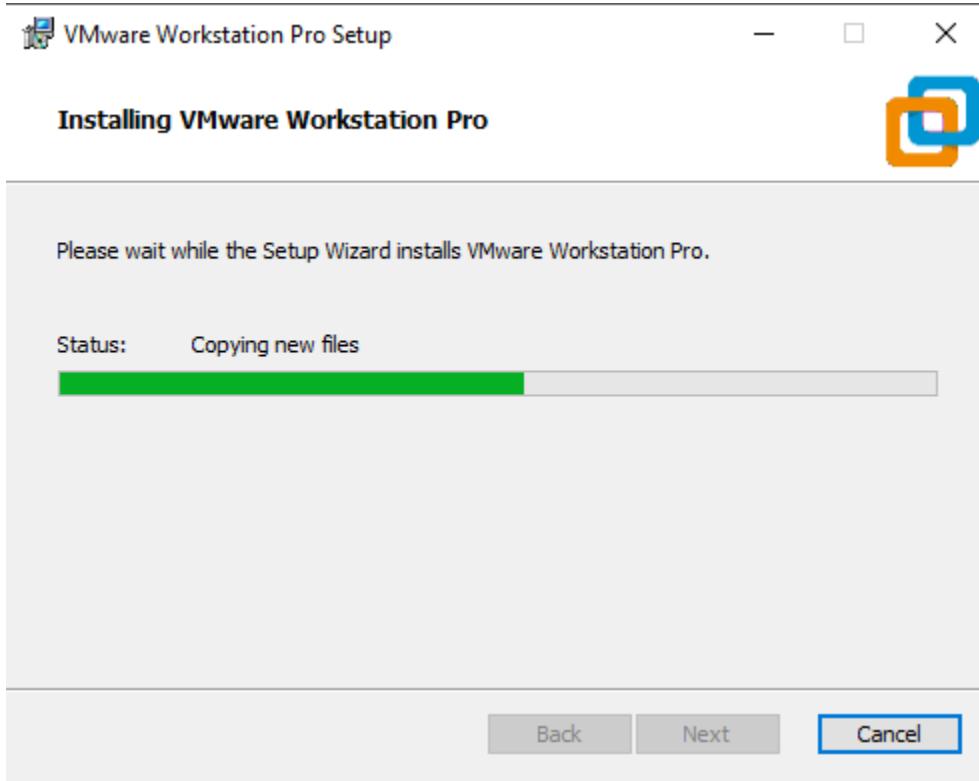




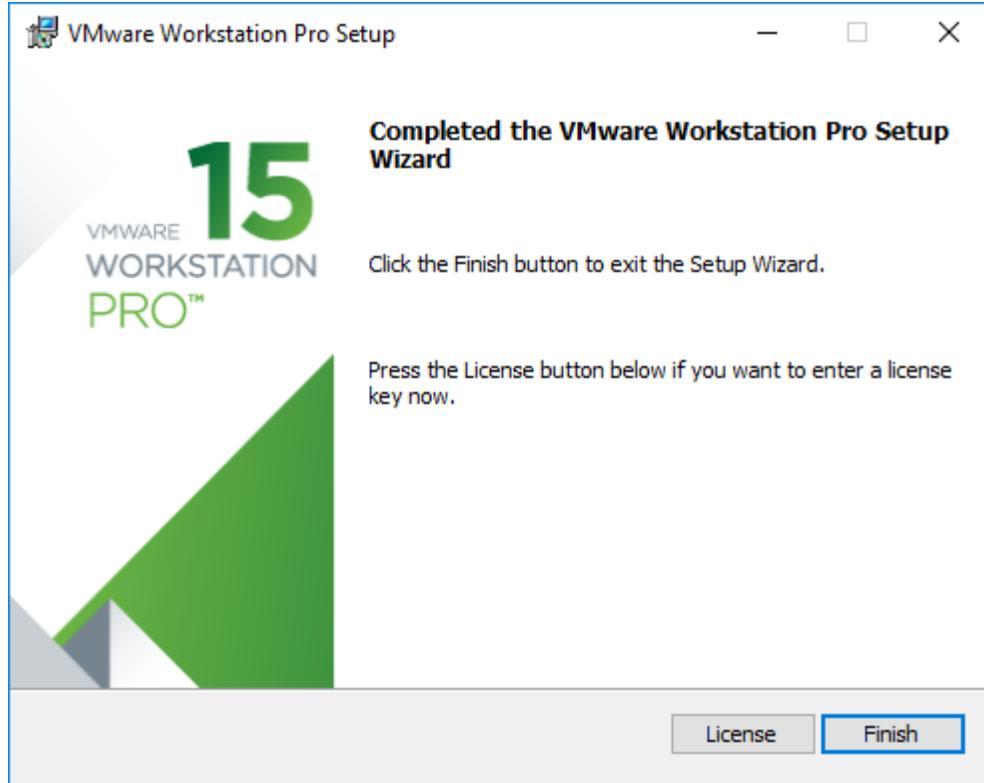
**Step 8.** At this step, VMware Workstation is ready to install. Click on Install.



**Step 9.** At this step you can see installation taking place. The installation will take some time, wait for it to properly install.



**Step 10.** Once the installation gets completed you will see the following dialogue box. Click on Finish. If you have purchased the product and have a license key, then you can click on License to enter key.



**Step 11.** Upon Finish, the window will close, and You can see VMware Workstation installed icon on your Desktop.

The icon looks like this,



Double Click on the Icon to open the application.

**Step 12.** For the first time opening, if you have not entered the License key in step 7, then it will ask for a License key. You can go for the trial version which is available free for 15 to 30 days. Click on Continue. Make sure you have Admin rights for this in Windows.

Welcome to VMware Workstation 15

X



VMware Workstation 15

I have a license key for VMware Workstation 15:

Do you need a license key?

[Buy now](#)

I want to try VMware Workstation 15 for 30 days

Continue

Cancel

At this stage, you will get the final installation message. Click on Finish.

Welcome to VMware Workstation 15

X



VMware Workstation 15

Thank you for evaluating VMware Workstation 15!

VMware Workstation 15 is the most advanced virtualization software that supports the broadest number of operating systems and delivers the richest desktop user experience.

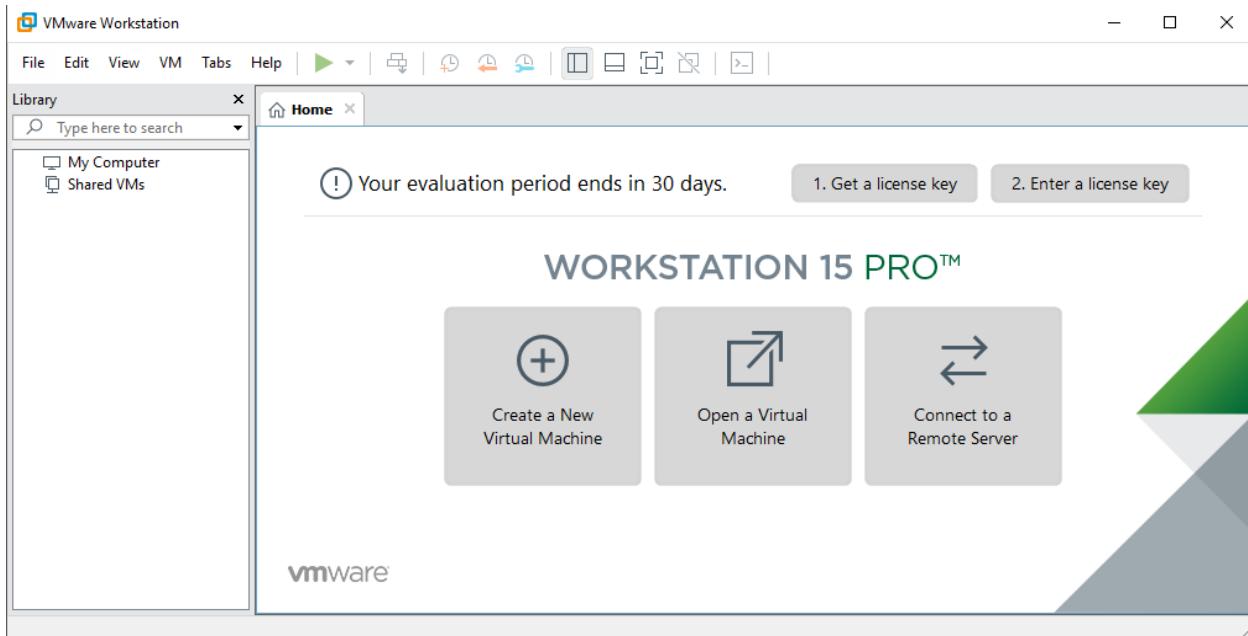
We believe that you will find VMware Workstation 15 to be an indispensable application that improves your productivity and becomes crucial to running your business.

When your evaluation expires, you may purchase a license key from the VMware Online Store or from one of our resellers.

Finish

Cancel

Finally, this will open a window of VMware Workstation Pro.



### **Mirroring:**

#### **Establishing a Database Mirroring Session**

The basic steps for establishing a mirroring session are as follows:

7. Create the mirror database by restoring the following backups, using RESTORE WITH NORECOVERY on every restore operation:
  - a. Restore a recent full database backup of the principal database, after making sure that the principal database was already using the full recovery model when the backup was taken. The mirror database must have the same name as the principal database.
  - b. If you have taken any differential backups of the database since the restored full backup, restore your most recent differential backup.
  - c. Restore all the log backups done since the full or differential database backup.

### **Log shipping:**

It backups the transactional log files to primary database server and restores to them on standby server (for read-only)

Log shipping consists of three operations:

- 1.Back up the transaction log at the primary server instance.
- 2.Copy the transaction log file to the secondary server instance.
- 3.Restore the log backup on the secondary server instance.

The log can be shipped to multiple secondary server instances. In such cases, operations 2 and 3 are duplicated for each secondary server instance.

A log shipping configuration does not automatically fail over from the primary server to the secondary server. If the primary database becomes unavailable, any of the secondary databases can be brought online manually.

You can use a secondary database for reporting purposes.

### **Prerequisites:**

- Same configuration is optional but highly recommended
- Witness server is absent
- Failover is not possible incase of higher to lower configuration but is possible in lower to higher configuration
- Recovery models are only bulk logged or full
- If recovery model is simple transactional log is not supported so logfiles does not exist.
- Shared folder is used only to copy, read only on primary and secondary databases.
- Synchronization is not possible (immediate updates is not possible).
- Database name in primary and secondary group can be same and it may not
- In log shipping database should be selected completely for performing operation.
- It can have both committed and uncommitted transactional logs are transferred to the secondary database.

### **AlwaysOn :**

SQL Server AlwaysOn is a marketing term which refers to the high availability and disaster recovery solution introduced when SQL Server 2012 was launched.

To be more specific, SQL Server AlwaysOn consists of two technologies:

- AlwaysOn Failover Clustering Instances (AlwaysOn FCI)
- AlwaysOn Availability Groups (AlwaysOn AG)

Whilst both technologies have similarities such as requiring Windows Server Failover Clustering (WSFC) as the foundation for its implementation, each is a distinct technology under the Always On umbrella.

## **Always On Failover Cluster Instances (SQL Server)**

- As part of the SQL Server Always On offering, Always On Failover Cluster Instances leverages Windows Server Failover Clustering (WSFC) functionality to provide local high availability through redundancy at the server-instance level—a *failover cluster instance* (FCI).
- An FCI is a single instance of SQL Server that is installed across Windows Server Failover Clustering (WSFC) nodes and, possibly, across multiple subnets. On the network, an FCI appears to be an instance of SQL Server running on a single computer, but the FCI provides failover from one WSFC node to another if the current node becomes unavailable.

## **AlwaysOn Availability Groups:**

- AlwaysOn AG does not require shared disk storage for the server hosting the SQL Server. This SQL Server high availability technology has been an Enterprise feature. This means you cannot configure SQL Server Standard Edition to use AlwaysOn AG with versions prior to SQL Server 2016.
- There is now an option to create a basic availability group with SQL Server 2016 Standard edition which I discuss below.
- When you install SQL Server you select the "New SQL stand-alone installation..." option

## **Encryption:**

- Data security is a critical task for any organization, especially if you store customer personal data such as Customer contact number, email address, social security number, bank and credit card numbers. Our main goal is to protect unauthorized access to data within and outside the organization.
- To achieve this, we start by providing access to relevant persons. We still have a chance that these authorized persons can also misuse the data; therefore, SQL Server provides encryption solutions. We can use these encryptions and protect the data.
- It is a crucial aspect in classifying the data based on the information type and sensitivity. For example, we might have customer DOB in a column and depending upon the requirement, and we should classify it as confidential, highly confidential.
- We have many encryptions available in SQL Server such as Transparent Data Encryption (TDE), Always Encrypted, Static data masking and Dynamic Data Masking. In this article, we will explore column level SQL Server encryption using symmetric keys.

### **TDE:**

*Transparent Data Encryption* (TDE) encrypts SQL Server, Azure SQL Database, and Azure Synapse Analytics data files. This encryption is known as encrypting data at rest.

To help secure a database, you can take precautions like:

- Designing a secure system.

- Encrypting confidential assets.
- Building a firewall around the database servers.

But a malicious party who steals physical media like drives or backup tapes can restore or attach the database and browse its data.

One solution is to encrypt sensitive data in a database and use a certificate to protect the keys that encrypt the data. This solution prevents anyone without the keys from using the data. But you must plan this kind of protection in advance.

TDE does real-time I/O encryption and decryption of data and log files. The encryption uses a database encryption key (DEK). The database boot record stores the key for availability during recovery. The DEK is a symmetric key. It's secured by a certificate that the server's master database stores or by an asymmetric key that an EKM module protects.

TDE protects data at rest, which is the data and log files. It lets you follow many laws, regulations, and guidelines established in various industries. This ability lets software developers encrypt data by using AES and 3DES encryption algorithms without changing existing applications.

Queries used :

1. If it does not already exist, create a database master key (DMK) for the master database. Ensure that the database master key is encrypted by the service master key (SMK).

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'C0mPlex@123456';
```

```
--drop master key
```

```
select * from sys.symmetric_keys
```

--2. Either create or designate an existing certificate for use as the database encryption key (DEK) protector. For the best security, it is recommended that you create a new certificate whose only function is to protect the DEK. Ensure that this certificate is protected by the DMK.

```
CREATE CERTIFICATE tdeCert WITH SUBJECT = 'TDE Certificate';
```

```
select * from sys.certificates
```

```
-- drop CERTIFICATE tdeCert
```

--3. Create a backup of the certificate with the private key and store it in a secure location. (Note that the private key is stored in a separate file—be sure to keep both files). Be sure to maintain backups of the certificate as data loss may occur otherwise.

```
BACKUP CERTIFICATE tdeCert TO FILE = 'H:\BACKUPS\Certificates\tdeCert.cert'
```

```
WITH PRIVATE KEY (
```

```
FILE = 'H:\BACKUPS\Certificates\tdeCert.pvk',
```

```
ENCRYPTION BY PASSWORD = 'ValueMomentum@123456');
```

--4. Optionally, enable SSL on the server to protect data in transit.

--Perform the following steps in the user database. These require CONTROL permissions on the

database.

--5. Create the database encryption key (DEK) encrypted with the certificate designated from step 2 above. This certificate is referenced as a server certificate to distinguish it from other certificates that may be stored in the user database.

```
CREATE DATABASE ENCRYPTION KEY  
WITH ALGORITHM = AES_256  
ENCRYPTION BY SERVER CERTIFICATE tdeCert
```

--6. Enable TDE. This command starts a background thread (referred to as the encryption scan), which runs asynchronously.

```
ALTER DATABASE TDE_DEMO1 SET ENCRYPTION ON
```

```
SELECT * FROM SYS.symmetric_keys  
select * from sys.certificates
```

```
SELECT db_name(database_id) as DBname, encryption_state  
FROM sys.dm_database_encryption_keys
```

### **Column level encryption:**

Encrypting the specified columns which should not be viewed by others are used for column encrypting.

- Create a new database and create **CustomerInfo** table

```
CREATE DATABASE CustomerData;  
Go  
USE CustomerData;  
GO  
  
CREATE TABLE CustomerData.dbo.CustomerInfo  
(CustID      INT PRIMARY KEY,  
 CustName    VARCHAR(30) NOT NULL,  
 BankACCNumber VARCHAR(10) NOT NULL  
) ;  
GO
```

- Insert sample data into **CustomerInfo** table

```
Insert into CustomerData.dbo.CustomerInfo (CustID,CustName,BankACCNumber)  
Select 1,'Rajendra',11111111 UNION ALL
```

```
Select 2, 'Manoj',22222222 UNION ALL  
Select 3, 'Shyam',33333333 UNION ALL  
Select 4,'Akshita',44444444 UNION ALL  
Select 5, 'Kashish',55555555
```

- View the records in **CustomerInfo** table

\*\*\* We use the following steps for column level encryption:

- Create a database master key
- Create a self-signed certificate for SQL Server
- Configure a symmetric key for encryption
- Encrypt the column data
- Query and verify the encryption

#### **Always encryption:**

- Always Encrypted is a feature designed to protect sensitive data, such as credit card numbers or national identification numbers (for example, U.S. social security numbers), stored in Azure SQL Database or SQL Server databases.
- Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to the Database Engine (SQL Database or SQL Server).
- As a result, Always Encrypted provides a separation between those who own the data and can view it, and those who manage the data but should have no access.
- By ensuring on-premises database administrators, cloud database operators, or other high-privileged unauthorized users, can't access the encrypted data, Always Encrypted enables customers to confidently store sensitive data outside of their direct control.

#### **Data masking:**

**Security has been one of the prime concerns of database developers since the inception of database management systems. Various data protection schemes have been introduced to provide secure access to sensitive data.**

One such security feature introduced in [SQL Server 2016](#) is called dynamic data masking. Dynamic data masking (as distinct from [static data masking](#)) is used to hide data from the user on the client-side.

It is important to mention that data masking is not the same as data encryption and should not be used as a primary security layer. It is only used to mask sensitive information such as social security numbers, email addresses, phone numbers, credit card numbers, etc. Following are the examples of dynamic data masking:

1.Phone number -> xxxx-xxx-xx-5458

2.Email -> [xxxx@xxxx.com](mailto:xxxx@xxxx.com)

The data is actually not masked physically in the database. Rather, the data is masked in the query result. By default, all the users will see the masked data in the output. The unmasked data is visible in the actual database.

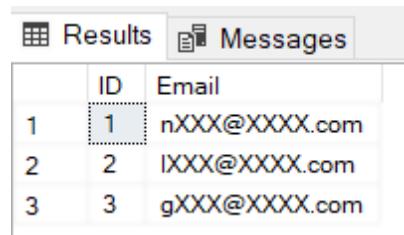
### **Data masking types:**

#### **1.Default**

The default mask, masks complete values in the specified column. To specify a mask for a particular column, you have to use the "MASKED WITH" clause. Inside the MASKED WITH clause, you have to specify the FUNCTION that you want to use for masking. If you want to perform default masking, you use the "default()" function.

#### **2.Email**

The email mask is used to dynamically mask data which is in the email format. The function used is "email()".



ID	Email
1	nXXX@XXXX.com
2	lXXX@XXXX.com
3	gXXX@XXXX.com

#### **3.Random**

The Random mask is used to mask the integer columns with random values. The range for random values is specified by the random function

### **Replication:**

- Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency.
- Use replication to distribute data to different locations and to remote or mobile users over local and wide area networks, dial-up connections, wireless connections, and the Internet.

### **Types of Replication:**

- 1)Snapshot Replication
- 2)Transaction Replication
- 3)Merge Replication
- 4)Peer to Peer Replication

**Transaction Replication:** Changes at the Publisher are delivered to the Subscriber as they occur (in near real time). The data changes are applied to the Subscriber in the same order and within the same transaction boundaries as they occurred on the publisher.

**Snapshot Replication:** Applies a snapshot from the Publisher to the Subscriber, which distributes data exactly as it appears at a specific moment in time, and does not monitor for updates to the data. When synchronization occurs, the entire snapshot is generated and sent to Subscribers.

**Merge Replication:** Data can be changed on both the Publisher and Subscriber, and are tracked with triggers. The Subscriber synchronizes with the Publisher when connected to the network and exchanges all rows that have changed between the Publisher and Subscriber since the last time synchronization occurred.

**Peer to Peer Replication:** Built on the foundation of transactional replication, peer-to-peer replication propagates transactionally consistent changes in near real-time between multiple server instances.

### **Attaching and detaching of a database:**

- The data and transaction log files of a database can be detached and then reattached to the same or another instance of SQL Server.
- Detaching and attaching a database is useful if you want to change the database to a different instance of SQL Server on the same computer or to move the database.

#### 1. Detaching a database:

- Detaching a database removes it from the instance of SQL Server but leaves the database intact within its data files and transaction log files.
- These files can then be used to attach the database to any instance of SQL Server, including the server from which the database was detached.

#### Backup and Restore and Detach:

- Detaching a read-only database loses information about the differential bases of differential backups.

#### 2. Attaching a database:

- When you attach a database, all data files (MDF and NDF files) must be available. If any data file has a different path from when the database was first created or last attached, you must specify the current path of the file.
- When an encrypted database is first attached to an instance of SQL Server, the database owner must open the master key of the database by executing the following statement:  
OPEN MASTER KEY DECRYPTION BY PASSWORD = 'password'. We recommend that you enable automatic decryption of the master key by executing the following statement:  
ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY

Backup and Restore and Attach:

- Like any database that is fully or partially offline, a database with restoring files cannot be attached. If you stop the restore sequence, you can attach the database. Then, you can restart the restore sequence.

## **Shrinking:**

- Shrinking data files recovers space by moving pages of data from the end of the file to unoccupied space closer to the front of the file.
- When enough free space is created at the end of the file, data pages at end of the file can be deallocated and returned to the file system.

### **To shrink a data or log file**

- In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.
- Expand Databases and then right-click the database that you want to shrink.
- Point to Tasks, point to Shrink, and then click Files.
- Database  
Displays the name of the selected database.
- File type  
Select the file type for the file. The available choices are Data and Log files. The default selection is Data. Selecting a different filegroup type changes the selections in the other fields accordingly.
- Filegroup  
Select a filegroup from the list of Filegroups associated with the selected File type above. Selecting a different filegroup changes the selections in the other fields accordingly.
- File name  
Select a file from the list of available files of the selected filegroup and file type.
- Location  
Displays the full path to the currently selected file. The path is not editable, but it can be copied to the clipboard.
- Currently allocated space  
For data files, displays the current allocated space. For log files, displays the current allocated space computed from the output of DBCC SQLPERF(LOGSPACE).
- Available free space  
For data files, displays the current available free space computed from the output of DBCC SHOWFILESTATS(fileid). For log files, displays the current available free space computed from the output of DBCC SQLPERF(LOGSPACE).
- Release unused space  
Cause any unused space in the files to be released to the operating system and shrink

the file to the last allocated extent, reducing the file size without moving any data. No attempt is made to relocate rows to unallocated pages.

- Reorganize pages before releasing unused space  
Equivalent to executing DBCC SHRINKFILE specifying the target file size. When this option is selected, the user must specify a target file size in the Shrink file to box.
- Shrink file to  
Specifies the target file size for the shrink operation. The size cannot be less than the current allocated space or more than the total extents allocated to the file. Entering a value beyond the minimum or the maximum will revert to the min or the max once the focus is changed or when any of the buttons on the toolbar are clicked.
- Empty file by migrating the data to other files in the same filegroup  
Migrate all data from the specified file. This option allows the file to be dropped using the ALTER DATABASE statement. This option is equivalent to executing DBCC SHRINKFILE with the EMPTYFILE option.
- Select the file type and file name.
- Optionally, select the Release unused space check box.
- Selecting this option causes any unused space in the file to be released to the operating system and shrinks the file to the last allocated extent. This reduces the file size without moving any data.
- Optionally, select the Reorganize files before releasing unused space check box. If this is selected, the Shrink file to value must be specified. By default, the option is cleared.
- Selecting this option causes any unused space in the file to be released to the operating system and tries to relocate rows to unallocated pages.
- Optionally, enter the maximum percentage of free space to be left in the database file after the database has been shrunk. Permissible values are between 0 and 99. This option is only available when Reorganize files before releasing unused space is enabled.
- Optionally, select the Empty file by migrating the data to other files in the same filegroup check box.
- Selecting this option moves all data from the specified file to other files in the filegroup. The empty file can then be deleted. This option is the same as executing DBCC SHRINKFILE with the EMPTYFILE option.
- Click OK.

## **Performance Monitoring:**

The Performance Dashboard helps to quickly identify whether SQL Server or Azure SQL Database is experiencing a performance bottleneck. And if a bottleneck is found, easily capture additional diagnostic data that may be necessary to resolve the problem. Some common performance problems which the Performance Dashboard can help identify include:

- CPU bottlenecks (and what queries are consuming the most CPU)
- I/O bottlenecks (and what queries are performing the most I/O)

- Index recommendations generated by the Query Optimizer (missing indexes)
- Blocking
- Resource contention (including latch contention)

The Performance Dashboard also helps to identify expensive queries that may have been executed before, and several metrics are available to define high cost: CPU, Logical Writes, Logical Reads, Duration, Physical Reads, and CLR Time.

The Performance dashboard is divided into the following sections and sub-reports:

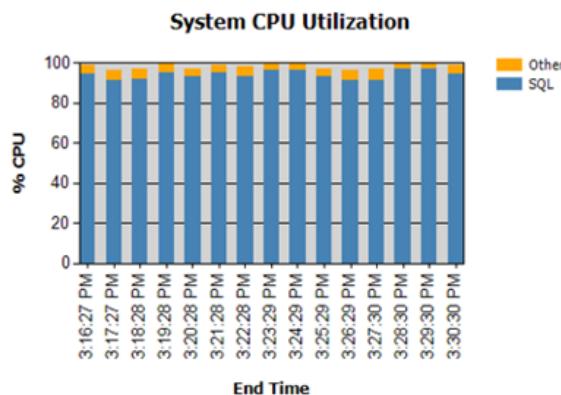
- System CPU Utilization
- Current Waiting Requests
- Current Activity
  - User Requests
  - User Sessions
  - Cache Hit Ratio
- Historical Information
  - Waits
  - Latches
  - I/O Statistics
  - Expensive Queries
- Miscellaneous Information
  - Active Traces
  - Active xEvent Sessions
  - Databases
  - Missing Indexes

## Microsoft SQL Server Performance Dashboard

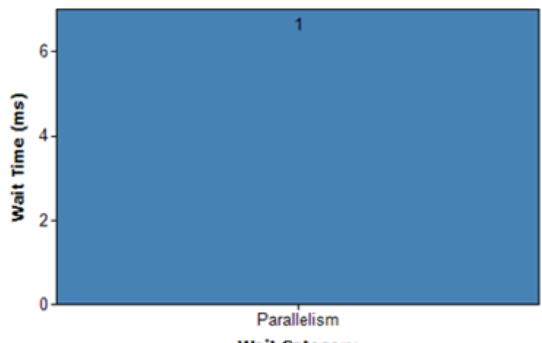
Report Local Time: [REDACTED] 3:31:04 PM

[REDACTED] (SQL2016(13.0.4422.0 - Enterprise Edition (64-bit)))

 Overall performance may be degraded because the system shows signs of being CPU-bound. This SQL Server instance is consuming the majority of the CPU. Click on any of the SQL data points in the chart below to investigate further.



### Current Waiting Requests



### Current Activity

	User Requests	User Sessions
Count	27	32
Elapsed Time (ms)	4573004	741818
CPU Time (ms)	2043203(44.68%)	101108(13.63%)
Wait Time (ms)	2529801(55.32%)	640710(86.37%)
Cache Hit Ratio	100.000%	98.313%

### Historical Information

<a href="#">Waits</a>	<a href="#">IO Statistics</a>
<a href="#">Latencies</a>	
<b>Expensive Queries</b>	
<a href="#">By CPU</a>	<a href="#">By Duration</a>
<a href="#">By Logical Reads</a>	<a href="#">By Physical Reads</a>
<a href="#">By Logical Writes</a>	<a href="#">By CLR Time</a>

### Miscellaneous Information

<a href="#">Active Traces</a>	1
<a href="#">Active Xevent Sessions</a>	4
<a href="#">Databases</a>	16
<a href="#">Missing Indexes</a>	11

## Deadlock Scenario:

- A deadlock occurs when there is a cyclic dependency between two or more threads, or processes, for some set of resources within SQL Server.
- Using SQL Server Profiler, you can create a trace that records, replays, and displays deadlock events for analysis.
- To trace deadlock events, add the Deadlock graph event class to a trace. This event class populates the TextData data column in the trace with XML data about the process and objects that are involved in the deadlock.
- SQL Server Profiler can extract the XML document to a deadlock XML (.xdl) file which you can view later in SQL Server Management Studio.

SQL Server Profiler to extract Deadlock graph events to a single file that contains all Deadlock graph events, or to separate files. This extraction can be done in any of the following ways:

- At trace configuration time, using the Events Extraction Settings tab. Note that this tab does not appear until you select the Deadlock graph event on the Events Selection tab.
- Using the Extract SQL Server Events option on the File menu.
- Individual events can also be extracted and saved by right-clicking a specific event and choosing Extract Event Data.

## **Deadlock Graphs**

SQL Server Profiler and SQL Server Management Studio use a deadlock wait-for graph to describe a deadlock. The deadlock wait-for graph contains process nodes, resource nodes, and edges representing the relationships between the processes and the resources. The components of wait-for graphs are defined in the following table:

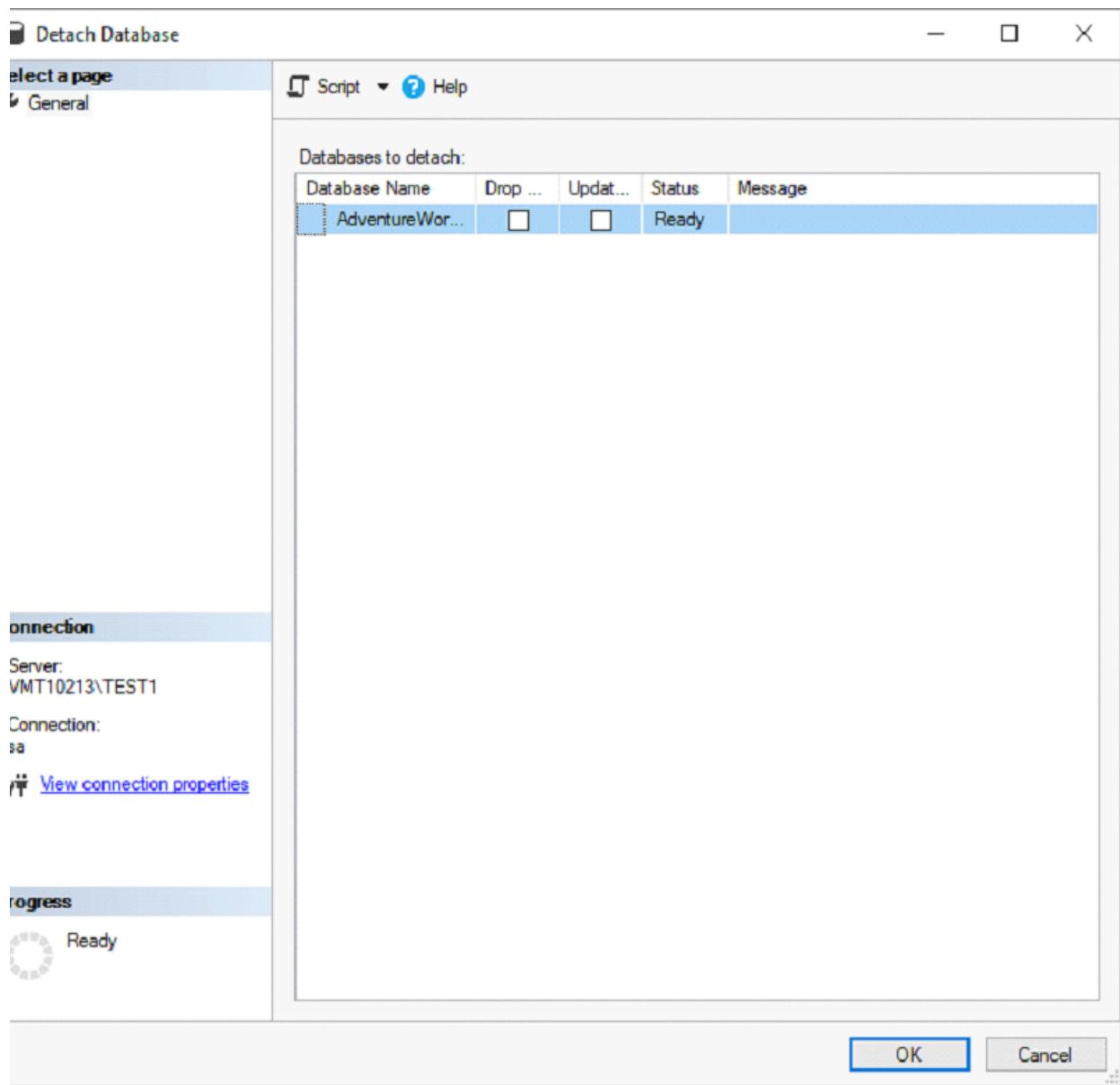
- Process node:  
A thread that performs a task; for example, INSERT, UPDATE, or DELETE.
- Resource node:  
A database object; for example, a table, index, or row.
- Edge:  
A relationship between a process and a resource. A **request** edge occurs when a process waits for a resource. An **owner** edge occurs when a resource waits for a process. The lock mode is included in the edge description.

## **Assignment:**

1.Detaching database adventureworks2017:

```
use [AdventureWorks2017];
GO
use [master];
GO
USE [master]
GO
ALTER DATABASE [AdventureWorks2017] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
GO
USE [master]
GO
EXEC master.dbo.sp_detach_db @dbname = N'AdventureWorks2017', @skipchecks = 'false'
GO
```

```
USE [master]
GO
EXEC master.dbo.sp_detach_db @dbname = N'AdventureWorks2017'
GO
```



2.Attaching a database :

```
USE [master]
```

```
GO
```

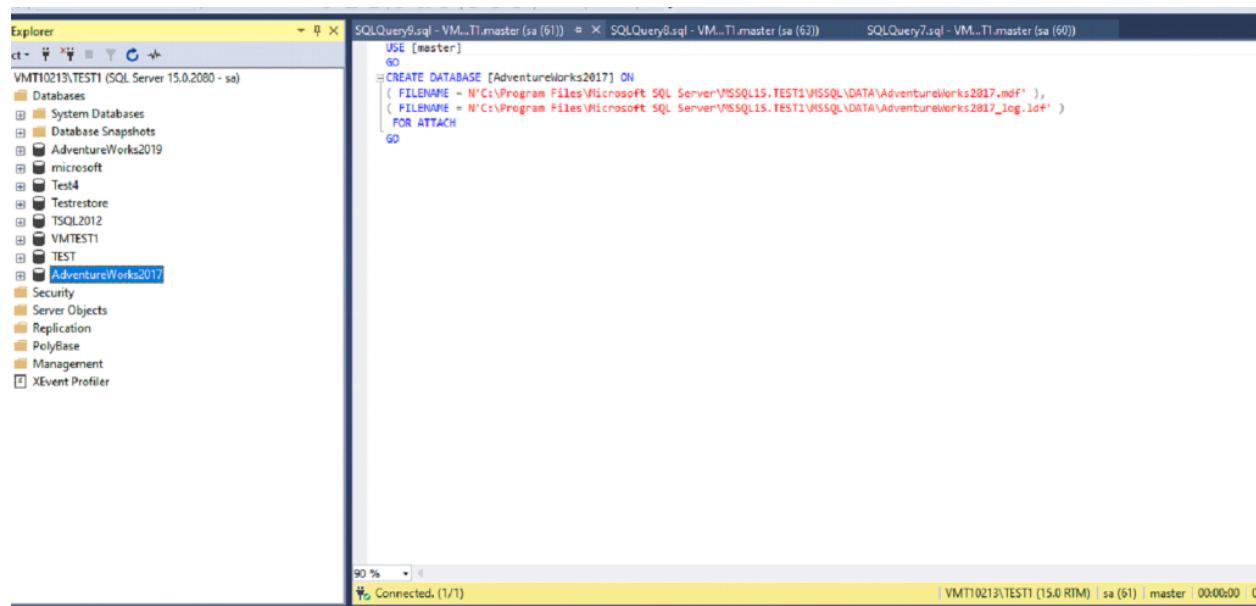
```
CREATE DATABASE [AdventureWorks2017] ON
```

```
( FILENAME = N'C:\Program Files\Microsoft SQL  
Server\MSSQL15.TEST1\MSSQL\DATA\AdventureWorks2017.mdf' ),
```

```
( FILENAME = N'C:\Program Files\Microsoft SQL  
Server\MSSQL15.TEST1\MSSQL\DATA\AdventureWorks2017_log.ldf' )
```

```
FOR ATTACH
```

```
GO
```



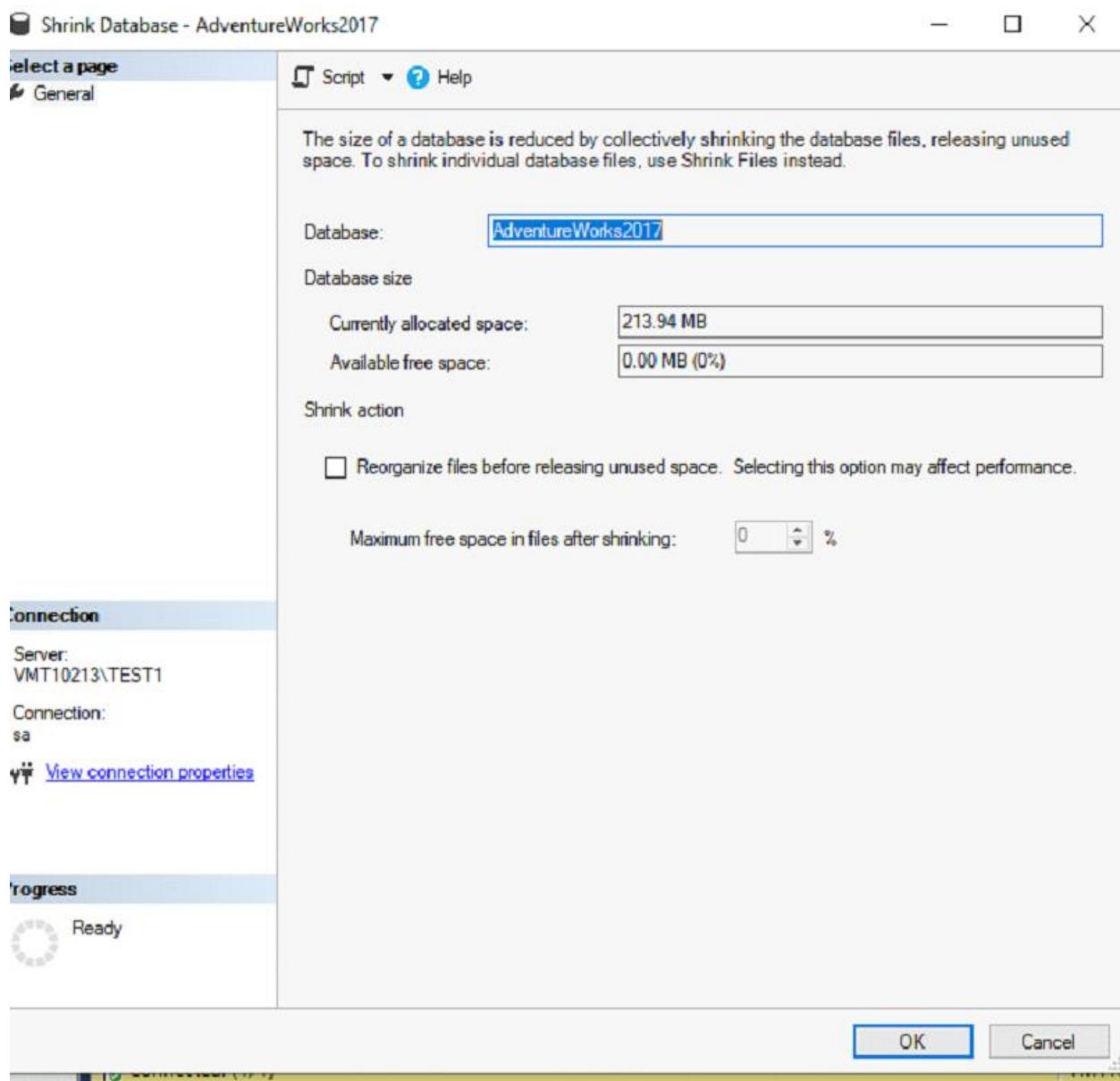
### 3. Shrinking of database adventureworks2017

```
USE [AdventureWorks2017]
```

```
GO
```

```
DBCC SHRINKDATABASE(N'AdventureWorks2017' )
```

```
GO
```



For files:

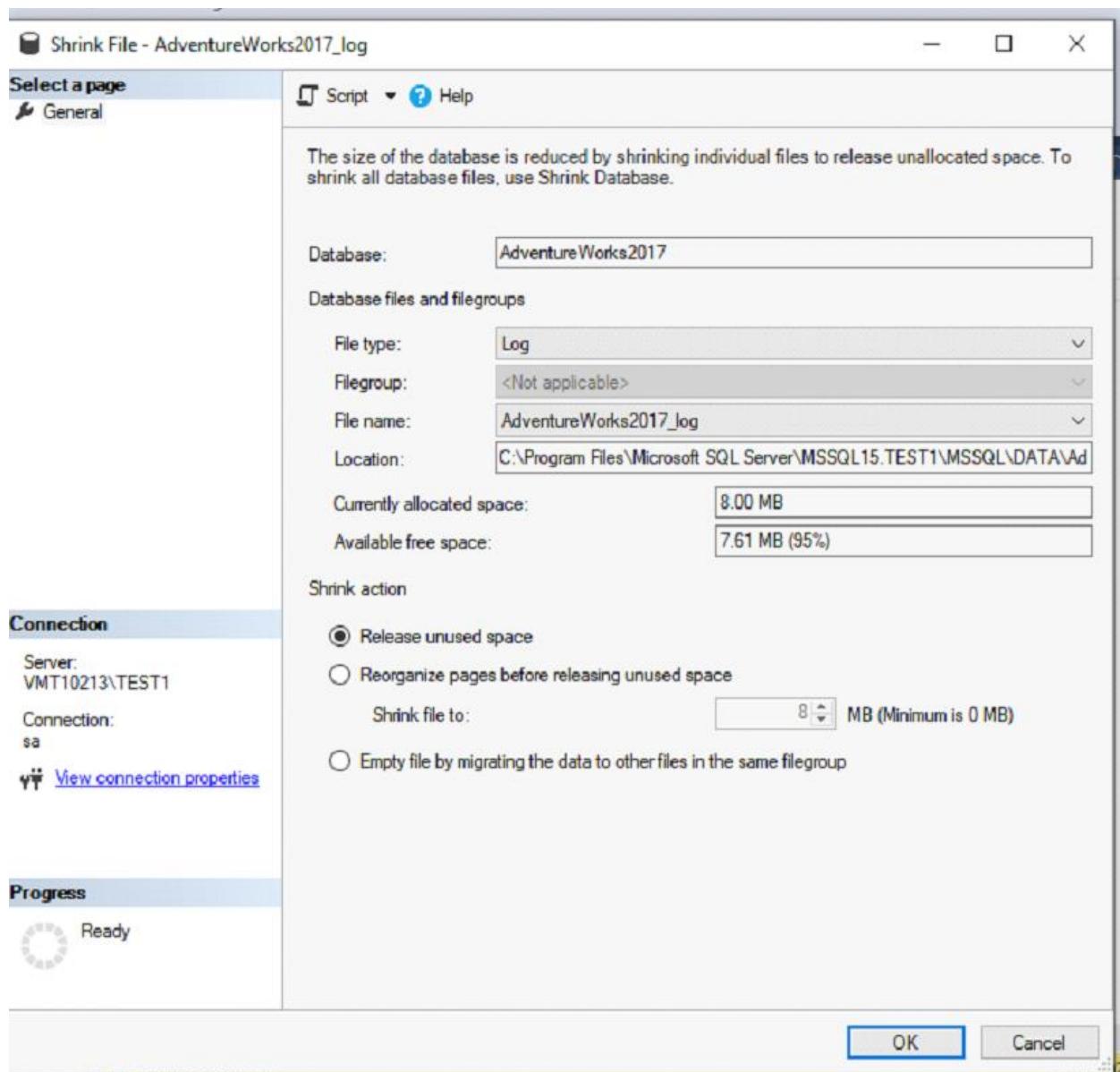
```
USE [AdventureWorks2017]
```

```
GO
```

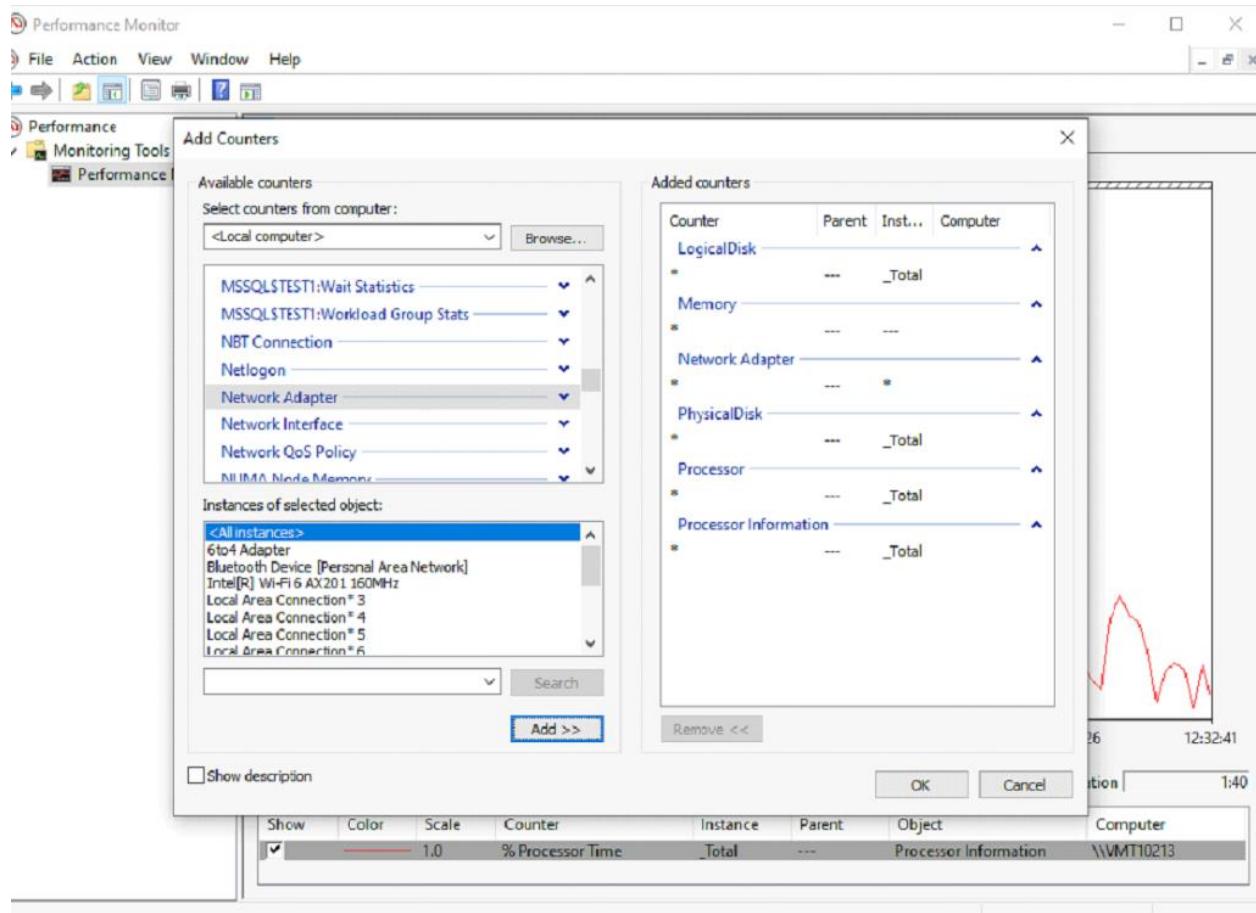
```
DBCC SHRINKFILE (N'AdventureWorks2017_log' , 0, TRUNCATEONLY)
```

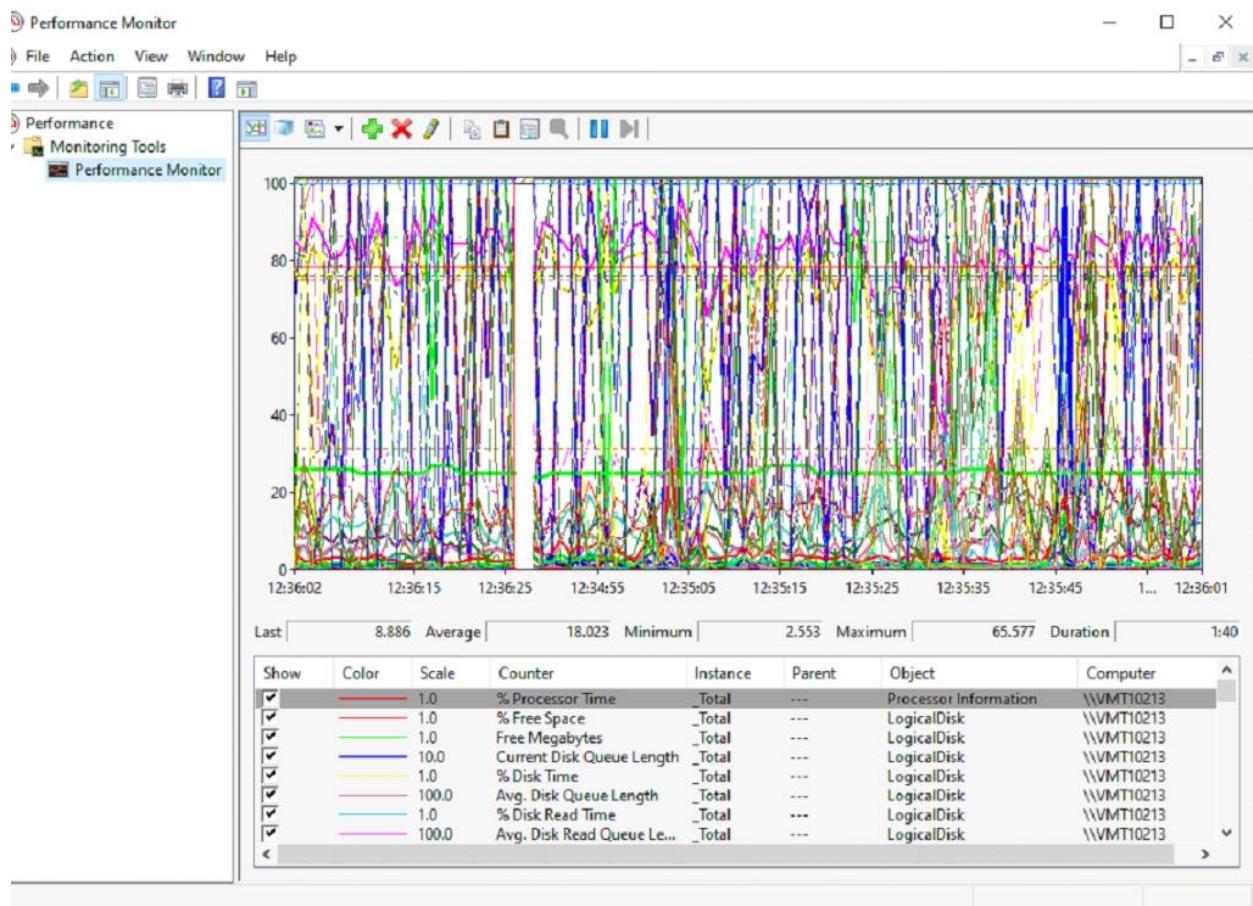
```
GO
```

Performance



4. Performance Monitoring:





## 5. Deadlock scenario using sql profiler

--Deadlock Thread 1

BEGIN TRANSACTION

UPDATE [Person].[PersonPhone]

SET PhoneNumber = '999-555-1212'

WHERE [BusinessEntityID] = 1

WAITFOR DELAY '00:00:10'

UPDATE [Person].[Person]

SET [FirstName] = 'Chris', [LastName] = 'Skorlinski'

WHERE [BusinessEntityID] = 1

ROLLBACK TRANSACTION

--Deadlock Thread 2

BEGIN TRANSACTION

UPDATE [Person].[Person]

SET [FirstName] = 'Chris', [LastName] = 'Skorlinski'

WHERE [BusinessEntityID] = 1

UPDATE [Person].[PersonPhone]

SET PhoneNumber = '999-555-1212'

WHERE [BusinessEntityID] = 1

WAITFOR DELAY '00:00:10'

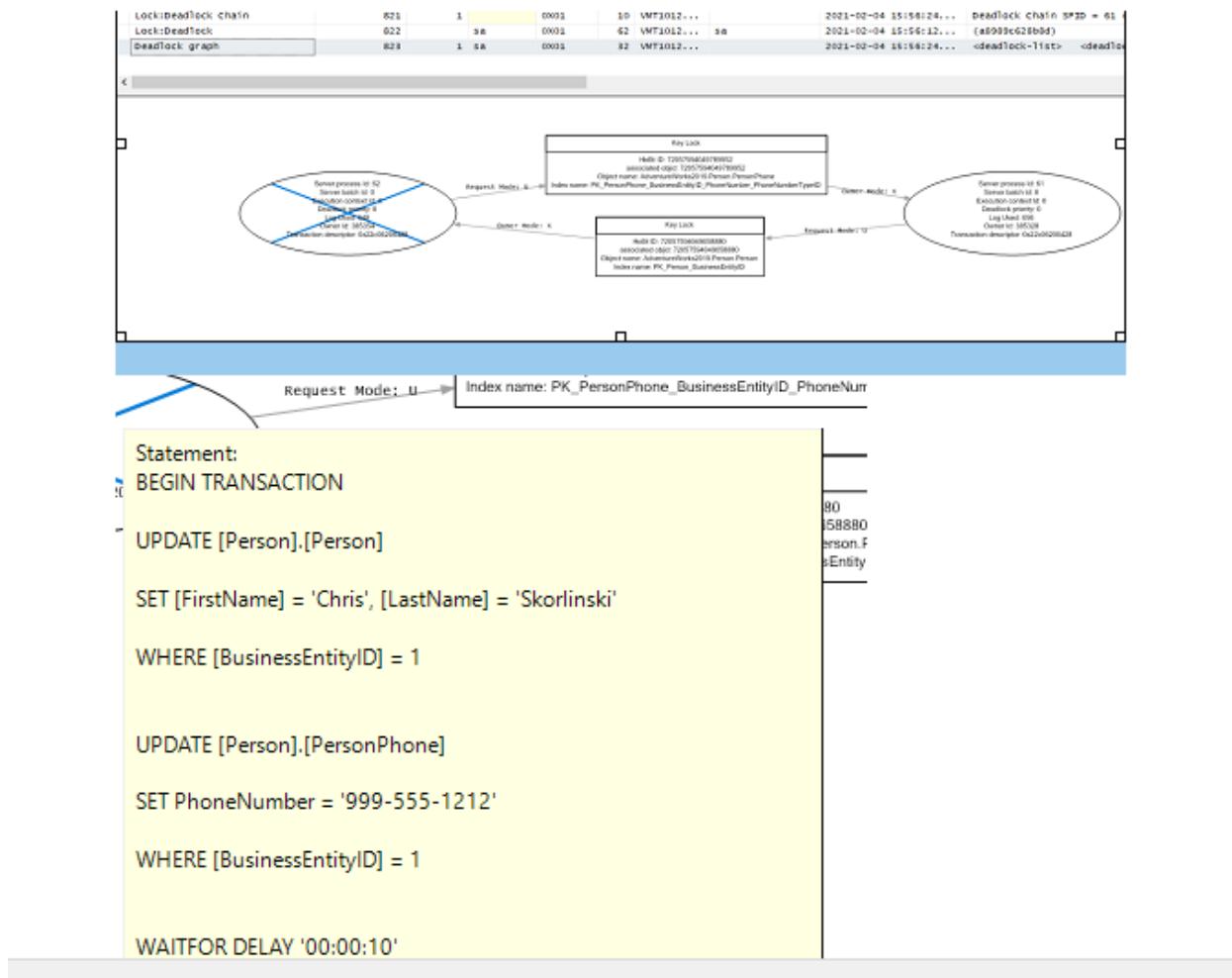
ROLLBACK TRANSACTION



The screenshot shows a 'Messages' window in SSMS. The title bar says 'Messages'. The content area displays the following text:

```
row affected)
1205, Level 12, State 48, Line 12
Transaction (Process ID 54) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Completion time: 2021-02-04T15:44:52.7554255+05:30
```



## Isolation Levels:

SQL Server has 4 isolation levels available.

**1. READ UNCOMMITTED:** This says that a transaction within the current session can read data that has been modified or deleted by another transaction but not yet committed. This imposes the least restrictions of isolation levels as the database engine doesn't issue any shared locks. As a result of this, it is highly likely that the transaction will end up reading data that has been inserted, updated, or deleted but never committed to the database. Such a scenario is known as *dirty reads*.

**2. READ COMMITTED:** This is the default setting for most SQL Server queries. It defines that a transaction within the current session cannot read data that has been modified by another transaction. For this reason, dirty reads are prevented when this setting is turned on.

**3. REPEATABLE READ:** In this setting, a transaction not only can read data that is modified by another transaction that has been committed but also imposes a restriction that no other

transaction can modify the data that is being read until the first transaction completes. This eliminates the condition of non-repeatable read

**4.SERIALIZABLE:** There are multiple properties that are being set by this isolation level. This isolation level is the most restrictive as compared to the others, thus there might be some performance issues with this. The properties are mentioned as below:

- a. The current transaction can only read data modified by other transaction that has been committed.
- b. Other transactions have to wait in queue until the execution of the first transaction is completed.
- c. No other transactions are allowed to insert data, which matches the condition of the current transaction.

IF OBJECT\_ID('Emp') is not null

```
begin  
DROP TABLE Emp  
end
```

```
create table Emp(ID int,Name Varchar(50),Salary Int)
```

```
insert into Emp(ID,Name,Salary)  
values( 1,'David',1000)  
insert into Emp(ID,Name,Salary)  
values( 2,'Steve',2000)  
insert into Emp(ID,Name,Salary)  
values( 3,'Chris',3000)
```

```
select * from Emp
```

#### **READ COMMITTED:**

```
begin tran  
update emp set Salary=999 where ID=1  
waitfor delay '00:00:15'  
Commit
```

In another new query window execute the below one

set transaction isolation level read committed

```
select Salary from Emp where ID=1
```

```
SQLQuery2.sql - VM...T3.master (sa (60))*
set transaction isolation level read committed
select Salary from Emp where ID=1

SQLQuery1.sql - VM...T3.master (sa (56))*
100 %
Results Messages
Salary
1 999
```

### **READ UNCOMMITTED**

```
begin tran
```

```
update emp set Salary=999 where ID=1
```

```
waitfor delay '00:00:15'
```

Rollback

IN ANOTHER NEW QUERY WINDOW EXECUTE THE BELOW ONE

set transaction isolation level read uncommitted

select Salary from Emp where ID=1

### **REPEATABLE READ**

set transaction isolation level repeatable read

begin tran

select \* from emp where ID in(1,2)

waitfor delay '00:00:15'

select \* from Emp where ID in (1,2)

rollback

IN ANOTHER NEW QUERY WINDOW EXECUTE THE BELOW ONE

update emp set Salary=999 where ID=1

ery2.sql - VM...T3.master (sa (60))\*

SQLQuery1.sql - VM...T3.master (sa (56))\*

```
begin tran
    update emp set Salary=999 where ID=1
    waitfor delay '00:00:15'
    commit

begin tran
    update emp set Salary=999 where ID=1
    waitfor delay '00:00:15'
    rollback

    set transaction isolation level repeatable read
begin tran
    select * from emp where ID in(1,2)
    waitfor delay '00:00:15'
    select * from Emp where ID in (1,2)
    rollback
```

Results Messages

ID	Name	Salary
1	David	999
2	Steve	2000

ID	Name	Salary
1	David	999
2	Steve	2000

## SERIALIZABLE

```
set transaction isolation level serializable
```

```
begin tran
    select * from emp
    waitfor delay '00:00:15'
    select * from Emp
    Rollback
```

IN ANOTHER NEW QUERY WINDOW EXECUTE THE BELOW ONE

```
insert into Emp(ID,Name,Salary)
```

```
values( 11,'Stewart',11000)
```

```
Query2.sql - VM... IS.master (sa (60))" SQLQuery1.sql - VM... IS.master (sa (50))"
begin tran
select * from emp where ID in(1,2)
waitfor delay '00:00:15'
select * from Emp where ID in (1,2)
rollback

set transaction isolation level serializable
begin tran
select * from emp
waitfor delay '00:00:15'
select * from Emp
rollback
```



ID	Name	Salary
1	David	999
2	Steve	2000
3	Chris	3000



ID	Name	Salary
1	David	999
2	Steve	2000
3	Chris	3000

## SNAPSHOT:

Snapshot isolation is similar to Serializable isolation. The difference is Snapshot does not hold lock on table during the transaction so table can be modified in other sessions. Snapshot isolation

maintains versioning in Tempdb for old data in case of any data modification occurs in othersessions then existing transaction displays the old data from Tempdb.

```
set transaction isolation level snapshot
```

```
begin tran
```

```
select * from Emp
```

```
waitfor delay '00:00:15'
```

```
select * from Emp
```

```
Rollback
```

IN ANOTHER NEW QUERY WINDOW EXECUTE THE BELOW

```
insert into Emp(ID,Name,Salary) values( 11,'Stewart',11000)
```

```
update Emp set Salary=4444 where ID=4
```

```
select * from Emp
```

```
set transaction isolation level serializable
begin tran
select * from emp
waitfor delay '00:00:15'
select * from Emp
rollback
```

```
set transaction isolation level snapshot
begin tran
select * from Emp
waitfor delay '00:00:15'
select * from Emp
rollback
```

#### ts Messages

Name	Salary
David	999
Steve	2000
Chris	3000
Stewart	11000

---

Name	Salary
David	999
Steve	2000
Chris	3000
Stewart	11000

---

```
USE [master]
```

```
GO
```

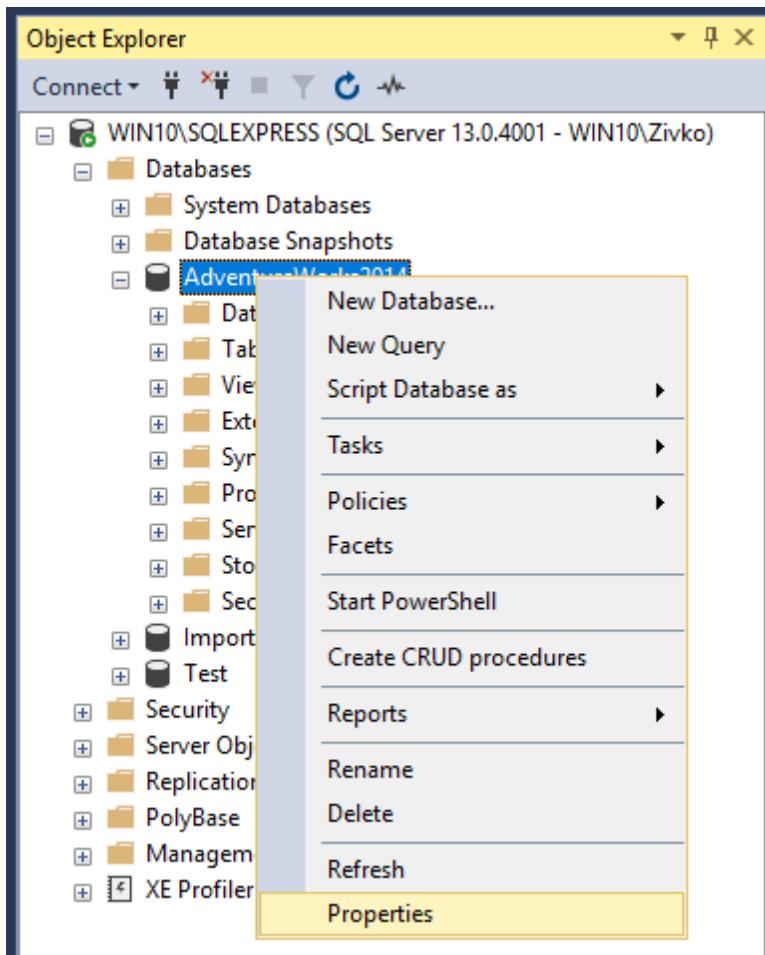
```
ALTER DATABASE [AdventureWorks2016] SET READ_COMMITTED_SNAPSHOT ON WITH NO_WAIT
```

```
Go
```

## Query Store:

The SQL Server Query Store is a relatively new feature introduced in SQL Server 2016. It is basically a SQL Server “flight recorder” or “black box”, capturing a history of executed queries, query runtime execution statistics, execution plans etc. against a specific database. This information helps in

identifying performance problems caused by query plan changes and troubleshooting by quickly finding performance differences, even after SQL Server restart or upgrade. All data that SQL Server Query Store capture are stored on disk.



Database Properties - AdventureWorks2014

Select a page

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Query Store

Script Help

General

Operation Mode (Actual)	Off
Operation Mode (Requested)	Off

Monitoring

Data Flush Interval (Minutes)	
Statistics Collection Interval	

Query Store Retention

Max Size (MB)	
Query Store Capture Mode	

**Data Flush Interval (Minutes)**  
The frequency at which query store data is flushed and persisted to disk.

Connection

Server: WIN10\SQLEXPRESS  
Connection: WIN10\Zivko  
[View connection properties](#)

Progress

Ready

Current Disk Usage

AdventureWorks2014	223.3 MB
Query Store Used	0.0 MB

Query Store Available	100.0 MB
Query Store Used	0.0 MB

Purge Query Data

OK Cancel

Database Properties - AdventureWorks2014

Select a page

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Query Store

Script Help

General

Operation Mode (Actual)	Off
Operation Mode (Requested)	Off

Monitoring

Data Flush Interval (Minutes)	Off
Statistics Collection Interval	Read Only

Query Store Retention

Max Size (MB)	
Query Store Capture Mode	
Size Based Cleanup Mode	
Stale Query Threshold (Days)	

**Operation Mode (Requested)**  
The requested query store operation mode

Connection

Server: WIN10\SQLEXPRESS  
Connection: WIN10\Zivko

[View connection properties](#)

Progress

Ready

Current Disk Usage

AdventureWorks2014	223.3 MB
Query Store Used	0.0 MB

Query Store Available	100.0 MB
Query Store Used	0.0 MB

Purge Query Data

OK Cancel

Database Properties - AdventureWorks2014

Select a page

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Query Store

Script Help

General

Operation Mode (Actual)	Off
Operation Mode (Requested)	<b>Read Write</b>

Monitoring

Data Flush Interval (Minutes)	<b>15</b>
Statistics Collection Interval	<b>1 Hour</b>

Query Store Retention

Max Size (MB)	<b>100</b>
Query Store Capture Mode	<b>All</b>
Size Based Cleanup Mode	<b>Auto</b>
Stale Query Threshold (Days)	<b>367</b>

**Stale Query Threshold (Days)**  
Duration to retain query store runtime statistics.

Connection

Server: WIN10\SQLEXPRESS

Connection: WIN10\Zivko

[View connection properties](#)

Progress

Scripting completed successfully.

Current Disk Usage

AdventureWorks2014	223.3 MB
Query Store Used	0.0 MB

Query Store Available	100.0 MB
Query Store Used	0.0 MB

Purge Query Data

OK Cancel

Object Explorer

Connect ▾

- WIN10\SQLEXPRESS (SQL Server 13.0.4001 - WIN10\Zivko)
  - Databases
    - System Databases
    - Database Snapshots
    - AdventureWorks2014**
  - Query Store
    - Regressed Queries
    - Overall Resource Consumption
    - Top Resource Consuming Queries
    - Queries With Forced Plans
    - Queries With High Variation
    - Tracked Queries
  - Service Broker
  - Storage
  - Security

Top 25 regressed queries for database AdventureWorks2016. Time period: Last hour ending at 05-02-2021 13:00

Metric	Duration (ms)	Statistic	Total
query_id	object_id	object_name	query_sql_text
1	15	0	select * from [Sales].[Customer]

Plan summary for query 15

Plan 1 [not forced]

Query 1: Query cost (relative to the batch): 100%  
select \* from [Sales].[Customer]

Force Plan | Unforce Plan

Execution Plan Diagram:

```

    graph LR
        A[SELECT Cost: 0 $] --> B[Compute Scalar Cost: 2 $]
        B --> C[Compute Scalar Cost: 2 $]
        C --> D[Clustered Index Scan (Clustered) [Customer].[PK_Customer_CustomerID] Cost: 97 $]
    
```

## IN Memory:

In-Memory OLTP is a specialized, memory-optimized relational data management engine and native stored procedure compiler, integrated into SQL Server. Microsoft designed In-Memory OLTP to handle the most demanding OLTP workloads. In many cases, the memory-optimized tables can be created with DURABILITY = SCHEMA\_ONLY option to avoid all logging and I/O.

In-Memory OLTP introduces the following concepts:

- In-Memory optimized tables and indexes
- Non-durable tables, traditional temp tables
- Natively compiled stored procedures and UDF's
- Memory-optimized table type for table variable – This can be used as a replacement for temporary objects

```
USE [TicketReservations]
```

```
GO
```

```
DROP PROCEDURE IF EXISTS [dbo].[InsertReservationDetails]
```

```
GO
```

```
DROP TABLE IF EXISTS [dbo].[TicketReservationDetail]
```

```
GO
```

```
CREATE TABLE [dbo].[TicketReservationDetail]
```

```
(
```

```
    [TicketReservationID] [bigint] NOT NULL,  
    [TicketReservationDetailID] [bigint] IDENTITY(1,1) NOT NULL,  
    [Quantity] [int] NOT NULL,  
    [FlightID] [int] NOT NULL,  
    [Comment] [nvarchar](1000) NULL,
```

```
CONSTRAINT [PK_TicketReservationDetail] PRIMARY KEY NONCLUSTERED HASH
```

```
(
```

```
    [TicketReservationDetailID]  
) WITH (BUCKET_COUNT=10000000)  
)WITH ( MEMORY_OPTIMIZED = ON , DURABILITY = SCHEMA_AND_DATA )
```

```
GO
```

```
DROP PROCEDURE IF EXISTS [dbo].[InsertReservationDetails]  
GO  
CREATE PROCEDURE InsertReservationDetails(@TicketReservationID int, @LineCount int,  
@Comment NVARCHAR(1000), @FlightID int)  
WITH NATIVE_COMPILATION, SCHEMABINDING  
AS  
BEGIN ATOMIC WITH (TRANSACTION ISOLATION LEVEL=SNAPSHOT, LANGUAGE=N'English')  
DECLARE @loop int = 0;  
while (@loop < @LineCount)  
BEGIN  
    INSERT INTO dbo.TicketReservationDetail (TicketReservationID, Quantity, FlightID,  
Comment)  
    VALUES(@TicketReservationID, @loop % 8 + 1, @FlightID, @Comment);  
    SET @loop += 1;  
END  
END  
GO  
SET NOCOUNT ON;  
DECLARE @start DATETIME= GETDATE();  
DECLARE @id INT= 1;  
WHILE @id < 10000  
BEGIN  
    INSERT INTO InsertInMemDemo  
(id,  
data  
)  
VALUES
```

```

(@id,
'SQLShackDemo'
);
SET @id = @id + 1;

END;
SELECT DATEDIFF(s, @start, GETDATE()) AS [MemInsert];
GO

```

```
SELECT * FROM InsertInMemDemo
```

```

CONSTRANT [PK_TicketReservationDetail] PRIMARY KEY NONCLUSTERED HASH
(
    [TicketReservationDetailID]
) WITH (BUCKET_COUNT=10000000)
)WITH (MEMORY_OPTIMIZED = ON , DURABILITY = SCHEMA_AND_DATA )

GO

```

Messages  
Commands completed successfully.  
Completion time: 2021-02-05T12:15:57.4687172+05:30

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which displays the database structure of the 'TicketReservations' database, including tables like Sales.SalesTerritory, Sales.TerritoryHistory, and TicketReservationDetail. The main area contains a query window titled 'SQLQuery10.sql - VM...erations (sa (52))'. The code in the query window is:

```
(@id,
    'SQLShackDemo'
);
SET @id = @id + 1;
END;
SELECT DATEDIFF(s, @start, GETDATE()) AS [MemInsert];
GO
```

Below the code, there is a 'Results' tab showing the output of the 'SELECT \* FROM InsertInMemDemo' query. The results are:

	id	data
1	1	SQLShackDemo
2	2	SQLShackDemo
3	3	SQLShackDemo
4	4	SQLShackDemo
5	5	SQLShackDemo
6	6	SQLShackDemo
7	7	SQLShackDemo
8	8	SQLShackDemo
9	9	SQLShackDemo
10	10	SQLShackDemo
11	11	SQLShackDemo

A message at the bottom of the results tab says 'Query executed successfully.' The status bar at the bottom right shows 'VMT10213\TEST1 (15.0 RTM)'.

### Changing the durability from schema\_and\_data to schema\_only

```
USE [TicketReservations]
```

```
GO
```

```
DROP PROCEDURE IF EXISTS [dbo].[InsertReservationDetails]
```

```
GO
```

```
DROP TABLE IF EXISTS [dbo].[TicketReservationDetail]
```

```
GO
```

```
CREATE TABLE [dbo].[TicketReservationDetail_1]
```

```
(
```

```
[TicketReservationID] [bigint] NOT NULL,  
[TicketReservationDetailID] [bigint] IDENTITY(1,1) NOT NULL,  
[Quantity] [int] NOT NULL,  
[FlightID] [int] NOT NULL,
```

```

[Comment] [nvarchar](1000) NULL,
CONSTRAINT [PK_TicketReservationDetail_1] PRIMARY KEY NONCLUSTERED HASH
(
    [TicketReservationDetailID]
) WITH (BUCKET_COUNT=10000000)
)WITH ( MEMORY_OPTIMIZED = ON , DURABILITY = SCHEMA_ONLY )

GO

DROP PROCEDURE IF EXISTS [dbo].[InsertReservationDetails]
GO

CREATE PROCEDURE InsertReservationDetails_1(@TicketReservationID int, @LineCount int, @Comment NVARCHAR(1000), @FlightID int)
WITH NATIVE_COMPILATION, SCHEMABINDING
AS
BEGIN ATOMIC WITH (TRANSACTION ISOLATION LEVEL=SNAPSHOT, LANGUAGE=N'English')
DECLARE @loop int = 0;
while (@loop < @LineCount)
BEGIN
    INSERT INTO dbo.TicketReservationDetail_1 (TicketReservationID, Quantity, FlightID, Comment)
        VALUES(@TicketReservationID, @loop % 8 + 1, @FlightID, @Comment);
    SET @loop += 1;
END
END
GO

SET NOCOUNT ON;
CREATE TABLE InsertInMemDemo_1 (ID INT, DATA VARCHAR(20))

```

```
DECLARE @start DATETIME= GETDATE();
DECLARE @id INT= 1;
WHILE @id < 10000
BEGIN
    INSERT INTO InsertInMemDemo_1
    (id,
        data
    )
    VALUES
    (@id,
        'SQLShackDemo'
    );
    SET @id = @id + 1;
END;
SELECT DATEDIFF(s, @start, GETDATE()) AS [MemInsert];
GO
```

```
SELECT * FROM InsertInMemDemo_1
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer tree view is expanded to show the database structure, including the 'TicketReservations' database and its tables. In the center, the 'SQLQuery10.sql - VM\T10213\TEST1.TicketReservations (sa (52))' window displays the T-SQL code for creating a table:

```
CREATE TABLE [dbo].[TicketReservationDetail1]
(
    [TicketReservationID] [bigint] NOT NULL,
    [TicketReservationDetailID] [bigint] IDENTITY(1,1) NOT NULL,
    [Quantity] [int] NOT NULL,
    [FlightID] [int] NOT NULL,
    [Comment] [nvarchar](1000) NULL,
)
```

Below the code, the 'Messages' pane shows the output: "Commands completed successfully." and the completion time: "Completion time: 2021-02-05T12:36:59.2205044+05:30".

## EXPORTING AND IMPORTING DATA BETWEEN SSMS AND AZURE:

### EXPORTING DATA:

Create a new database and table is created.

The screenshot shows the SQL Server Management Studio (SSMS) interface. The Object Explorer on the left shows the database structure for 'vmsn1test' (Windows.net | SQL Server (2008)) including databases, tables, and security. The central pane displays a query window with the following T-SQL code:

```
--check inserting rows and verify identity is proper or not
create table department
(emoid int identity(1,1) primary key,deptname varchar(10), emename varchar(10), phone varchar(10), zip varchar(10))

Insert into department
values ('sales','shivani',988000564,500079)

select * from department

Insert into department
values ('support','supriya',988776744,500592)
Insert into department
values ('ict','priya',898776744,98592)
```

The results pane at the bottom shows the data inserted into the 'department' table:

	emoid	deptname	emename	phone	zip
1	1	sales	shivani	988000564	500079
2	2	support	supriya	988776744	500592
3	3	ict	priya	898776744	98592

**Storage account is created**

Home > Storage accounts >

## Create storage account

Subscription \*

Resource group \*  [Create new](#)

**Instance details**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name \*

Location \*

Performance  Standard  Premium

Account kind

Replication

[Review + create](#)

< Previous

Next : Networking >

Home >

## Microsoft.StorageAccount-20210208112949 | Overview [✖](#)

[Deployment](#)

[Search \(Ctrl+ /\)](#) << [Delete](#) [Cancel](#) [Redeploy](#) [Refresh](#)

[We'd love your feedback! →](#)

 Your deployment is complete

 Deployment name: Microsoft.StorageAccount-20210208112949 Start time: 2/8/2021, 11:32:39 AM  
Subscription: Free Trial Correlation ID: d5e259b8-270c-4412-a699-52917b5c9b25  
Resource group: vmtestrg

 Deployment details [\(Download\)](#)

 Next steps

[Go to resource](#)



**Security Center**  
Secure your apps and infrastructure  
[Go to Azure security center >](#)

**Free Microsoft tutorials**  
[Connect an app to Azure Storage](#)  
[Store application data with Azure Storage](#)  
[Secure your Azure Storage Account](#)  
[Monitor, diagnose, and troubleshoot Azure Storage](#)

## vmsqlbackups | Storage Explorer (preview)

Storage account

 Search (Ctrl+ /)

«  Search

### BLOB CONTAINERS

-  FILE SHARES
-  QUEUES
-  TABLES

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data transfer

Events

Storage Explorer (preview)

Settings

Access keys

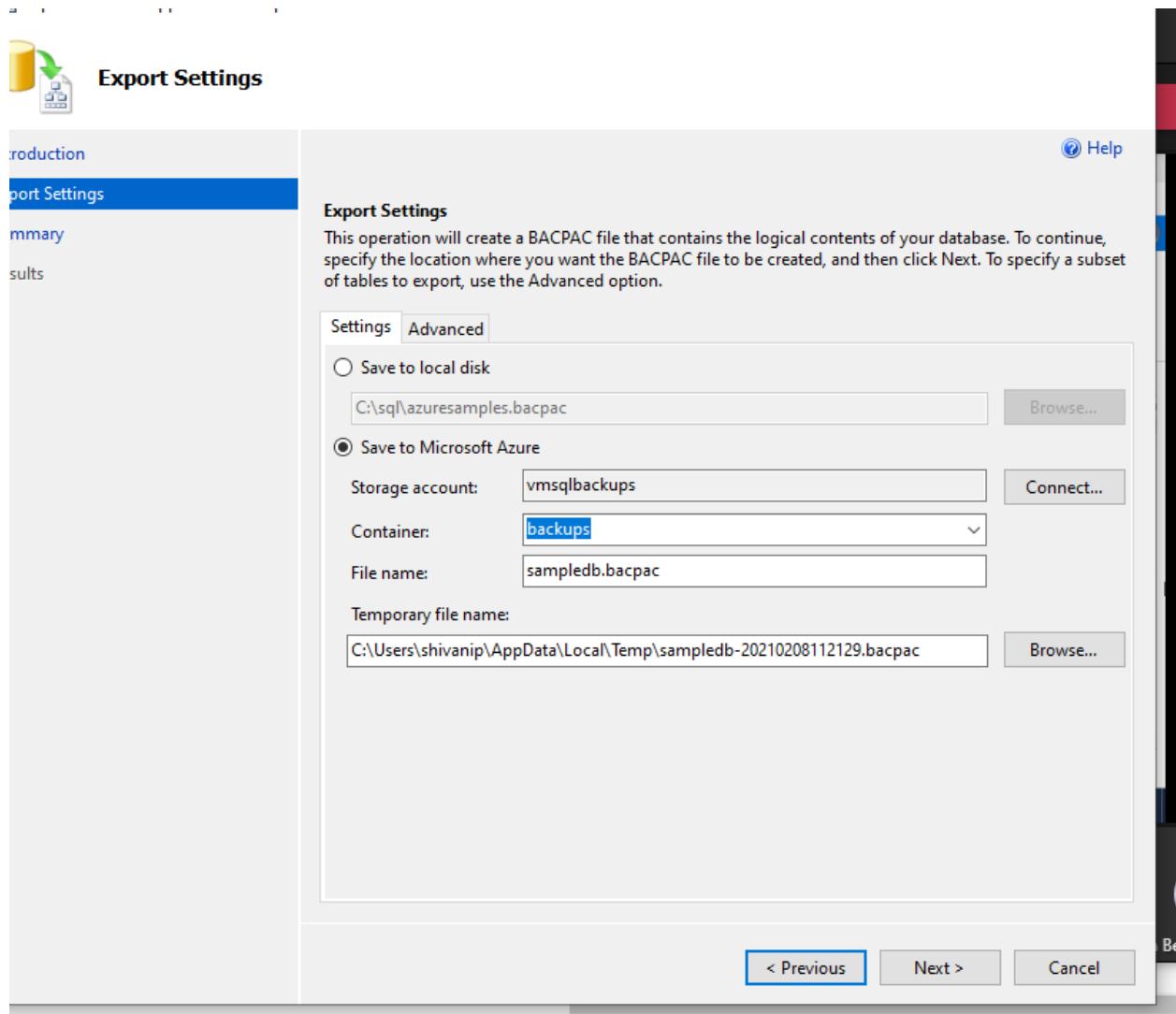
Geo-replication

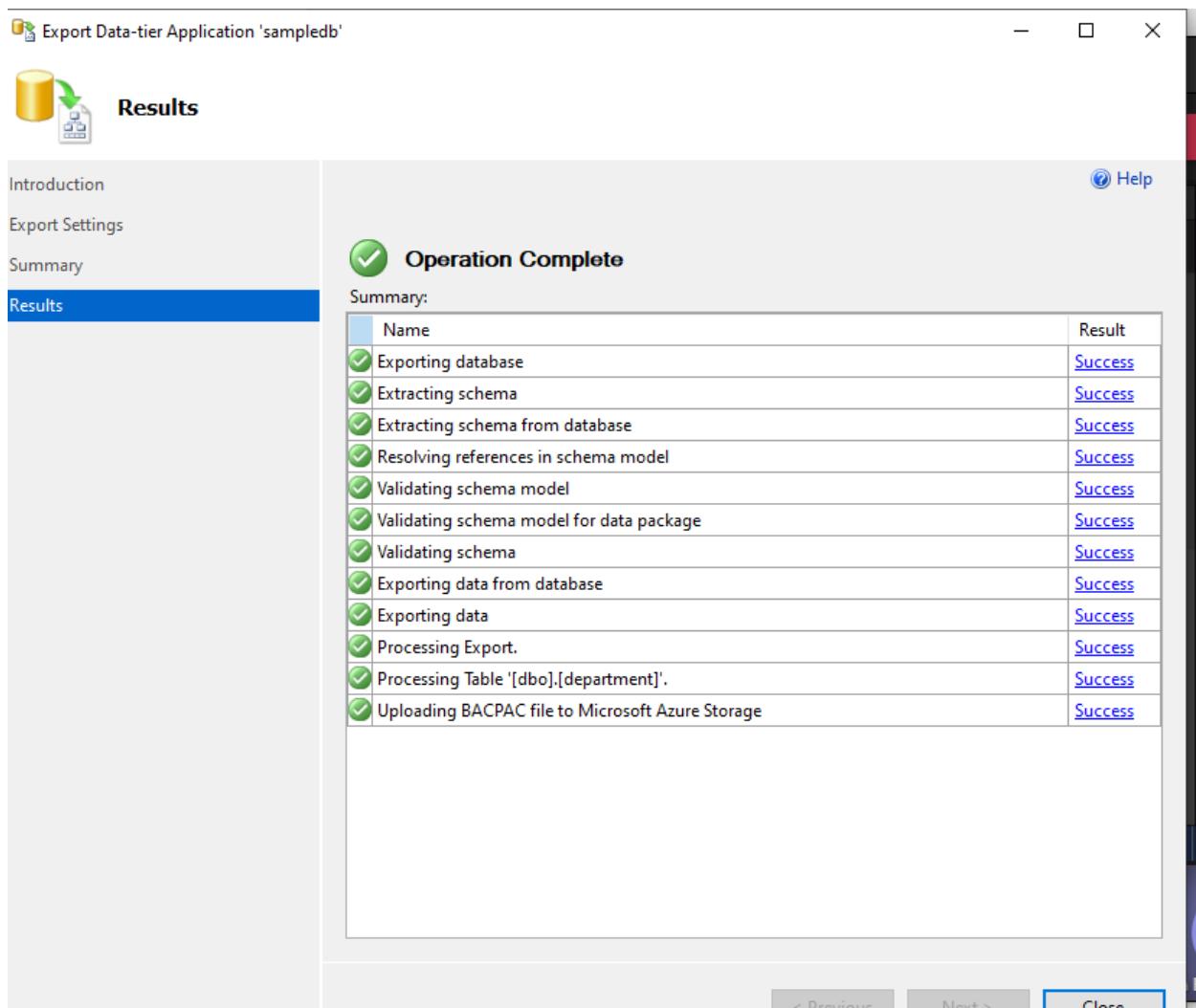
CORS

Configuration

-

## Exporting data

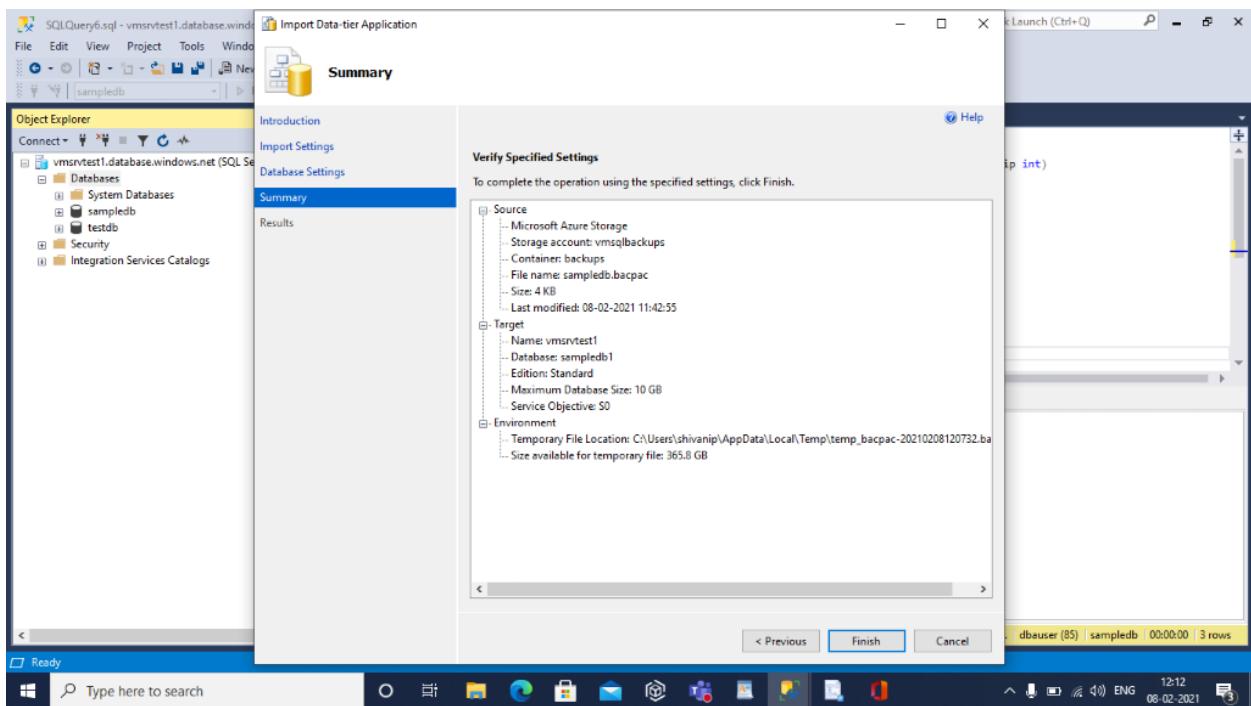


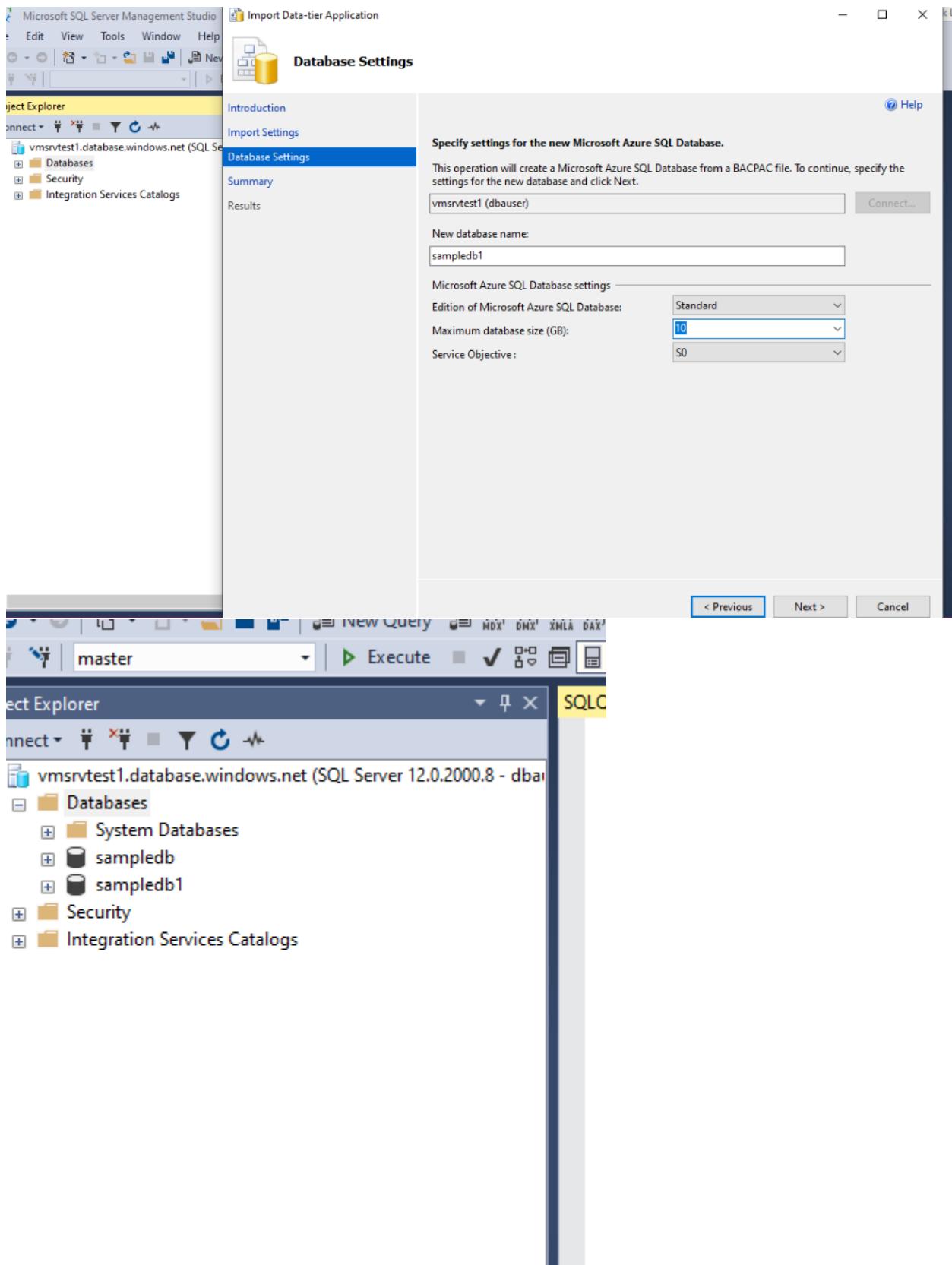


The screenshot shows the 'Storage Explorer (preview)' interface for an Azure storage account named 'rmysqlbackups'. The left sidebar lists storage resources like 'Storage account', 'Container', 'File share', 'Queue', and 'Table'. The main pane shows a 'BLOB CONTAINERS' view with a 'backups' container selected. The 'backups' container contains two blobs: 'sampledb.bacpac' and 'shivani\_vam 5221\_day10\_4FEB.docx'. The table below provides detailed information about these blobs:

NAME	ACCESS TIER	ACCESS TIER LAST MODIFIED	LAST MODIFIED	BLOB TYPE	CONTENT TYPE
sampledb.bacpac	Hot (inferred)	2/8/2021, 11:42:55 AM	2/8/2021, 11:42:55 AM	Block Blob	application/octet-stream
shivani_vam 5221_day10_4FEB.docx	Hot (inferred)	2/8/2021, 11:37:27 AM	2/8/2021, 11:37:27 AM	Block Blob	application/octet-stream

**Import data:**





## Elastic pools:

Azure SQL Database elastic pools are a simple, cost-effective solution for managing and scaling multiple databases that have varying and unpredictable usage demands. The databases in an elastic pool are on a single server and share a set number of resources at a set price. Elastic pools in Azure SQL Database enable SaaS developers to optimize the price performance for a group of databases within a prescribed budget while delivering performance elasticity for each database.

SaaS developers build applications on top of large scale data-tiers consisting of multiple databases. A common application pattern is to provision a single database for each customer. But different customers often have varying and unpredictable usage patterns, and it's difficult to predict the resource requirements of each individual database user. Traditionally, you had two options:

- Over-provision resources based on peak usage and over pay, or
- Under-provision to save cost, at the expense of performance and customer satisfaction during peaks.

Elastic pools solve this problem by ensuring that databases get the performance resources they need when they need it. They provide a simple resource allocation mechanism within a predictable budget.

- Go to the Azure portal to create an elastic pool. Search for and select **Azure SQL**.
- Select **+Add** to open the **Select SQL deployment option** page. You can view additional information about elastic pools by selecting **Show details** on the **Databases** tile.
- On the **Databases** tile, select **Elastic pool** in the **Resource type** dropdown, then select **Create**:
- 

The screenshot shows the 'Select SQL deployment option' page in the Azure portal. At the top, there is a breadcrumb navigation: Home > Azure SQL >. Below this, the title 'Select SQL deployment option' is displayed. There is a 'Feedback' link and a 'Feedback' button. The main content area is titled 'How do you plan to use the service?'. It contains three cards:

- SQL databases**: Best for modern cloud applications. Hyperscale and serverless options are available. Resource type dropdown is set to 'Elastic pool'. Buttons: Create, Show details.
- SQL managed instances**: Best for most migrations to the cloud. Lift-and-shift ready. Resource type dropdown is set to 'Single instance'. Buttons: Create, Show details.
- SQL virtual machines**: Best for migrations and applications requiring OS-level access. Lift-and-shift ready. Resource type dropdown is set to 'None'. Buttons: Create, Show details.

Or you can create an elastic pool by navigating to an existing server and clicking **+ New pool** to create a pool directly into that server.

The screenshot shows the Microsoft Azure portal interface for creating a new SQL elastic pool. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main page title is 'elasticpool1 (vmsrvtest1/elasticpool1)'. The left sidebar lists 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings' (selected), 'Quick start', 'Configure', 'Locks', 'Monitoring' (selected), 'Database Resource Utilization', 'Alerts', and 'Metrics'. The right pane shows the 'Essentials' section with details: Resource group (change) : vmtestrg, Status : Ready, Location : East US 2, Subscription (change) : Free Trial, Subscription ID : 87c70a0b-ed61-4866-a52b-651e64ab1598, and Tags (change) : created by : shivani. It also shows resource utilization graphs for the elastic pool.

- Create elastic pool, move these two databases in same elastic pool.

Deploy databases test1 ,test2 in azure.

Create elasticpool in azure.

The screenshot shows the Microsoft Azure portal interface for creating a new SQL elastic pool. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main page title is 'vmtest1 (vmazuretest/vmtest1)'. The left sidebar lists 'Add' (selected), 'Reservations', '...'. Below it is a 'Filter for any field...' input field. A list item 'vmtest1 (vmazuretest/vmtest1)' is shown with three dots next to it. The right pane shows the 'Essentials' section with details: Resource group (change) : vmazuretest, Status : Ready, Location : East US 2, Subscription (change) : Free Trial, Subscription ID : 87c70a0b-ed61-4866-a52b-651e64ab1598, and Tags (change) : Click here to add tags. It also shows resource utilization graphs for the elastic pool.

click on add databases

SQL elastic pools

**vmtest1 (vmazuretest/vmtest1) | Configure**

SQL elastic pool

Search (Ctrl+ /)

Save Cancel Feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings Quick start Configure Locks Monitoring Database Resource Utilization Alerts Metrics

Add databases Remove from pool Revert selected

Databases to be removed from pool

Search to filter databases...

Database name Pricing tier

Currently, there are no databases selected to be removed from this pool. To remove databases, select them and click 'Remove from pool'.

Ready to be added to this pool

Search to filter databases...

Database name Pricing tier

Selected/Total database 2/2

test1,test2 databases are added in elastic pools.

SQL elastic pools

**vmtest1 (vmazuretest/vmtest1)**

SQL elastic pool

Search (Ctrl+ /)

Configure Delete Create database Feedback

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings Quick start Configure Locks Monitoring Database Resource Utilization Alerts Metrics

Resource group (change)  
newgrp

Status Ready

Location East US 2

Subscription (change)  
Free Trial

Subscription ID  
87c70a0b-ed61-4866-a52b-651e64ab1598

Tags (change)  
Click here to add tags

Show data for last: 1 hour 24 hours 7 days

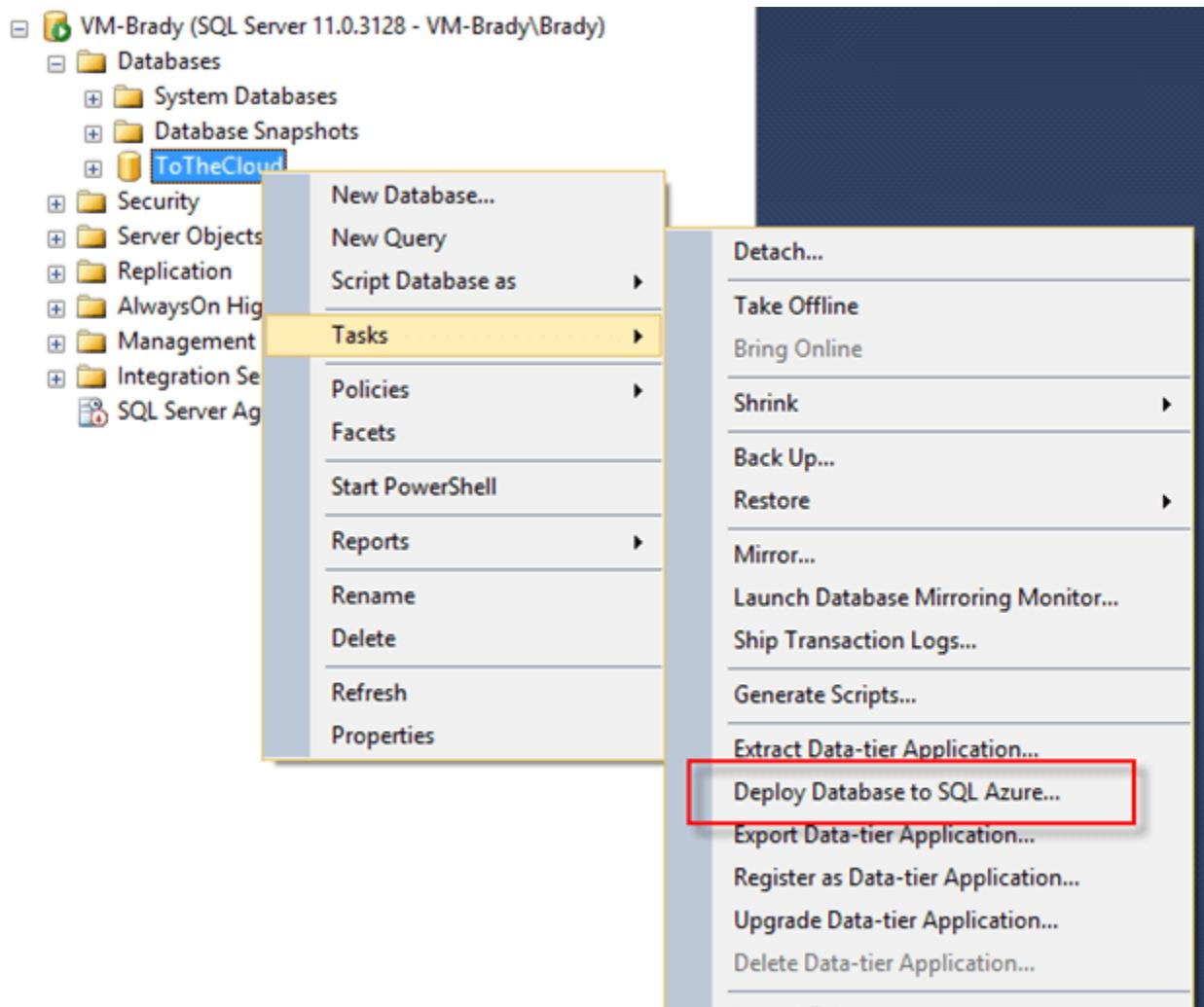
Resource utilization (vmtest1)

100%

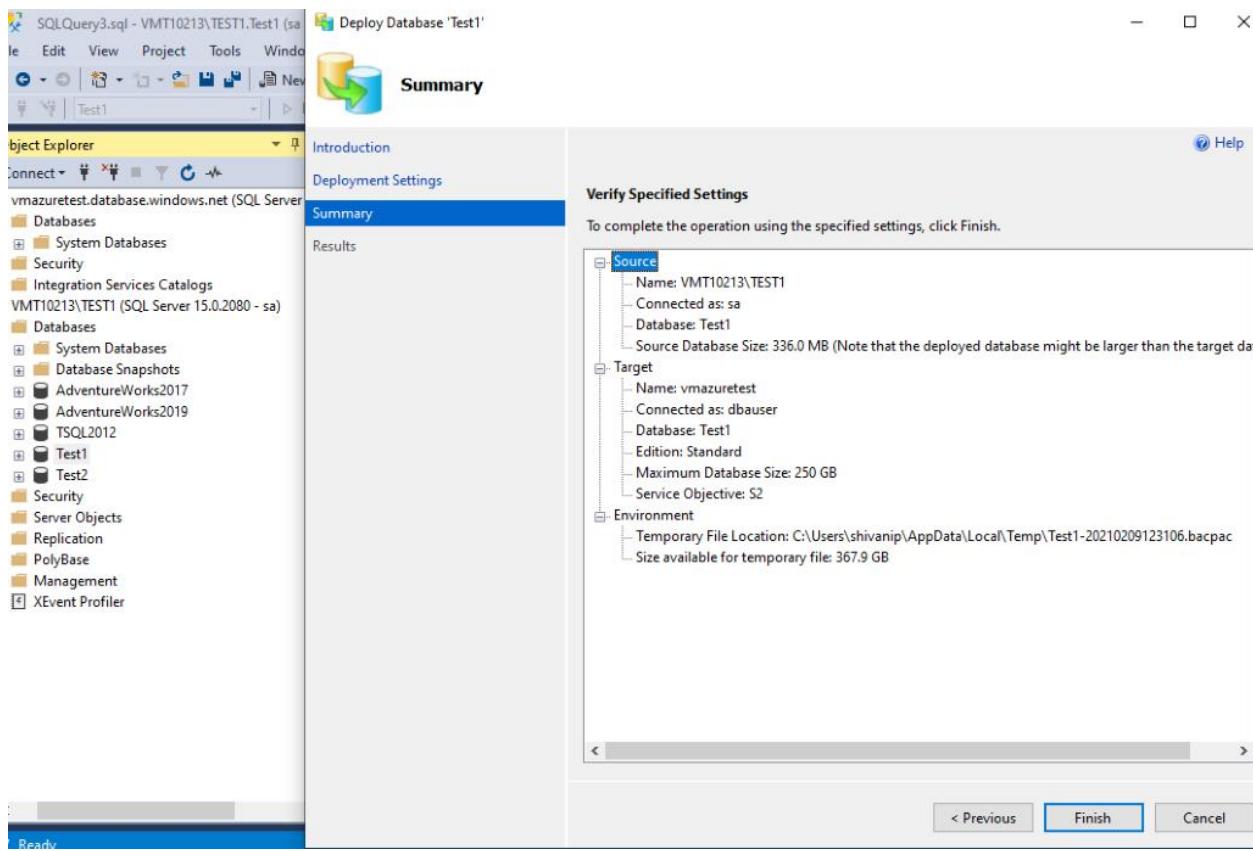
## Deploy Database to SQL Azure:

In SQL Server 2012, there is a new wizard that will allow you to move an on premise database to SQL Azure. This is only available in SQL Server 2012 since SQL Azure was nonexistent in previous versions.

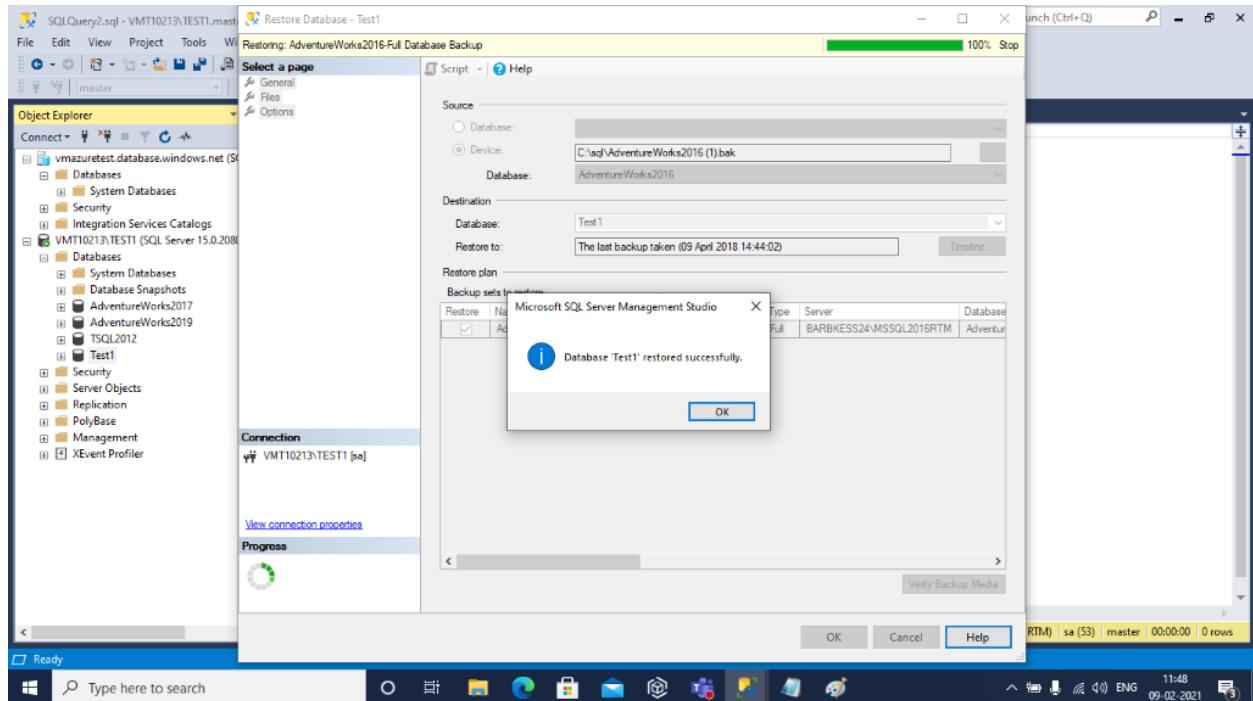
To use this wizard, open SQL Server Management Studio (SSMS) and connect to the server. Right click the database you want to move, click Tasks, Deploy Database to SQL Azure..



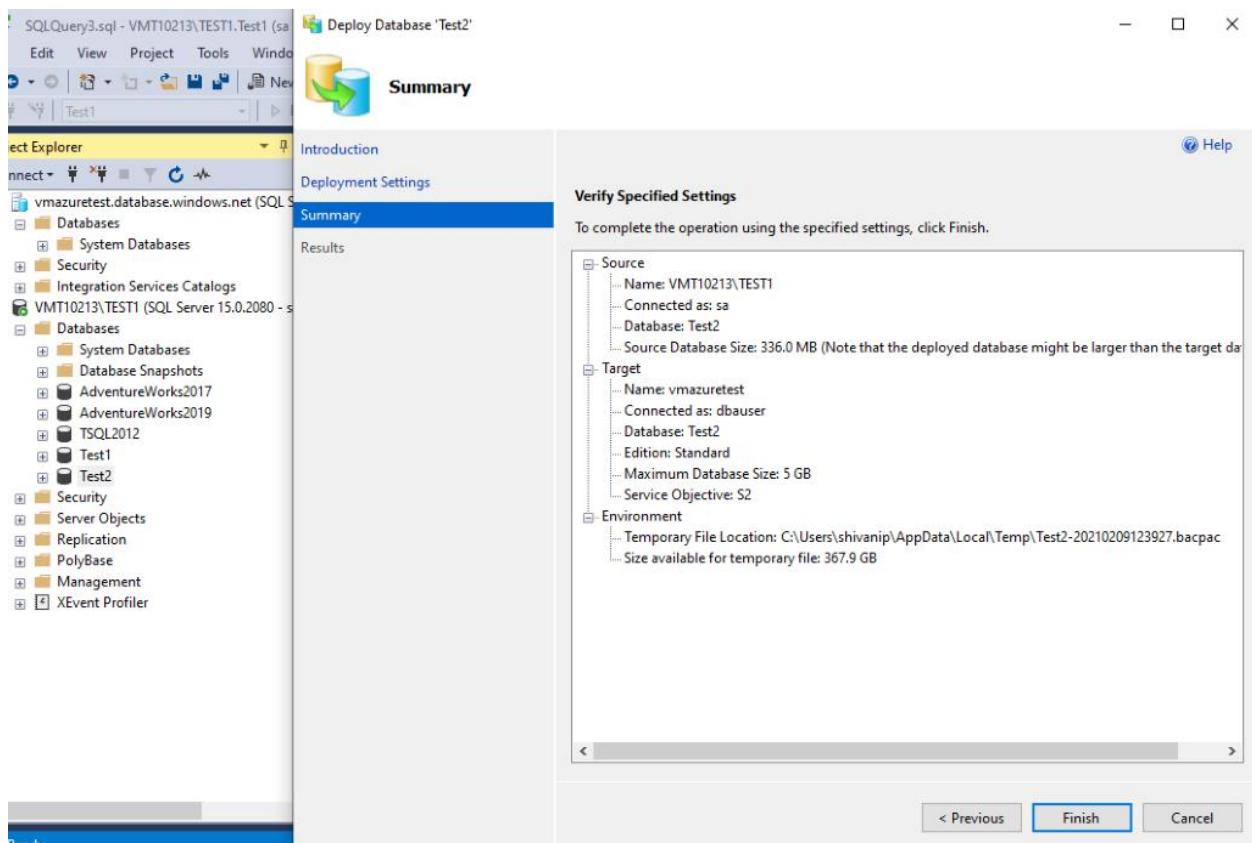
Deploying database test1



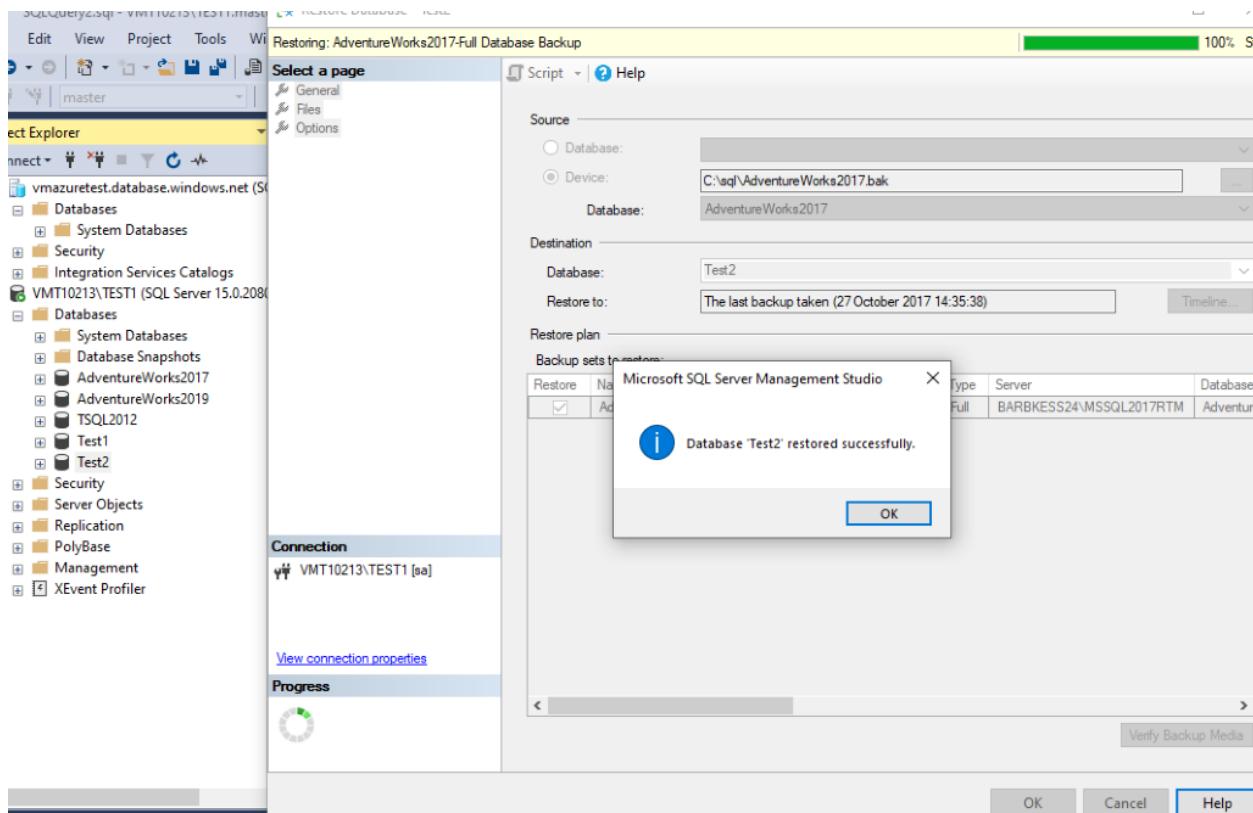
test1 is restored



Deploying database test2



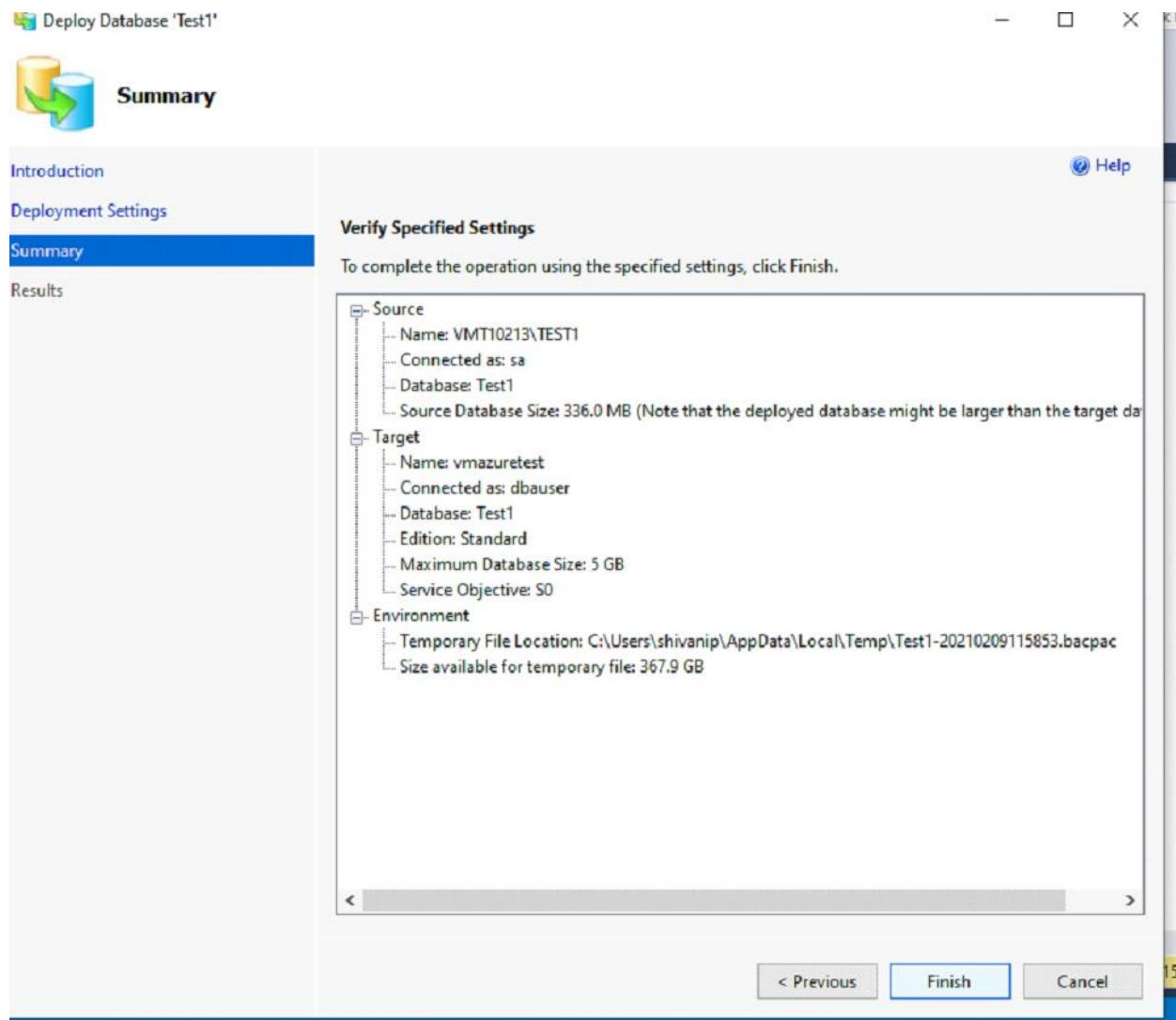
Test2 is restored



Databases Test1,Test2 are deployed in azure.

Database	Status	Pricing tier
Test1	Online	Standard S2: 50 DTUs
Test2	Online	Standard S2: 50 DTUs

Move these databases to Azure SQL server using deploy option from SSMS.



 Results

Auction  
Payment Settings  
Currency  
Tasks

Help

 **Operation Complete**

Summary:

Name	Result
Enabling index 'AK_SalesOrderHeader_rowguid'.	Success
Enabling index 'AK_SalesOrderHeader_SalesOrderNumber'.	Success
Enabling index 'IX_SalesOrderHeader_CustomerID'.	Success
Enabling index 'IX_SalesOrderHeader_SalesPersonID'.	Success
Enabling index 'AK_SalesPerson_rowguid'.	Success
Enabling index 'AK_SalesPersonQuotaHistory_rowguid'.	Success
Enabling index 'AK_SalesTaxRate_rowguid'.	Success
Enabling index 'AK_SalesTaxRate_StateProvinceID_TaxType'.	Success
Enabling index 'AK_SalesTerritory_Name'.	Success
Enabling index 'AK_SalesTerritory_rowguid'.	Success
Enabling index 'AK_SalesTerritoryHistory_rowguid'.	Success
Enabling index 'IX_ShoppingCartItem_ShoppingCartID_ProductID'.	Success
Enabling index 'AK_SpecialOffer_rowguid'.	Success
Enabling index 'AK_SpecialOfferProduct_rowguid'.	Success
Enabling index 'IX_SpecialOfferProduct_ProductID'.	Success
Enabling index 'AK_Store_rowguid'.	Success
Enabling index 'IX_Store_SalesPersonID'.	Success
Refreshing Object Explorer	Success

< Previous Next > Close



| (vmazuretest/Test1) | Query editor (preview)

base

+/- Login New Query Open query Feedback

Tables

- > dbo.AWBuildVersion
- > dbo.DatabaseLog
- > dbo.ErrorLog
- > HumanResources.Department
- > HumanResources.Employee
- > HumanResources.EmployeeDepartmentHistory
- > HumanResources.EmployeePayHistory
- > HumanResources.JobCandidate
- > HumanResources.Shift
- > Person.Address
- > Person.AddressType
- > Person.BusinessEntity
- > Person.BusinessEntityAddress
- > Person.BusinessEntityContact
- > Person.ContactType

Query 1 × Query 2 × Query 3 ×

Run Cancel query Save query Export data as Show only Editor

```
1 SELECT TOP (1000) * FROM [HumanResources].[Department]
```

Results Messages

1	Engineering	Research and Development	2008-04-30T00:00:00.0000
2	Tool Design	Research and Development	2008-04-30T00:00:00.0000
3	Sales	Sales and Marketing	2008-04-30T00:00:00.0000

Query succeeded | 1s

Works201...bak ... AdventureWorks201...bak ... Open file Show all X

The screenshot shows the Azure Data Studio interface with the title bar '(vmazuretest/Test1) | Query editor (preview)'. The left sidebar displays a tree view of database objects under 'Tables', including tables like dbo.AWBuildVersion, dbo.DatabaseLog, dbo.ErrorLog, HumanResources.Department, HumanResources.Employee, etc. The main area shows three tabs: 'Query 1', 'Query 2', and 'Query 3', with 'Query 3' selected. The query editor contains the following SQL code:

```
1 SELECT TOP (1000) * FROM [HumanResources].[Department]
```

The results tab shows the output of the query:

1	Engineering	Research and Development	2008-04-30T00:00:00.0000
2	Tool Design	Research and Development	2008-04-30T00:00:00.0000
3	Sales	Sales and Marketing	2008-04-30T00:00:00.0000

A message at the bottom indicates 'Query succeeded | 1s'.

## Deployment Settings

Specify Target Connection

Specify the name of the instance of SQL Server or the Microsoft Azure SQL Database server that will host the deployed database, name the new database, and then click Connect to login to the target server.

Server connection:

vmaurtest (dbauer)

New database name:

Test2

Microsoft Azure SQL Database settings

Edition of Microsoft Azure SQL Database:

Standard

Maximum database size (GB):

5

Service Objective :

S2

Other settings

Temporary file name:

C:\Users\shivanip\AppData\Local\Temp\Test2-20210209123927.bacpac

Browse...

< Previous

Next >

Cancel

## Results

on

ant Settings

Help



### Operation Complete

Summary:

Name	Result
Exporting database	Success
Extracting schema	Success
Extracting schema from database	Success
Resolving references in schema model	Success
Validating schema model	Success
Validating schema model for data package	Success
Validating schema	Success
Exporting data from database	Success
Exporting data	Success
Processing Export.	Success
Processing Table '[Sales].[SalesTaxRate]'	Success
Processing Table '[Sales].[PersonCreditCard]'	Success
Processing Table '[Person].[PersonPhone]'	Success
Processing Table '[Sales].[SalesTerritory]'	Success
Processing Table '[Person].[PhoneNumberType]'	Success
Processing Table '[Production].[Product]'	Success
Processing Table '[Sales].[SalesTerritoryHistory]'	Success
Processing Table '[Production].[ScrapReason]'	Success
Processing Table '[HumanResources].[Shift]'	Success

< Previous

Next >

**Close**

est2 (vmazuretest/Test2) | Query editor (preview)

X

.database

View

ty log

lose and solve problems

:start

editor (preview)

form

BI (preview)

Apps (preview)

Automate (preview)

gure

lization

Test2 (dbuser)

Showing limited object explorer here.  
For full capability please open SSDT.

Tables

dbo\_TransactionIndex\_dc09b17...

72d9-4a0f-a922-9e9ac084e4de

dbo\_AWBuildVersion

...

dbo\_DatabaseLog

...

dbo\_ErrorLog

...

HumanResources.Department

...

HumanResources.Employee

...

HumanResources.EmployeeDep...

...

HumanResources.EmployeePay...

...

HumanResources.JobCandidate

...

Query 1 X Query 2 X

Run

Cancel query

Save query

Export data as

Show only Editor

1 SELECT TOP (1000) \* FROM [HumanResources].[EmployeeDepartmentHistory]

Results Messages

BusinessEntityID	DepartmentID	ShiftID	StartDate	EndDate
1	16	1	2009-01-14T00:00:00...	
2	1	1	2008-01-31T00:00:00...	
3	1	1	2007-11-11T00:00:00...	

Query succeeded | 0s

## **Automated Backup for SQL Server:**

### **Database configuration:**

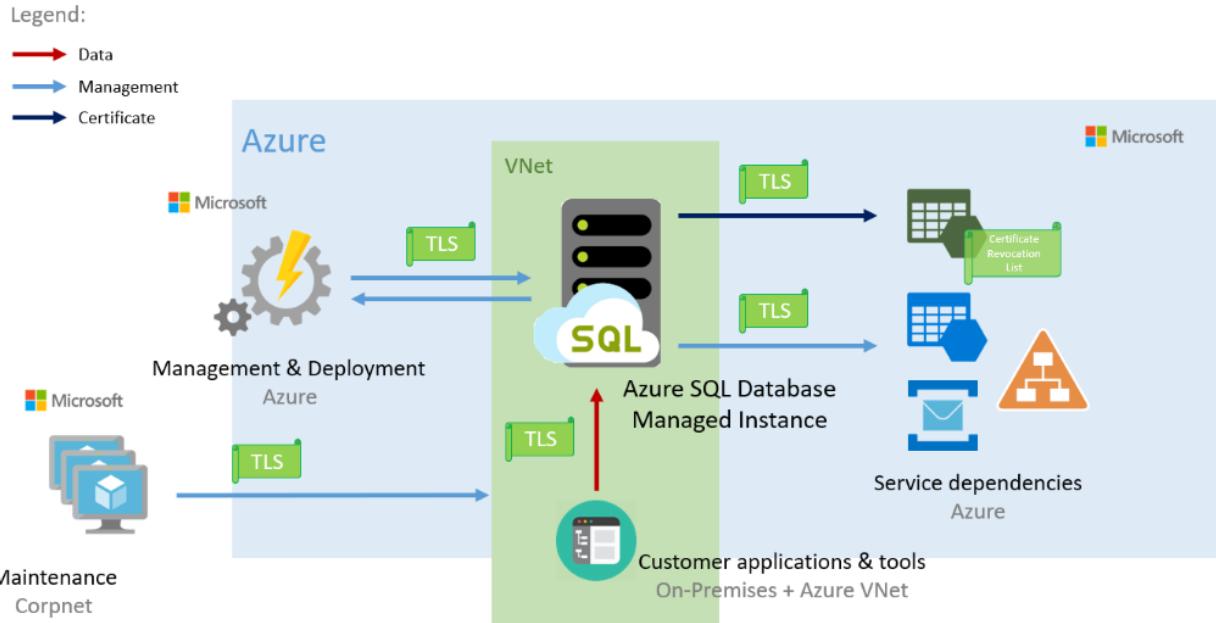
- Target *user* databases must use the full recovery model. System databases do not have to use the full recovery model. However, if you require log backups to be taken for Model or MSDB, you must use the full recovery model. For more information about the impact of the full recovery model on backups, see Backup under the full recovery model
- The SQL Server VM has been registered with the SQL IaaS Agent extension in full management mode
- Automated backup relies on the full SQL Server IaaS Agent Extension. As such, automated backup is only supported on target databases from the default instance, or a single named instance. If there is no default instance, and multiple named instances, the SQL IaaS extension fails and automated backup will not work.

### **Managed Instance:**

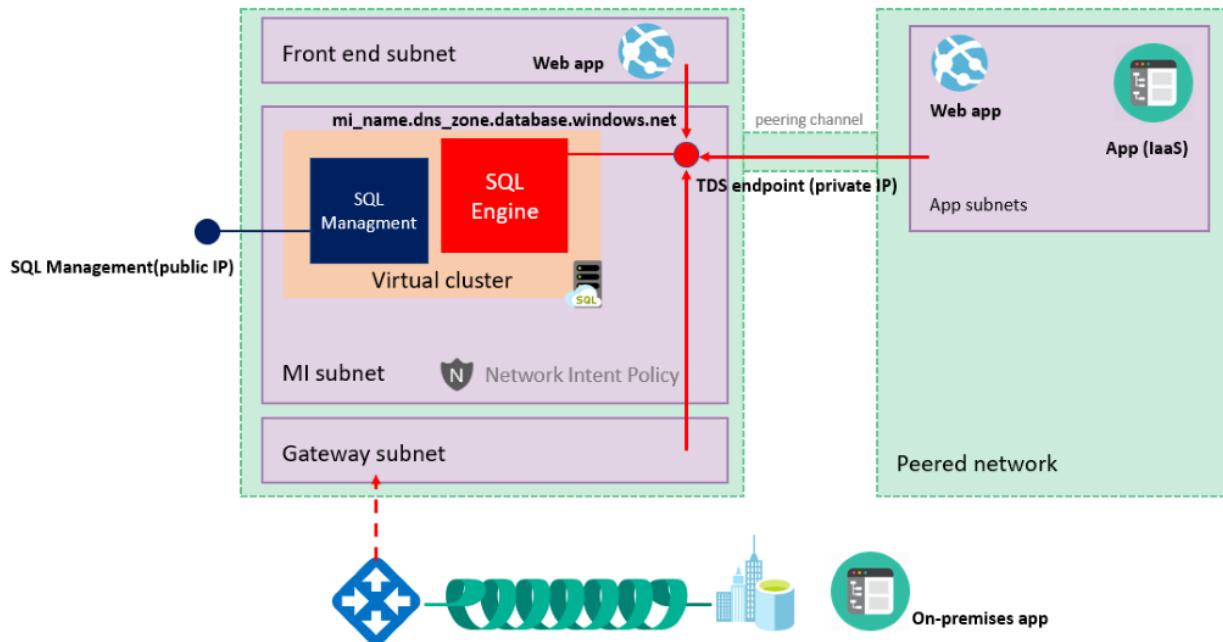
SQL Managed Instance is placed inside the Azure virtual network and the subnet that's dedicated to managed instances. This deployment provides:

- A secure private IP address.
- The ability to connect an on-premises network to SQL Managed Instance.
- The ability to connect SQL Managed Instance to a linked server or another on-premises data store.
- The ability to connect SQL Managed Instance to Azure resources.

SQL Managed Instance is a platform as a service (PaaS) offering. Azure uses automated agents (management, deployment, and maintenance) to manage this service based on telemetry data streams. Because Azure is responsible for management, customers can't access the SQL Managed Instance virtual cluster machines through Remote Desktop Protocol (RDP).



We create managed instance to reduce the cost of the VM which is in azure ,because of that instance there is no need of multiple instances and replicates are not needed.



### Elastic Query:

The elastic query feature (in preview) enables you to run a Transact-SQL query that spans multiple databases in Azure SQL Database. It allows you to perform cross-database queries to access remote tables, and to connect Microsoft and third-party tools (Excel, Power BI, Tableau, etc.) to query across data tiers with multiple databases.

Using this feature, you can scale out queries to large data tiers and visualize the results in business intelligence (BI) reports.

Configure elastic query to access TEST4 DB table into Test3 database.

```
create user shivanipalle for login shivanipalle
```

```
go
```

```
sp_addrolemember 'db_owner','shivanipalle'
```

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Pass@12345!';
```

```
-- drop DATABASE SCOPED CREDENTIAL DbCredential
```

```
CREATE DATABASE SCOPED CREDENTIAL DbCredential
```

```
WITH IDENTITY = 'shivanipalle',
```

```
SECRET = 'abcd@1234';
```

```
CREATE EXTERNAL DATA SOURCE RemoteDBSource
```

```
WITH
```

```
(
```

```
TYPE=RDBMS,
```

```
LOCATION='dbauser.database.windows.net',
```

```
DATABASE_NAME='TEST4',
```

```
CREDENTIAL= DbCredential
```

```
);
```

```
CREATE external TABLE department1(
    empid int ,deptname varchar(10), empname varchar(50),phone decimal,zip int
)
WITH
(
    DATA_SOURCE = RemoteDBSource
    ,SCHEMA_NAME = 'dbo'           -- external table schema
    OBJECT_NAME = 'department1'
);
select *from department1
```

Object Explorer

dbauer.database.windows.net (SQL Server 12.0.2000.8 - dba)

- Databases
  - System Databases
  - db1
  - Test3
    - Database Diagrams
    - Tables
      - System Tables
      - External Tables
      - GraphTables
    - Views
    - External Resources
    - Synonyms
    - Programmability
    - Query Store
    - Extended Events
    - Storage
    - Security
  - Test4
    - Database Diagrams
    - Tables
      - System Tables
      - External Tables
      - GraphTables
      - dbo.department
    - Views

SQLQuery1.sql - dba...ster (dbauser (70)) ~vsCC23.sql - dbaus...est3 (dbauser (93))

```

create user shivanipalle for login shivanipalle
go
sp_addrolemember 'db_owner', 'shivanipalle'

create table department1
(empid int ,deptname varchar(10), empname varchar(50),phone decimal,zip int)

insert into department1
values (1,'sales','shivani',900000564,500079)

select * from department1

insert into department
values (2,'support','supriya',9988776744,500592)
insert into department
values (3,'ict','nriya',890776744,90592)

```

Results

empid	deptname	empname	phone	zip
1	sales	shivani	900000564	500079

Object Explorer

dbauer.database.windows.net (SQL Server 12.0.2000.8 - dba)

- Databases
  - System Databases
  - db1
  - Test3
    - Database Diagrams
    - Tables
      - System Tables
      - External Tables
      - GraphTables
    - Views
    - External Resources
    - Synonyms
    - Programmability
    - Query Store
    - Extended Events
    - Storage
    - Security
  - Test4
    - Database Diagrams
    - Tables
      - System Tables
      - External Tables
      - GraphTables
      - dbo.department
    - Views

SQLQuery1.sql - dba...ster (dbauser (70)) ~vsCC23.sql - dbaus...est3 (dbauser (93))

```

create user shivanipalle for login shivanipalle
go
sp_addrolemember 'db_owner', 'shivanipalle'

create table department1
(empid int ,deptname varchar(10), empname varchar(50),phone decimal,zip int)

insert into department1
values (1,'sales','shivani',900000564,500079)

select * from department1

insert into department
values (2,'support','supriya',9988776744,500592)
insert into department
values (3,'ict','nriya',890776744,90592)

```

Results

empid	deptname	empname	phone	zip
1	sales	shivani	900000564	500079

```

CREATE EXTERNAL TABLE department1(
    empid int ,deptname varchar(10), empname varchar(50),phone decimal,zip int
)
WITH
(
    DATA_SOURCE = RemoteDBSource
    ,SCHEMA_NAME = 'dbo'          -- external table schema
    ,OBJECT_NAME = 'department1'
);
select *from department1

```

	empid	deptname	empname	phone	zip
1	1	sales	shivani	900000564	500079

### Cross query:

Starting with SQL Server 2014 (12.x), memory-optimized tables do not support cross-database transactions. You cannot access another database from the same transaction or the same query that also accesses a memory-optimized table. You cannot easily copy data from a table in one database, to a memory-optimized table in another database.

Table variables are not transactional. Therefore, memory-optimized table variables can be used in cross-database queries, and can thus facilitate moving data from one database into memory-optimized tables in another. You can use two transactions. In the first transaction, insert the data from the remote table into the variable. In the second transaction, insert the data into the local memory-optimized table from the variable.

### Azure SQL VM:

SQL Server on Azure Virtual Machines enables you to use full versions of SQL Server in the cloud without having to manage any on-premises hardware. SQL Server virtual machines (VMs) also simplify licensing costs when you pay as you go.

Azure virtual machines run in many different geographic regions around the world. They also offer a variety of machine sizes. The virtual machine image gallery allows you to create a SQL Server VM with the right version, edition, and operating system. This makes virtual machines a good option for many different SQL Server workloads.

Q: Create Azure VM with SQL server 2016 developer edition. Add 2 required disk with lower configuration to save cost

## Create a sql virtual machine

The screenshot shows the Microsoft Azure portal interface for creating a SQL virtual machine. The top navigation bar includes 'Microsoft Azure', 'Upgrade', a search bar ('Search resources, services, and docs (G+)'), and user information ('shivanireddy2905@gmail.com'). Below the navigation, the breadcrumb path shows 'Home > SQL virtual machines >'. A title 'Select SQL deployment option' is displayed above three cards:

- SQL databases**: Best for modern cloud applications. Hyperscale and serverless options are available. Resource type: Single database. Buttons: Create, Show details.
- SQL managed instances**: Best for most migrations to the cloud. Lift-and-shift ready. Resource type: Single instance. Buttons: Create, Show details.
- SQL virtual machines**: Best for migrations and applications requiring OS-level access. Lift-and-shift ready. License: Free License: SQL Server 2016 SP1 Develop... (button highlighted with a red box). Buttons: Create, Hide details.

A modal window titled 'SQL virtual machine' provides detailed information about the service and its capabilities:

**SQL virtual machine**  
SQL virtual machines offer full administrative control over the SQL Server instance and underlying OS for migration to Azure.

**Featured capabilities:**

- ✓ SQL Server and OS access
- ✓ Expansive SQL Server and OS version support
- ✓ Automated manageability features for SQL

## Create a virtual machine

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Free Trial

Resource group \* ⓘ

(New) newgrp

[Create new](#)

### Instance details

Virtual machine name \* ⓘ

sqlvm

Region \* ⓘ

(US) East US 2

Availability options ⓘ

No infrastructure redundancy required

Image \* ⓘ

Free License: SQL Server 2016 SP1 Developer on Windows Server 2016 - GenV

[See all images](#)

Azure Spot instance ⓘ

[Review + create](#)

[< Previous](#)

[Next : Disks >](#)

## Create a virtual machine

Size \* ⓘ

Standard\_B2s - 2 vcpus, 4 GiB memory (₹2,608.61/month)

[See all sizes](#)

### Administrator account

Username \* ⓘ

dbouser



Password \* ⓘ

\*\*\*\*\*



Confirm password \* ⓘ

\*\*\*\*\*



### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ

None

Allow selected ports

Select inbound ports \*

HTTP (80), HTTPS (443), RDP (3389)

[Review + create](#)

< Previous

[Next : Disks >](#)

## Create a virtual machine

Basics Disks Networking Management Advanced SQL Server settings Tags Review + create

Configure monitoring and management options for your VM.

### Azure Security Center

Azure Security Center provides unified security management and advanced threat protection across hybrid cloud workloads.  
[Learn more](#)

Enable basic plan for free (i)



This will apply to every VM in the selected subscription

### Monitoring

Boot diagnostics (i)



Enable with managed storage account (recommended)



Enable with custom storage account



Disable

Enable OS guest diagnostics (i)



### Identity

**Review + create**

< Previous

Next : Advanced >



## Create a virtual machine

Enable OS guest diagnostics

### Identity

System assigned managed identity

### Auto-shutdown

Enable auto-shutdown

Shutdown time

Time zone

Notification before shutdown

Email \*

### Guest OS updates

Patch installation

Azure-orchestrated patching (preview): patches will be installed by Azure

OS-orchestrated patching: patches will be installed by OS

**Review + create**

< Previous

Next : Advanced >

## Create a virtual machine

### Security & Networking

SQL connectivity \*

Private (within Virtual Network)

Port \*

1433

### SQL Authentication

SQL Authentication ⓘ

Disable

Enable

Login name \* ⓘ

dbauer

Password \* ⓘ

.....

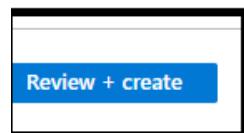
Azure Key Vault integration ⓘ

Disable

Enable

### Storage configuration

Customize performance, size, and workload type to optimize storage for this virtual machine. For optimal performance, separate drives will be created for data and log storage by default. [Learn more about SQL Server best performance practices.](#)



< Previous

Next : Tags >

CreateVm-MicrosoftSQLServer.SQL2016SP1-WS2016-SQL-20210210104955 | Overview ↗

Deployment

Search (Ctrl+ /) << Delete Cancel Redeploy Refresh

We'd love your feedback! →

Your deployment is complete

Deployment name: CreateVm-MicrosoftSQLServer.SQL2016SP1-WS... Start time: 2/10/2021, 11:00:33 AM  
Subscription: Free Trial Correlation ID: 28611f0f-cc7b-4361-a221-7e48039e0963  
Resource group: newgrp

Deployment details (Download)

Next steps

Go to resource

The screenshot shows the Azure portal's "Overview" page for a completed deployment named "CreateVm-MicrosoftSQLServer.SQL2016SP1-WS2016-SQL-20210210104955". The main content area displays deployment details: Deployment name, Subscription (Free Trial), Resource group (newgrp), and deployment start time (2/10/2021, 11:00:33 AM). Below this, there are two expandable sections: "Deployment details" and "Next steps". The "Next steps" section contains a prominent blue "Go to resource" button, which is also highlighted with a black rectangular border.

disks are created

SQVM | DISKS

Virtual machine

Search (Ctrl+ /)

Save Discard Refresh Additional settings

Swap OS disk

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings Networking Connect Disks Size Security Advisor recommendations Extensions Continuous delivery

Disk name Storage type Size (GiB) Max IOPS Max throughput (...) Encryption

sqvm\_OsDisk\_1\_0610a3efcd7040d0be7e3c Standard SSD 127 500 60 SSE with PMK

Data disks

Filter by name

Showing 4 of 4 attached data disks

Create and attach a new disk Attach existing disks

LUN	Disk name	Storage type	Size (GiB)	Max IOPS	Max throughput (...)	Encryption
0	sqlvm_DataDisk_0	Premium SSD	16	120	25	SSE with PMK
1	sqlvm_DataDisk_1	Premium SSD	16	120	25	SSE with PMK
2	sqlvm_DataDisk_2	Premium SSD	16	120	25	SSE with PMK
3	sqlvm_DataDisk_3	Premium SSD	16	120	25	SSE with PMK

Connect to rdp

sqlvm

Virtual machine

Search (Ctrl+ /)

Connect Start Restart Stop Capture Delete Refresh

Resource group (change) : newgrp

Status : Running

Location : East US 2

Subscription (change) : Free Trial

Subscription ID : 87c70a0b-ed61-4866-a52b-651e64ab1598

Tags (change) : Click here to add tags

Properties Monitoring Capabilities (7) Recommendations Tutorials

**Virtual machine**

Computer name	sqlvm
Operating system	Windows (Windows Server 2016 Datacenter)
Publisher	MicrosoftSQLServer
Offer	SQL2016SP1-WS2016
Plan	SQLDEV
VM generation	v1

download rdp file

Virtual machine

Search (Ctrl+ /)

To improve security, enable just-in-time access on this VM. →

RDP    SSH    BASTION

Connect with RDP

To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file.

IP address \*

Public IP address (52.177.183.87)

Port number \*

3389

Download RDP File

Can't connect?

Test your connection

Troubleshoot RDP connectivity issues

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Networking

Connect

Disks

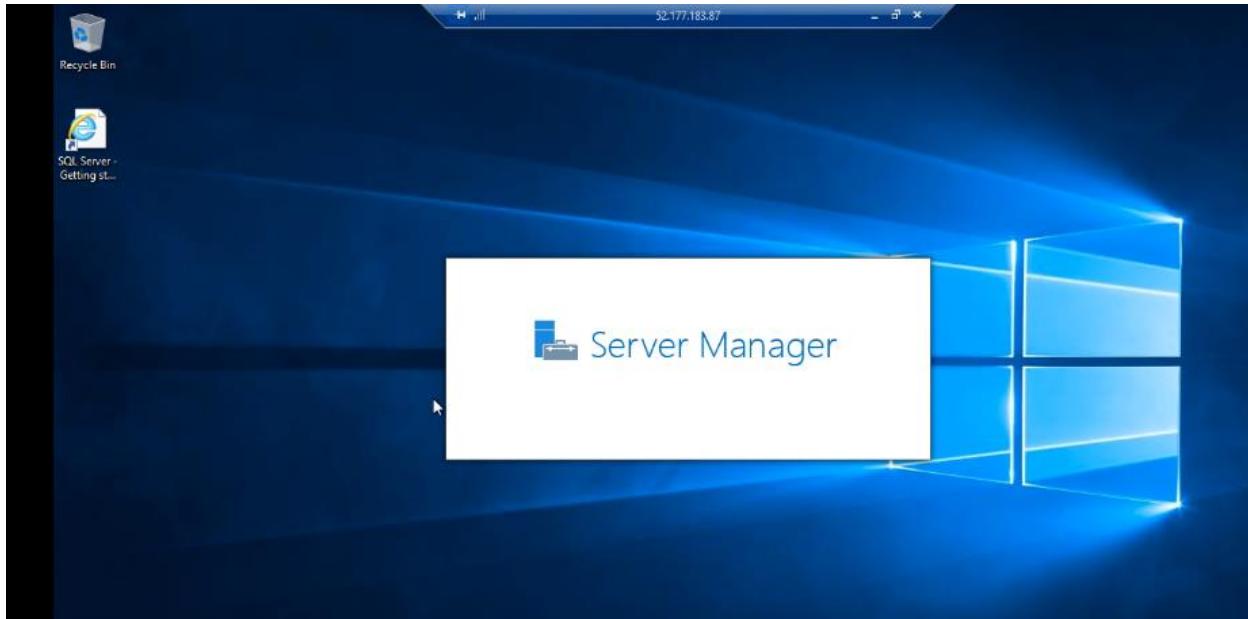
Size

Security

Advisor recommendations

Extensions

Remote server is opened



Drives are created

Server Manager • File and Storage Services • Volumes •

Servers  
Volumes **Volumes**  
Disks  
Storage Pools

**VOLUMES**  
All volumes | 5 total

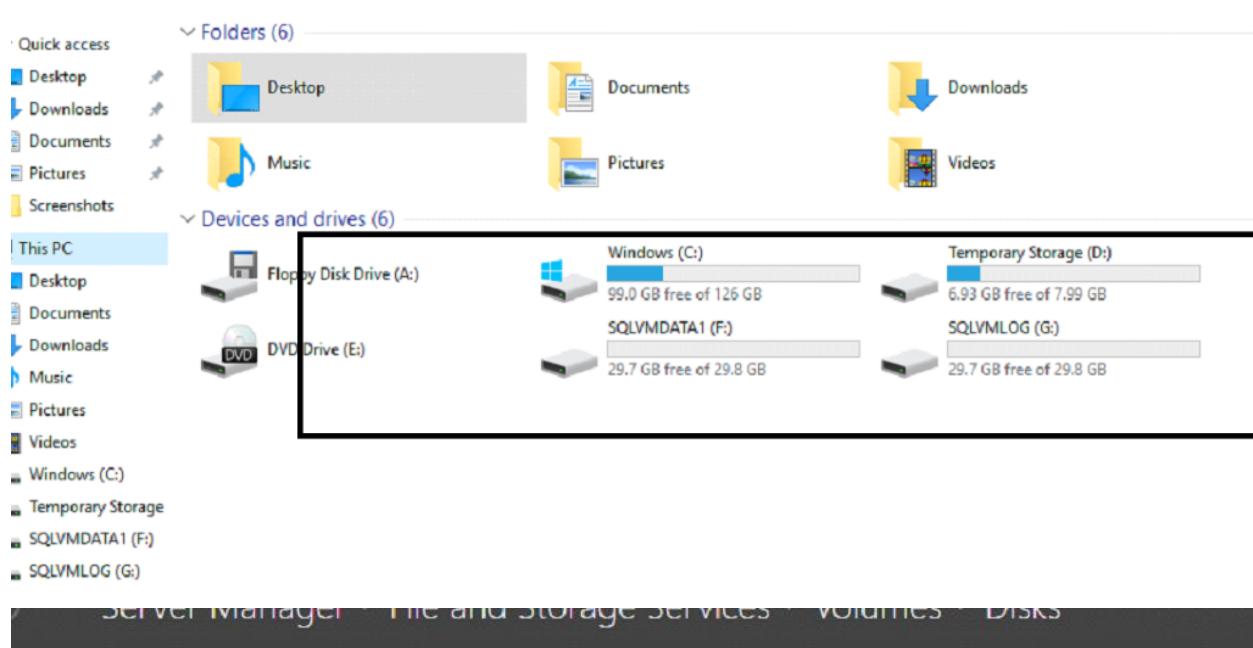
Volume	Status	File System	Label	Provisioning	Capacity	Free Space	Deduplication Rate	Deduplication S
Volume[60...]	System Reserved	Fixed		500 MB	466 MB			
C:	Windows	Fixed		127 GB	98.3 GB			
D:	Temporary Storage	Fixed		8.00 GB	6.94 GB			
F:	SQLVMDATA1	Fixed		29.9 GB	29.8 GB			
G:	SQLVMLOG	Fixed		29.9 GB	29.8 GB			

Last refreshed on 2/10/2021 5:57:49 AM

**SHARES**  
No volume is selected.

**DISK**  
No volume is selected.

To use this functionality, install the File Server role service.  
Start the Add Roles and Features Wizard.



SERVER Manager    FILE AND STORAGE SERVICES    VOLUMES    DISKS

Servers  
Volumes  
**Disks**  
Storage Pools

**DISKS**  
All disks | 4 total

Number	Virtual Disk	Status	Capacity	Unallocated	Partition	Read Only	Clustered	Subsystem
0		Online	127 GB	2.00 MB	MBR			
1		Online	8.00 GB	0.00 B	MBR			
6	SQLVMVirt...	Online	30.0 GB	0.00 B	GPT			Windows S...
7	SQLVMVirt...	Online	30.0 GB	0.00 B	GPT			Windows S...

Last refreshed on 2/10/2021 6:00:46 AM

**VOLUMES**  
Related Volumes | 2 total

Volume	Status	Provisioning	Capacity	Free Space	Deduplication Rate	Dedu

**STORAGE POOL**  
Virtual HD on sqh

STORAGE POOLS									TASKS	
All storage pools   2 total										
	Name	Type	Managed by	Available to	Read-Write Server	Capacity	Free Space	Percent Allocated	Status	
<b>Windows Storage (2)</b>										
	SQLVMStoragePool1	Storage Pool	sqlvm	sqlvm	sqlvm	30.5 GB	0.00 B	100%	Optimal	
	SQLVMStoragePool2	Storage Pool	sqlvm	sqlvm	sqlvm	30.5 GB	0.00 B	100%	Optimal	

Last refreshed on 2/10/2021 6:02:55 AM

VIRTUAL DISKS  
SQLVMStoragePool1 on sqlvm

	Name	Status	Layout	Provisioning	Capacity	Allocated	Volume	Cluster
	SQLVMVirtualDisk1	Simple	Fixed	30.0 GB	30.0 GB	F:		

PHYSICAL DISKS  
SQLVMStoragePool1 on sqlvm

	Slot	Name	Status	Capacity	Bus	Usage	Chassis
		Msft Virtual Disk (sqlvm)	Optimal	16.0 GB	SAS	Automatic	Integrate
		Msft Virtual Disk (sqlvm)	Optimal	16.0 GB	SAS	Automatic	Integrate

**Q:**Copy on prem database backup and restore in Azure VM.

Test1 database is restored with full backup

Screenshot of SQL Server Management Studio (SSMS) showing the properties of a database named 'TEST1'.

The 'Recovery model' dropdown is highlighted with a red box, showing the value 'Full'.

**Collation:** SQL\_Latin1\_General\_CI\_AS

**Recovery model:** Full

**Compatibility level:** SQL Server 2017 (140)

**Containment type:** None

**Other options:**

**Automatic**

Auto Close	False
Auto Create Incremental Statistics	False
Auto Create Statistics	True
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False

**Containment**

Default Fulltext Language LCID	1033
Default Language	English
Nested Triggers Enabled	True
Transform Noise Words	False
Two Digit Year Cutoff	2049

**Cursor**

Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL

**Database Scoped Configurations**

Legacy Cardinality Estimation	OFF
Legacy Cardinality Estimation For Secondary	PRIMARY
Max DOP	0

**Auto Close**



## Set Properties

Introduction

 Help

### Set Properties

Validation and Summary

Build Package

#### Set the DAC properties.

Application name:

TEST1

Version (use x.x.x where x is a number):

1.0.0.0

Description:

Save to DAC package file (include .dacpac extension with the file name):

ivanip\Documents\SQL Server Management Studio\DAC Packages\TEST1.dacpac

 Browse... Overwrite existing file

The application name, version, and description are displayed in SQL Server Management Studio after the DAC has been deployed.

 < Previous Next > Finish Cancel

Results

Production  
Export Settings  
Summary  
**Results**

**Operation Complete**

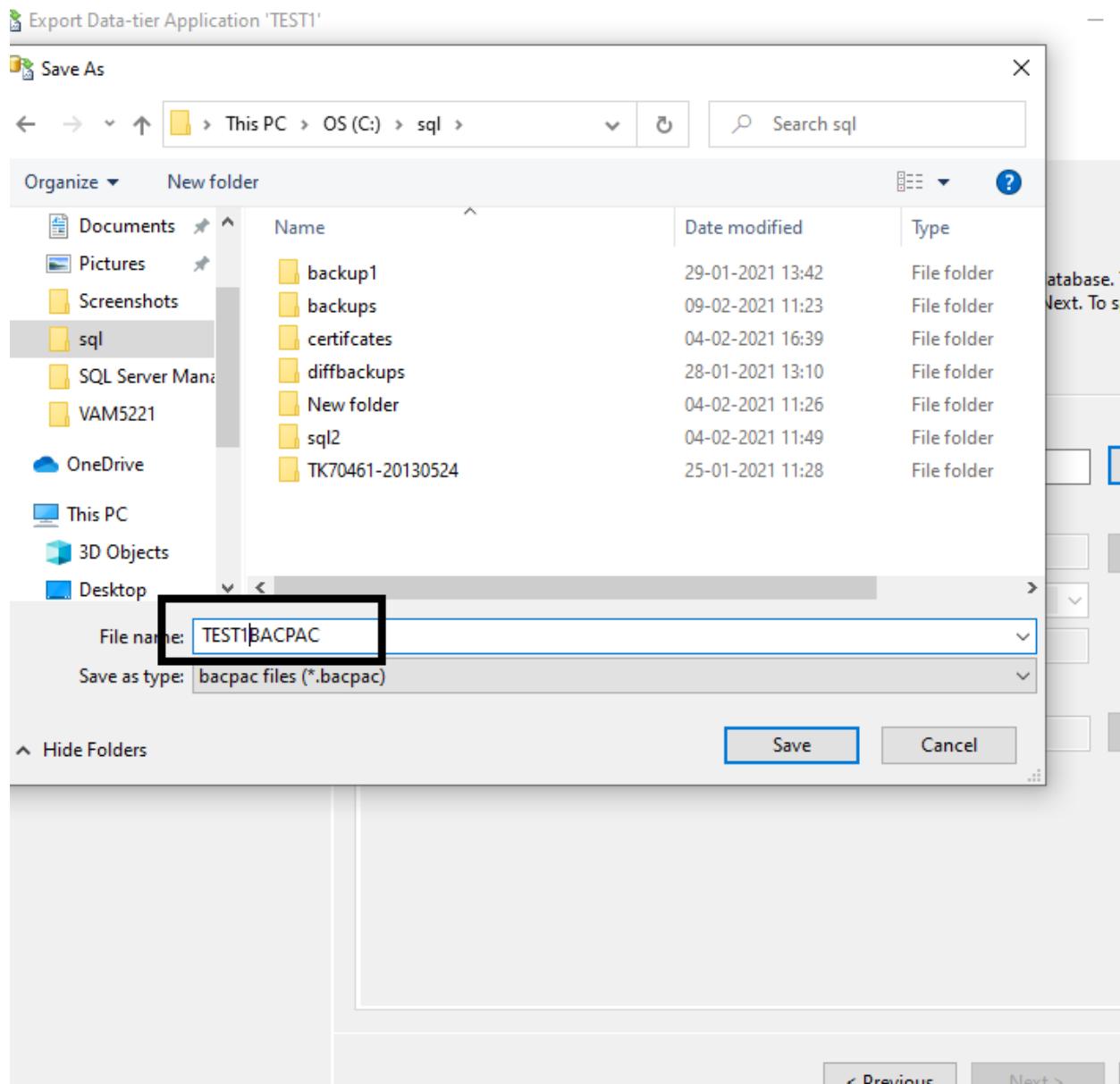
Summary:

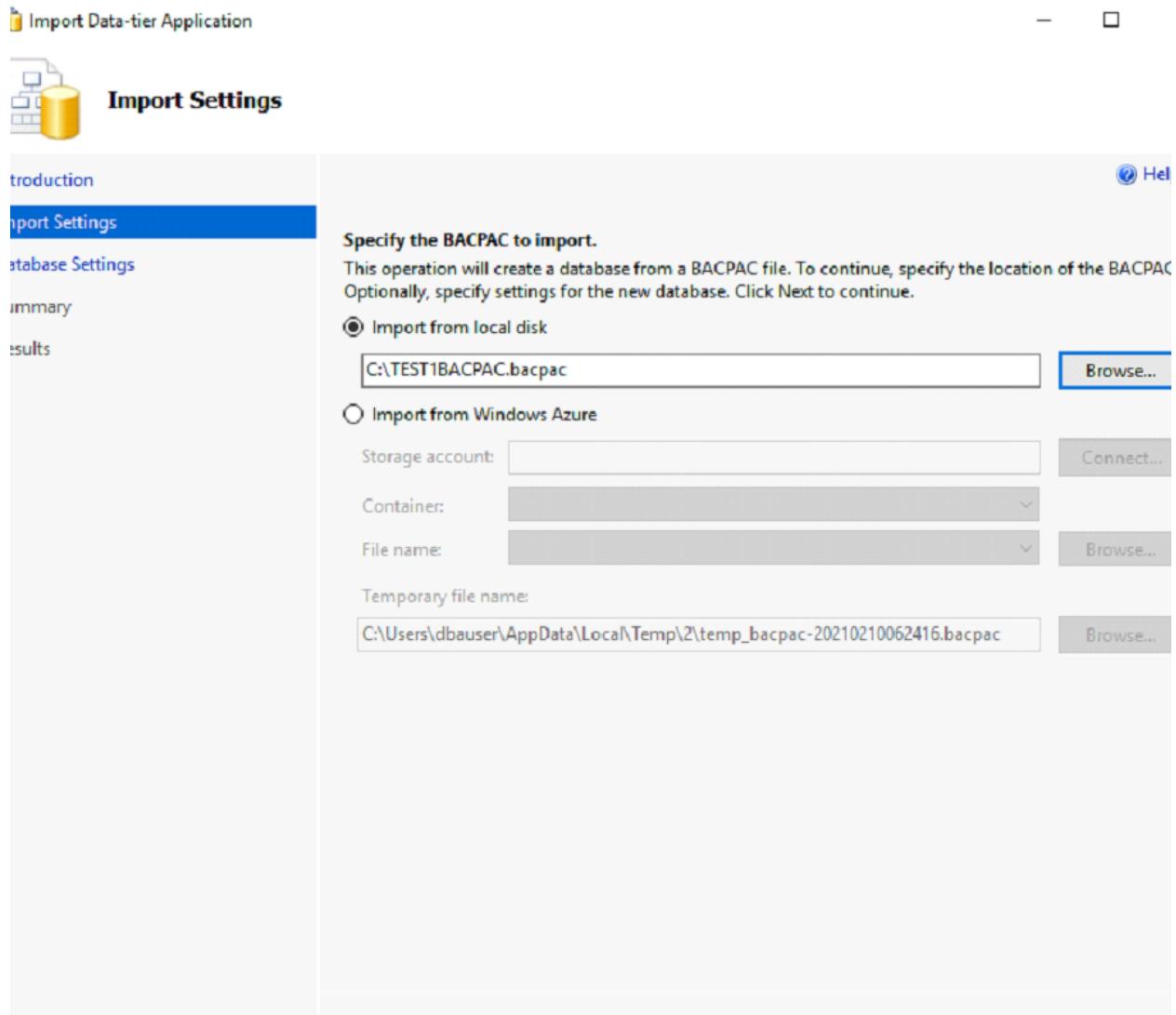
Name	Result
Exporting database	Success
Extracting schema	Success
Extracting schema from database	Success
Resolving references in schema model	Success
Validating schema model	Success
Validating schema model for data package	Success
Validating schema	Success
Exporting data from database	Success
Exporting data	Success
Processing Export.	Success
Processing Table '[HumanResources].[EmployeePayHistory]'	Success
Processing Table '[Sales].[SalesOrderHeaderSalesReason]'	Success
Processing Table '[Sales].[SalesPerson]'	Success
Processing Table '[Production].[Illustration]'	Success
Processing Table '[HumanResources].[JobCandidate]'	Success
Processing Table '[Production].[Location]'	Success
Processing Table '[Person].[Password]'	Success
Processing Table '[Sales].[SalesPersonQuotaHistory]'	Success
Processing Table '[Person].[Person]'	Success

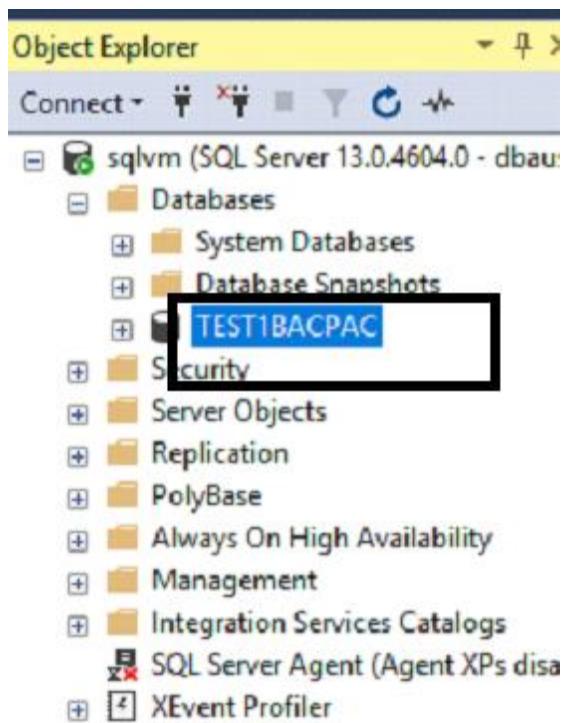
< Previous    Next >    **Close**

In RDP importing data:

testbacpac file is created







## Data migration Assistant:

The Data Migration Assistant (DMA) helps you upgrade to a modern data platform by detecting compatibility issues that can impact database functionality in your new version of SQL Server or Azure SQL Database. DMA recommends performance and reliability improvements for your target environment and allows you to move your

- Assess on-premises SQL Server instance(s) migrating to Azure SQL database(s).  
The assessment workflow helps you to detect the following issues that can affect

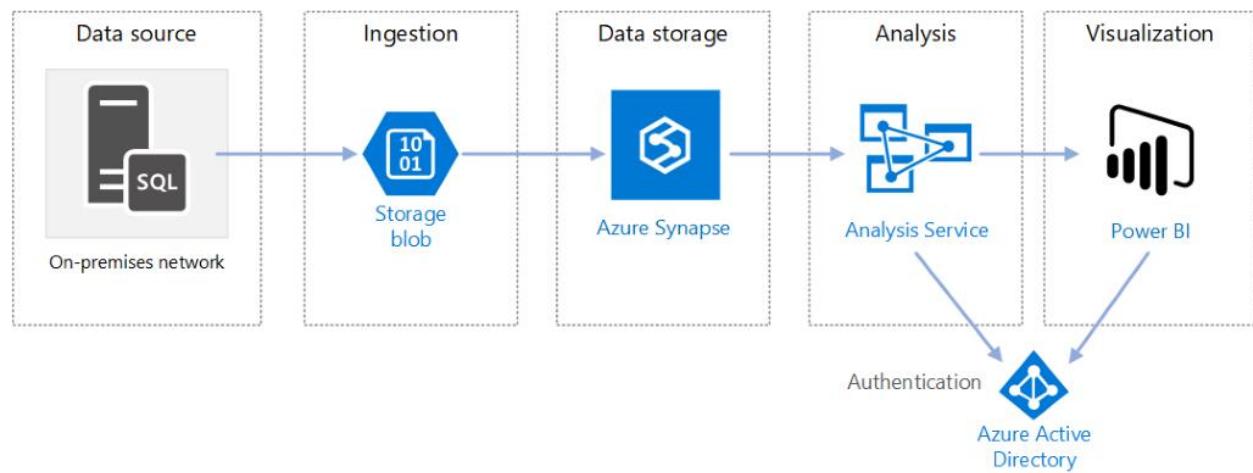
Azure SQL database migration and provides detailed guidance on how to resolve them.

- Migration blocking issues: Discovers the compatibility issues that block migrating on-premises SQL Server database(s) to Azure SQL Database(s). DMA provides recommendations to help you address those issues.
- Partially supported or unsupported features: Detects partially supported or unsupported features that are currently in use on the source SQL Server instance. DMA provides a comprehensive set of recommendations, alternative approaches available in Azure, and mitigating steps so that you can incorporate them into your migration projects.
- Discover issues that can affect an upgrade to an on-premises SQL Server. These are described as compatibility issues and are organized in the following categories:
  - Breaking changes
  - Behavior changes
  - Deprecated features
- Discover new features in the target SQL Server platform that the database can benefit from after an upgrade. These are described as feature recommendations and are organized in the following categories:
  - Performance
  - Security
  - Storage
- Migrate an on-premises SQL Server instance to a modern SQL Server instance hosted on-premises or on an Azure virtual machine (VM) that is accessible from your on-premises network. The Azure VM can be accessed using VPN or other technologies. The migration workflow helps you to migrate the following components:
  - Schema of databases
  - Data and users
  - Server roles
  - SQL Server and Windows logins
- After a successful migration, applications can connect to the target SQL Server databases seamlessly.
- Assess on-premises SQL Server Integration Services (SSIS) package(s) migrating to Azure SQL Database or Azure SQL Managed Instance. The assessment helps to discover issues that can affect the migration. These are described as compatibility issues and are organized in the following categories:

- Migration blockers: discovers the compatibility issues that block migrating source package(s) to Azure. DMA provides recommendations to help you address those issues.
- Information issues: detects partially supported or deprecated features that are used in source package

## **Data warehousing:**

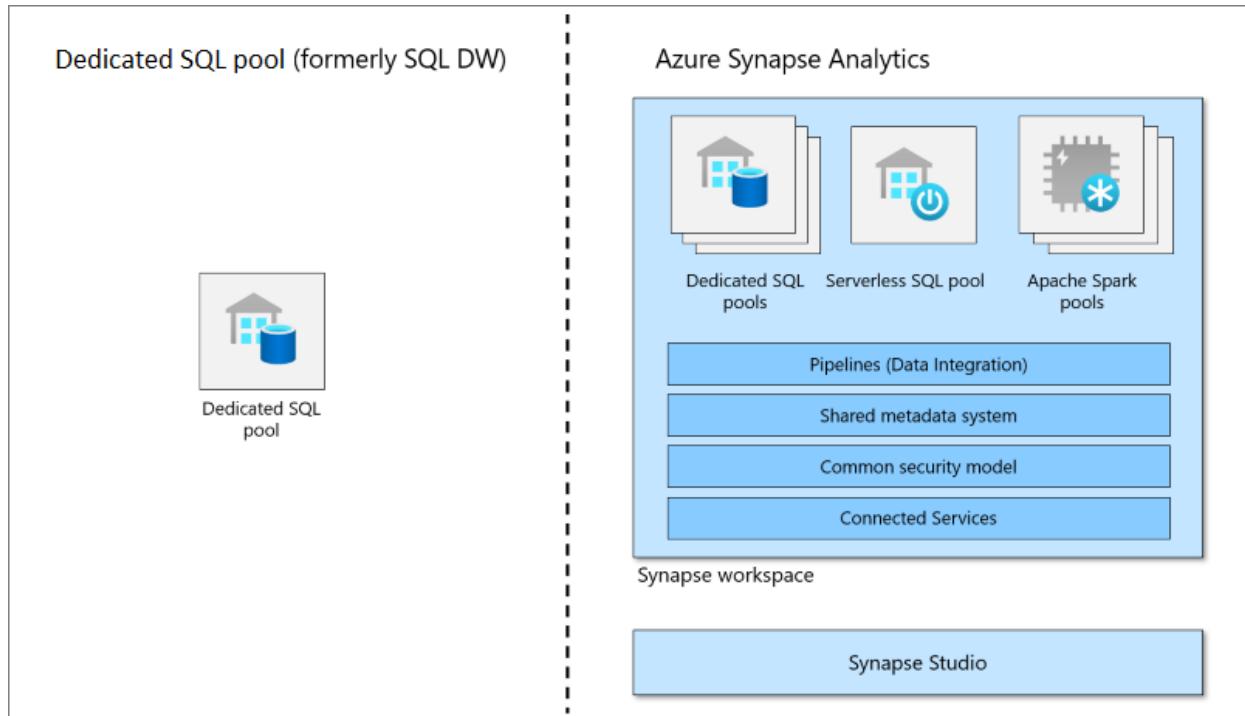
A data warehouse is a centralized repository of integrated data from one or more disparate sources. Data warehouses store current and historical data and are used for reporting and analysis of the data



To move data into a data warehouse, data is periodically extracted from various sources that contain important business information. As the data is moved, it can be formatted, cleaned, validated, summarized, and reorganized. Alternatively, the data can be stored in the lowest level of detail, with aggregated views provided in the warehouse for reporting. In either case, the data warehouse becomes a permanent data store for reporting, analysis, and business intelligence (BI).

## **Azure synapse analytics:**

It is formerly known as data warehousing. Azure Synapse Analytics is an analytics service that brings together enterprise data warehousing and Big Data analytics. Dedicated SQL pool (formerly SQL DW) refers to the enterprise data warehousing features that are available in Azure Synapse Analytics.



Dedicated SQL pool (formerly SQL DW) represents a collection of analytic resources that are provisioned when using Synapse SQL. The size of a dedicated SQL pool (formerly SQL DW) is determined by Data Warehousing Units (DWU).

Once your dedicated SQL pool is created, you can import big data with simple PolyBase T-SQL queries, and then use the power of the distributed query engine to run high-performance analytics. As you integrate and analyze the data, dedicated SQL pool (formerly SQL DW) will become the single version of truth your business can count on for faster and more robust insights.

Q: Create Azure synapse analytics database, access it similar to azure db.

Home >

## Azure Synapse Analytics

fault Directory

[Add](#) Manage view ▾ Refresh Export to CSV Open query | Assign tags | Feedback

Subscription == all Resource group == all Location == all [+ Add filter](#)

Showing 0 to 0 of 0 records.

Name ↑↓	Type ↑↓	Resource group
---------	---------	----------------



Home > Azure Synapse Analytics >

## Create Synapse workspace

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like [tags](#) if your resources.

Subscription \* ⓘ

Free Trial

 The Synapse and SQL resource providers are not included in this subscription.

Resource group \* ⓘ

newgrp

[Create new](#)

Managed resource group ⓘ

sqlmrg

### Workspace details

Name your workspace, select a location, and choose a primary Data Lake Storage Gen2 file location for logs and job output.

Workspace name \*

sqlwork

Region \*

East US 2

Select Data Lake Storage Gen2 \* ⓘ

From subscription  Manually via URL

## Create Synapse workspace

Select Data Lake Storage Gen2

Account name \* (New) sqlacc [Create new](#)

File system name \* (New) sqlfile [Create new](#)

Assign myself the Storage Blob Data Contributor role on the Data Lake Storage Gen2 account 'sqlacc'.

**i** We will automatically grant the workspace identity data access to the specified Data Lake Storage Gen2 account, using the **Storage Blob Data Contributor** role. To enable other users to use this storage account after you create your workspace, perform these tasks:

- Assign other users to the **Contributor** role on workspace
- Assign other users to a **Workspace, SQL, or Spark admin** role using Synapse Studio
- Assign yourself and other users to the **Storage Blob Data Contributor** role on the storage account

[Learn more](#)

**Review + create** < Previous Next: Security >

Type here to search Microsoft Azure search resources, services, and docs (F11)

me > vmmrg > **vmfile** Container

Search (Ctrl+ /) Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break

Authentication method: Access key (Switch to Azure AD User Account)  
Location: vmfile

Search blobs by prefix (case-sensitive)

Name	Modified	Access tier	Blob type	Size
No results				

**Upload blob** vmfile/

Files (0) "Azureresources.csv" Overwrite if file already exists

Advanced

Upload

Home >

# vmwork

Synapse workspace

Search (Ctrl+ /)

New dedicated SQL pool

+ New Apache Spark pool Refresh Reset SQL admin

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

SQL Active Directory admin

Properties

Locks

Analytics pools

New dedicated SQL pool

Resource group (change) : newgrp

Status : Succeeded

Location : East US 2

Subscription (change) : Free Trial

Subscription ID : 87c70a0b-ed61-4866-a52b-651e64ab1598

Managed virtual network : No

Managed Identity object ... : 283758e7-1792-4de2-8121-b8bbc456a208

Workspace web URL : <https://web.azuresynapse.net?workspace=%2fsub...>

Tags (change) : Click here to add tags

Getting started

Open Synapse Studio Read documentation

Decreasing of DWU then cost is :

Microsoft Azure Search resources, services, and docs (G+ /)

> vmwork >

## Create dedicated SQL pool

\* Additional settings Tags Review + create

Create a dedicated SQL pool with your preferred configurations. Complete the Basics tab then go to Review + Create to finish with smart defaults, or visit each tab to customize. [Learn more](#)

Dedicated SQL pool details

Select your dedicated SQL pool and choose its initial settings.

Dedicated SQL pool name \*

Performance level  DW 100c  DW 100c

Priced price  [View pricing details](#)

**Est. Cost Per Hour**  
86.45 INR [View pricing details](#)

[view + create](#) [Next: Additional settings >](#)

Increasing of DWU then cost is :

Dedicated SQL pool name \*

Performance level  DW 8000c

The performance level selected is currently limited in this region. To create a dedicated SQL pool at this performance level, see available regions.

With the currently selected performance level, the quota for this server will be exceeded. To create a dedicated SQL pool at this performance level, please submit a quota support request using the following options:

Quota type: Azure Synapse Analytics  
Details: Please provide the DWU required

Estimated price  Est. Cost Per Hour  
25936.29 INR  
[View pricing details](#)

[Review + create](#) [Next: Additional settings >](#)

## Geo Replication:

- In the Azure portal browse to the database that you want to set up for geo-replication.
- On the SQL Database page, select **geo-replication**, and then select the region to create the secondary database. You can select any region other than the region hosting the primary database, but we recommend the paired region

sqlazure (vmsrv/sqlazure) [SQL database](#)

Status: Online | Location: East US | Subscription: Free Trial | Subscription ID: 87c70a0b-ed61-4866-a52b-651e64ab1598 | Tags: Click here to add tags

Compute utilization

100%  
90%  
80%  
70%  
60%  
50%  
40%  
30%

Show data for last:    Aggregation type:

Select or configure the server and pricing tier for the secondary database.

← → ⌂ https://portal.azure.com/#@shivaniredd

Microsoft Azure Search resources, services, and docs

Home > sqlazure (vmsrv/sqlazure) >

## Create secondary

**1 Create geo-replicated secondaries to protect against prolonged datacenter outages.**

---

Region  lock

Database name

---

\*Secondary type  lock

\*Target server >

*Configure required settings*

---

Elastic pool lock

None

---

\*Pricing tier

**OK** cancel

---

Azureresources.csv  ...

When the seeding process is complete, the secondary database displays its status.

The screenshot shows the Azure portal interface for managing a SQL database named 'sqlazure'. The left sidebar includes options like Tags, Diagnose and solve problems, Quick start, Query editor (preview), Power Platform (Power BI, Power Apps, Power Automate), Configuration, Geo-Replication (selected), Connection strings, and Sync to other databases. The main content area displays the 'Geo-Replication' settings for the 'sqlazure' database. It shows a primary database in 'East US' and a secondary database in 'South Central US' with a status of 'Online' and no failover policy. A progress bar indicates 'Seeding 0%'. Below this, a list of 'Target regions' is shown, with 'South Central US' being the only one currently selected and marked as 'Readable'.

## REFERENCES

**Microsoft Azure**

[Get started guide for developers on Azure | Microsoft Docs](#)

**Splunk**

[Splunk Cloud™ - Splunk Documentation](#)

## **Docker**

[Docker overview | Docker Documentation](#)

## **Kubernetes**

[Kubernetes Documentation | Kubernetes](#)

## **Azure Kubernetes Services**

[Azure Kubernetes Service \(AKS\) documentation | Microsoft Docs](#)

## **Azure DevOps**

[Azure DevOps documentation | Microsoft Docs](#)