

# AZURE MS-SQL DAILY ACTIVITY

## PRANEET AYUSH MANDA.

22/01/24 ACTIVITY: IMPLEMENT PDSH.

```
azureuser@vm-22jan-praneet:~$ sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3028 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [489 kB]
Get:13 http://azure.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2579 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [360 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1154 kB]
Get:16 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [277 kB]
Get:17 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [25.7 kB]
Get:18 http://azure.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [26.1 kB]
Get:19 http://azure.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7768 B]
Get:20 http://azure.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [620 B]
Get:21 http://azure.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [45.7 kB]
Get:22 http://azure.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [16.3 kB]
Get:23 http://azure.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [1420 B]
Get:24 http://azure.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://azure.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [25.0 kB]
Get:26 http://azure.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [16.3 kB]
Get:27 http://azure.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [880 B]
Get:28 http://azure.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://azure.archive.ubuntu.com/ubuntu focal-security/main amd64 Packages [2647 kB]
Get:30 http://azure.archive.ubuntu.com/ubuntu focal-security/main Translation-en [406 kB]
Get:31 http://azure.archive.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [2461 kB]
Get:32 http://azure.archive.ubuntu.com/ubuntu focal-security/restricted Translation-en [343 kB]
Get:33 http://azure.archive.ubuntu.com/ubuntu focal-security/universe amd64 Packages [929 kB]
Get:34 http://azure.archive.ubuntu.com/ubuntu focal-security/universe Translation-en [195 kB]
Get:35 http://azure.archive.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [19.2 kB]
Get:36 http://azure.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [23.9 kB]
Get:37 http://azure.archive.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5796 B]
Get:38 http://azure.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [548 B]
Fetched 29.7 MB in 5s (5500 kB/s)
Reading package lists... Done
azureuser@vm-22jan-praneet:~$ sudo apt-get install pdsh
Reading package lists... Done
```

23/01/24:

CREATION OF MS SQL SERVER AND DATABASE AND CONNECTION USING DBEAVER.

The screenshot shows the Azure Resource Group 'rg\_24jan\_sql' overview page. Key details include:

- Subscription: Inedo\_AZ\_Training
- Subscription ID: 59c106e1-c9e3-4440-961d-8bd04fe978a7
- Tags: None (>Add tags)
- Location: East US
- Deployments: 1 Succeeded

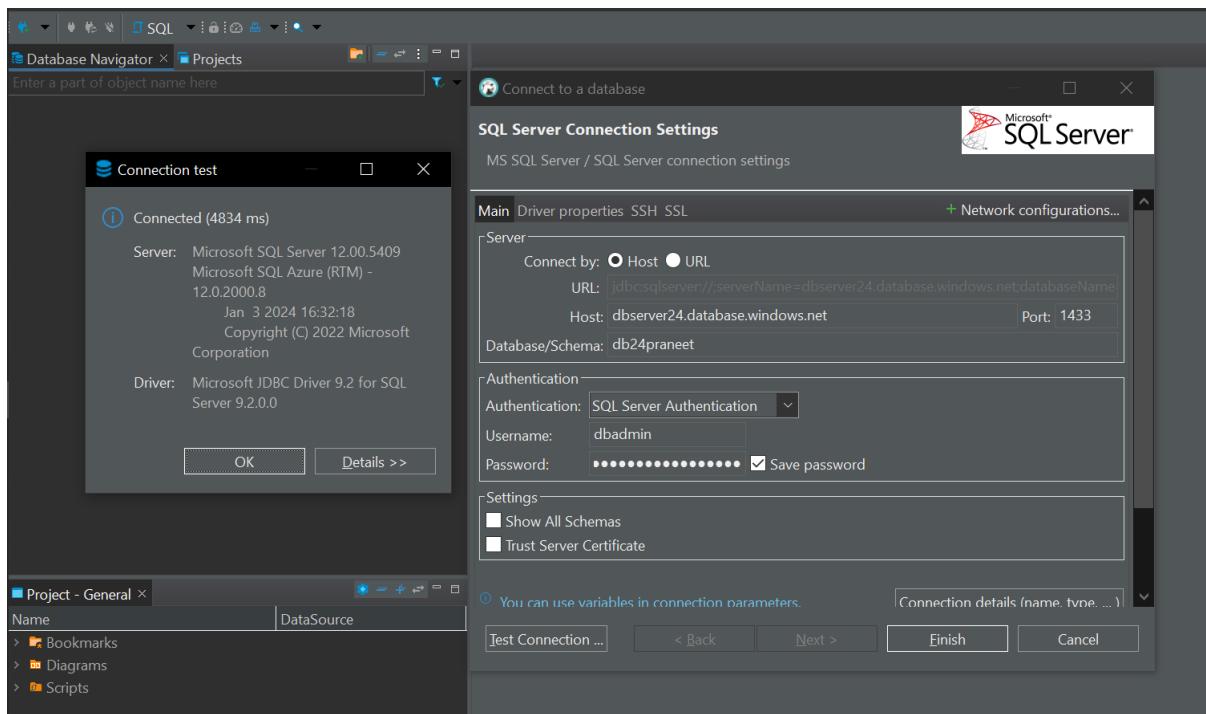
## Create SQL Database

Microsoft

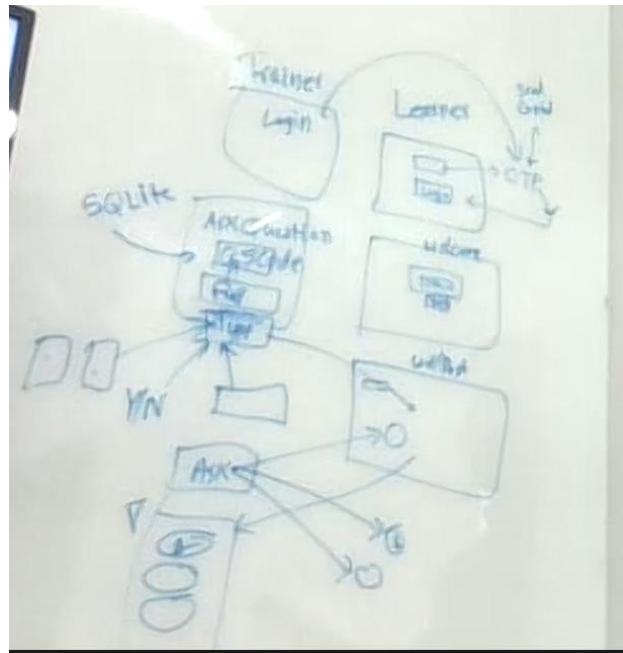
By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. For additional details see [Azure Marketplace Terms](#).

### Basics

Subscription	Inedo_AZ_Training
Resource group	rg_24jan_sql
Region	East US
Database name	db24praneet
Server	(new) dbserver24
Authentication method	SQL authentication
Server admin login	dbadmin
Compute + storage	General Purpose - Serverless: Standard-series (Gen5), 1 vCore, 32 GB storage, zone redundant disabled
Backup storage redundancy	Locally-redundant backup storage



## 2.) application:



### 1. LEARNER POV.

- LOGIN PAGE: 2 STEP VERIFICATION (SHOULD HAVE A LOGIN BUTTON) (USING MAIL ADDRESS AND PASSWORD) (send grid otp) -> welcome page and if it is the first time then a pop up menu.
- FIRST TIME: MAKE A SIGN UP PAGE. (specify full name)
- MAKE AN EXCEL SHEET OF THE RESPONSIBILITIES.
- Python & Django.
- Dashboard/Welcome: (Contains a list)

2. TRAINER POV:

- SAME LOGIN PAGE.
- NEXT PAGE SHOULD CONTAIN: QUESTION SHORT CODE, ASK QUESTIONS, ANSWER: YES, NO, ERROR.
- ASK QUESTION: SHOULD PROMPT TO LEARNER AND BASED ON THEIR RESPONSE IT WILL DISPLAY THAT HOW MANY STUDENTS HAVE ANSWERED THE QUESTION (RIGHT OR WRONG) AND HOW MANY WERE UNABLE TO ANSWER. THE ANSWERS GO INTO A QUEUE. FIRST COME FIRST SERVE PRIORITY.
- ALL OF THIS SHOULD BE DEPLOYED IN MICROSOFT AZURE.

A screenshot of a Microsoft OneNote page titled "23Jan.Deployments of An applicat...". The page shows a table of deployment details. The table has two columns: "Web Framework" and "Database Server".

Web Framework	Django
Database Server	SQL Server
Runtime	VM --> Serverless
VCS	Github
Editor	Online/Offline VSCode
Package Availability	1. VM(Terminal) 2. SSH Tunnel

SQL:

## SQL

- Query Language
- Standard (ANSI SQL)
  - Close Source :
    - MS SQL Server(Paid)
  - Open Source :
    - MySQL Server(Community)
- MS SQL Server
  - Implementation of
    - Standard SQL
    - RDBMS
    - Package for OORDBMS(BI)---> DWH(ETL) --> BigData
      - SSI(Integration)S, SSR(Reporting)S, SSA(Analysis)S
    - Tools
      - CLI - SQL Trace
      - GUI - SQL Profiler
      - DB IDE - Data Studio
      - SQL Client (GUI) - Sql Server Management Studio
      - SQL Client (CLI) - SQLCMD



Mujahed > Trainings > Cor

## SQL - MS SQL Server, MySQL, Postgres, etc

### NoSQL -

- Categories:
  - Document Oriented
  - Key- Value Oriented
  - Graph Oriented
  - Time Series
  - Columnar
- Implementations of NoSQL
  - MongoDB, DynamoDB, GraphQL, Neo4J, Cosmos, Cassandra, Hbase
  - 256 NoSQL

## Azure SQL

- Server -->|
- Database -->
- Client

## Azure SQL

- Server --> infinite program to bring data files
- Database --> set of objects(Table, View, Trigger, Stored Procedure, Functions, Packages, etc)
  - Master(Simple Data Entry Table)
    - Helps to Perform transactions
    - Product
    - Category
  - Transactional(Relational Complex)
    - Stock
    - Sales
    - Purchase
  - Temp Table
  - CTE(Common Table Expression)
- Client

Dedicated Machine --> VM(Shared) --> Cluster --> Instance --> Databases ---> Database --> Tables/Object --> Row/cols

## Creation of Azure SQL

1. Active Subscription
2. Create Resource Group
3. Azure SQL
  - a. Database Server
  - b. Database
  - c. Credentials
  - d. Order

24 Jan.

Data Containerization in Cloud:

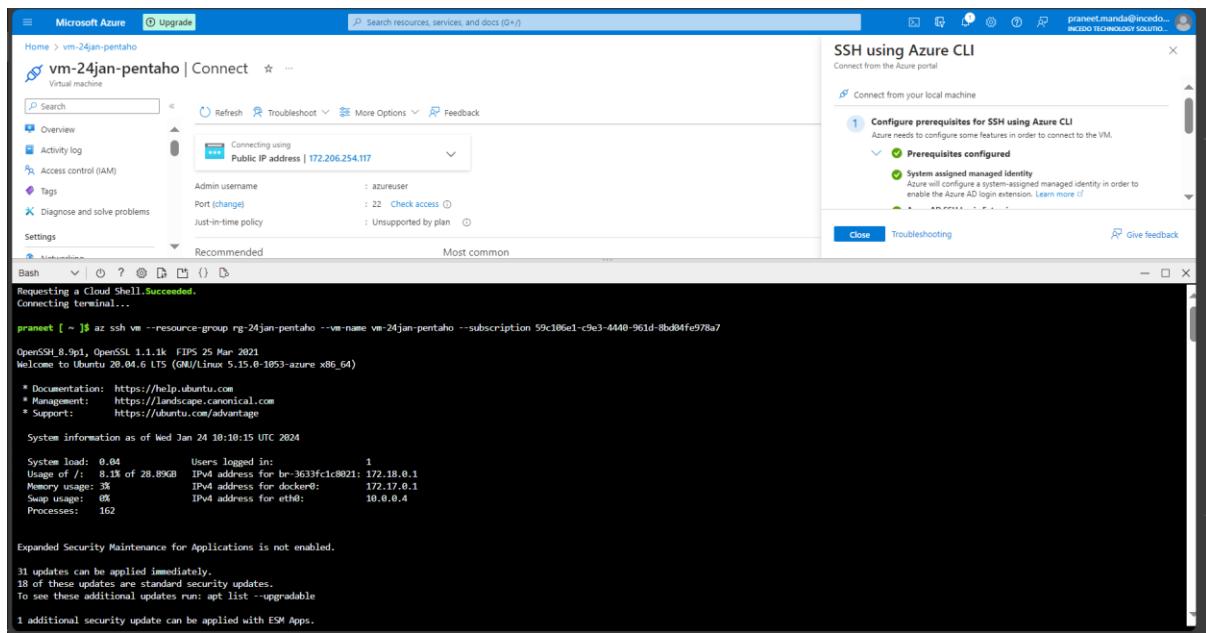
Data:

- Traditional Data (RDBMS) 30%. (not capable of handling all kind of data)(only structured)
- Big Data & New Data 70%.
  - DWH, ETL, BI Hadoop, Informatica, Pentaho (helps to integrate any kind of data and perform transformations on it), Spark, Databricks.

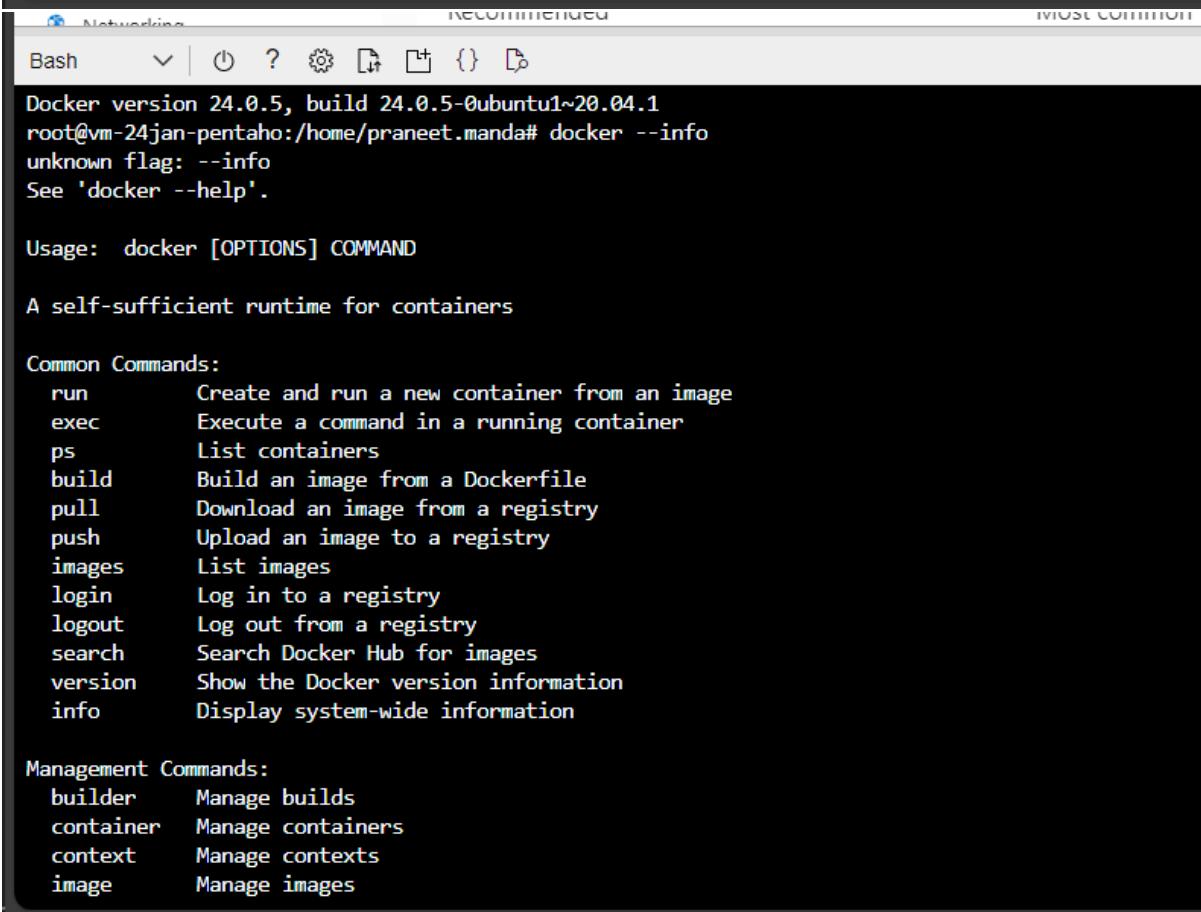
Virtualization (Microsoft Azure V) Containerization (Docker):

- Image -File(Local/Remote)
- Container –Process/Running File(Image)
- (Archive Registry/Container Registry)
  - Market Place(Remote Image)
    - Hub.Docker.com
    - Microsoft → ACR – Azure Container Registry
    - Google GCR, ECR, etc.
  - Shops (Repo)
    - Docker pull mujahed/welcome1:tagname
    - ShopName/ProductName:Version

1. Check Sub
2. Create Resource Group
3. Order VM
  - a. Configuration 4CPU, 16GB
  - b. UserName: azureuser
  - c. Password
4. Connect VM
  - a. Azure CLI
    - i. \$ az ssh vm --resource group r1 --vm name v1 --subscription s1
  - b. SSH CLI
  - c. GUI Tools
    - i. MobaXterm
    - ii. Putty
5. Install Docker:
  - a. \$ sudo su
  - b. # apt update
  - c. # apt install docker.io -y
  - d. # docker --version
  - e. # docker --info
6. Create Application Deployment
  - a. Create Docker Network
    - i. # docker network create incedo-nw
  - b. Pull Remote Image from CR(hub.docker.com)
    - i. # docker images
    - ii. # docker pull nginx:latest
  - c. Local Image(nginx:latest) → Container [detachable mode]
    - i. # docker run --rm --net nw --name ws --d(detachable) nginx:latest
    - ii. #export CID=[Container id]
    - iii. #docker inspect \$CID | grep IPAddress
    - iv. #export WSIP=[Container IP]
  - d. Pull Remote Image → Local Image → Container
    - i. # docker run --net nw -it busybox sh
  - e. Test Web Server From 2<sup>nd</sup> Container
    - i. # wget -q -O -http://ContainerName:80
    - ii. wget -q -O -http://IPAddress:80
    - iii. wget -q -O -http://IPAddress
    - iv. Wget -q -O -IPAddress
    - v. Wget -q -O -ContainerName
  - f. Delete All Container & Images
    - i. # docker rm \$(docker ps -aq) --force && docker rmi \$(docker images -q) --force



```
Last login: Wed Jan 24 10:03:06 2024 from 4.224.4.218
praneet.manda@incedoinc.com@vm-24jan-pentaho:~$ 
praneet.manda@incedoinc.com@vm-24jan-pentaho:~$ sudo su
root@vm-24jan-pentaho:/home/praneet.manda# apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://packages.microsoft.com/repos/microsoft-ubuntu-focal-prod focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
33 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@vm-24jan-pentaho:/home/praneet.manda# apt install docker.io -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (24.0.5-0ubuntu1~20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 33 not upgraded.
root@vm-24jan-pentaho:/home/praneet.manda# docker --version
```



The screenshot shows a terminal window with the title bar "Bash" and various icons. The terminal displays the Docker documentation:

```
Docker version 24.0.5, build 24.0.5-0ubuntu1~20.04.1
root@vm-24jan-pentaho:/home/praneet.manda# docker --info
unknown flag: --info
See 'docker --help'.

Usage: docker [OPTIONS] COMMAND
      A self-sufficient runtime for containers

Common Commands:
  run      Create and run a new container from an image
  exec     Execute a command in a running container
  ps       List containers
  build    Build an image from a Dockerfile
  pull     Download an image from a registry
  push     Upload an image to a registry
  images   List images
  login    Log in to a registry
  logout   Log out from a registry
  search   Search Docker Hub for images
  version  Show the Docker version information
  info     Display system-wide information

Management Commands:
  builder  Manage builds
  container Manage containers
  context   Manage contexts
  image    Manage images
```

```
Bash x | o ? & { } &

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

root@vm-24jan-pentaho:/home/praneet.manda# docker network create incedo-nw
Error response from daemon: network with name incedo-nw already exists
root@vm-24jan-pentaho:/home/praneet.manda# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@vm-24jan-pentaho:/home/praneet.manda# docker pull nginx:latest
latest: Pulling from library/nginx
2f44b7a888fa: Pull complete
8b7dd3ed1dc3: Pull complete
35497dd96569: Pull complete
36664b6cce6b: Pull complete
2d455521f76c: Pull complete
dc94fd83d6: Pull complete
8056d2bcfc3b6: Pull complete
Digest: sha256:4cfdaa8b6341bfdec5a518f7837462c80cff90527ee35ef185571e1c327beac
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
root@vm-24jan-pentaho:/home/praneet.manda# docker run --rm --net incedo-nw --name ws1 -d nginx:latest
Unable to find image 'name:latest' locally
docker: Error response from daemon: pull access denied for name, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
root@vm-24jan-pentaho:/home/praneet.manda# docker run --rm --net incedo-nw --name ws1 -d nginx:latest
Unable to find image 'name:latest' locally
docker: Error response from daemon: pull access denied for name, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
root@vm-24jan-pentaho:/home/praneet.manda# docker run --rm --net incedo-nw --name ws1 -d nginx:latest

SEE 'docker run --help'.

root@vm-24jan-pentaho:/home/praneet.manda# docker run --rm --net incedo-nw --name ws2 -d nginx:latest
2e98ec86000c822aa4894edfc3a211903d8d56c08eccbd1a64a508b7c82552f5
root@vm-24jan-pentaho:/home/praneet.manda# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2e98ec86000c nginx:latest "/docker-entrypoint..." 5 seconds ago Up 5 seconds 80/tcp ws2
root@vm-24jan-pentaho:/home/praneet.manda# export CID=2e98ec86000c
root@vm-24jan-pentaho:/home/praneet.manda# docker inspect $CID | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "",
        "IPAddress": "172.18.0.2",
root@vm-24jan-pentaho:/home/praneet.manda# export WSIP=172.18.0.2
root@vm-24jan-pentaho:/home/praneet.manda# docker run --net incedo-nw -it busybox sh
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
9ad63333ebc9: Pull complete
Digest: sha256:6d9ac9237a84afe1516540f40a0fafdc86859b2141954b4d643af7066d598b74
Status: Downloaded newer image for busybox:latest
/ # wget -q -O - http://ContainerName:80
BusyBox v1.36.1 (2024-01-17 21:57:33 UTC) multi-call binary.

Usage: wget [-cqS] [--spider] [-O FILE] [-o LOGFILE] [--header STR]
           [--post-data STR | --post-file FILE] [-Y on/off]
           [--no-check-certificate] [-P DIR] [-U AGENT] [-T SEC] URL...
           
```

```
Bash  ⌂ ?  { }  ⌂

Retrieve files via HTTP or FTP

--spider      Only check URL existence: $? is 0 if exists
--header STR   Add STR (of form 'header: value') to headers
--post-data STR Send STR using POST method
--post-file FILE   Send FILE using POST method
--no-check-certificate Don't validate the server's certificate
-c           Continue retrieval of aborted transfer
-q           Quiet
-P DIR       Save to DIR (default .)
-S           Show server response
-T SEC        Network read timeout is SEC seconds
-O FILE       Save to FILE ('-' for stdout)
-o LOGFILE    Log messages to FILE
-U STR        Use STR for User-Agent header
-Y on/off     Use proxy

/ # exit
root@vm-24jan-pentaho:/home/praneet.manda# docker rm $(docker ps -aq) --force && docker rmi $(docker images -q) --force
9ea083cd9e4f
2e98ec86000c
Untagged: busybox:latest
Untagged: busybox@sha256:6d9ac9237a84afe1516540f40a0fafdc86859b2141954b4d643af7066d598b74
Deleted: sha256:3f57d9401f8d42f986df300f0c69192fc41da28ccc8d797829467780db3dd741
Deleted: sha256:2e112031b4b923a873c8b3d685d48037e4d5ccd967b658743d93a6e56c3064b9
Untagged: nginx:latest
Untagged: nginx@sha256:4c0fdcaa8b6341bfdeca5f18f7837462c80cff90527ee35ef185571e1c327beac
Deleted: sha256:a8758716bb6aa4d90071160d27028fe4eaee7ce8166221a97d30440c8eac2be6
Deleted: sha256:8cdc90978fea2fcd2db1988760aff35f035a8a0324748925fe07e53cd7804fb4
```

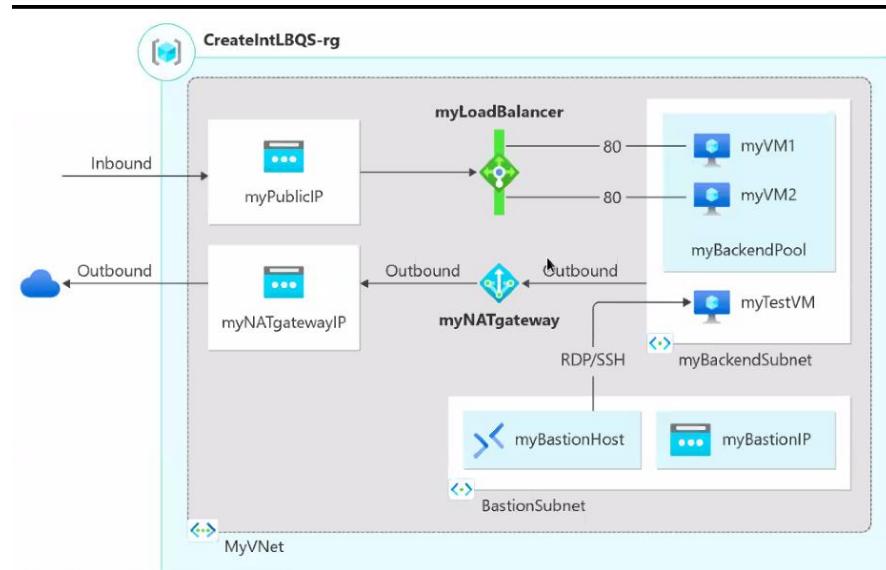
29-01-24

Azure Load Balancer:

APP1: → WEBSERVER → VM(OS) → IP.

Why Azure LB?

- 1) Load Balance internal/external
  - a) Traffic (request or response) to Azure VM. You can manage this using LB.
- 2) Increase availability distribute resources within and across = Zones.
- 3) Configure outbound connectivity to LB resources.
  - a) Internet -> Incoming Traffic -> VM= Inbound.
  - b) VM->Outgoing Traffic->Internet= Outbound.
- 4) Use Health probes to monitor LB resources (VMs).
- 5) Port Forwarding.



RDP: Remote Desktop Protocol.

NIC: Network Interface Card.

NAT is used for complete network but NIC is used for within the virtual machine.

```
1) Subscription → Resource Group  
    # az group create \  
        --name CreatePubLBQS-rg \  
        --location eastus  
2) VNet(10.1.0.0/16) → myPubIP  
    # az network vnet create \  
        --resource-group CreatePubLBQS-rg \  
        --location eastus \  
        --name myVNet \  
        --address-prefixes 10.1.0.0/16 \  
        --subnet-name myBackendSubnet \  
        --subnet-prefixes 10.1.0.0/24
```

3) LB → HP → LB Route

4) NSG → NSG Rule

5) Bastion Host

    PubIP - MyBastionIP  
    Subnet - AzBastionSubnet

6) Backend Subnet:

    Create NIC with name as MyNicVM1, MyNicVM2

    VM1, VM2

    LB ← VM1, VM2 ← MyNicVM1, MyNicVM2

```

1) Subscription → Resource Group
2) VNet(10.1.0.0/16) → myPubIP
3) LB → HP → LB Route
4) NSG → NSG Rule
5) Bastion Host
    PubIP - MyBastionIP
    Subnet -| AzBastionSubnet
6) Backend Subnet:
    Create NIC with name as MyNicVM1, MyNicVM2
    VM1, VM2
    LB ← VM1, VM2 ← MyNicVM1, MyNicVM2
7) NAT Gateway
    MyNATGWIP
    MyNATGW → BackendSubnet

```

8) WebServer (IIS) <-- HelloWorld ← VM1, VM2

9) Test

10) Delete

29<sup>th</sup> JAN:

```

az network vnet create \
--resource-group CreatePubLBQS-rg \
--location eastus \
--name myVNet \
--address-prefixes 10.1.0.0/16 \
--subnet-name myBackendSubnet \
--subnet-prefixes 10.1.0.0/24

```

```

az network public-ip create --resource-group CreatePubLBQS-rg --name myPublicIP --
sku Standard --zone 1

```

**Mujahed Hussaini** 15:59

```

az network lb create \
--resource-group CreatePubLBQS-rg \
--name myLoadBalancer \
--sku Standard \
--public-ip-address myPublicIP \
--frontend-ip-name myFrontEnd \

```

```
--backend-pool-name myBackEndPool
```

Messages addressed to "Meeting Group Chat" will also appear in the meeting group chat in Team Chat

**Mujahed Hussaini** 16:15

```
az network lb probe create \  
--resource-group CreatePubLBQS-rg \  
--lb-name myLBMujahed \  
--name myHealthProbe \  
--protocol tcp \  
--port 80
```

**Mujahed Hussaini** 16:25

```
az network lb rule create \  
--resource-group CreatePubLBQS-rg \  
--lb-name myLBMujahed \  
--name myHTTPRule \  
--protocol tcp --frontend-port 80 \  
--backend-port 80 \  
--frontend-ip-name myFrontEnd \  
--backend-pool-name myBackEndPool \  
--probe-name myHealthProbe \  
--disable-outbound-snat true \  
--idle-timeout 15 --enable-tcp-reset true  
az network nsg create \  
--resource-group CreatePubLBQS-rg \  
--name myNSG
```

**Mujahed Hussaini** 16:31

```
az network nsg rule create \  
--resource-group CreatePubLBQS-rg \  
--nsg-name myNSG \  
--name myNSGRuleHTTP \  
--protocol '*' --direction inbound \  
--source-address-prefix '*' \  
--source-port-range '*' \  
--destination-address-prefix '*' \  
--destination-port-range 80 \  
--access allow --priority 200
```

```
az network bastion create \
--resource-group Create_PUB \
--name myBastionHost \
--public-ip-address myBastionIP \
--vnet-name myVNet \
--location eastus
```

```
Bash

praneet [ ~ ]$ az network public-ip create --resource-group Create_PUB --name myBastionIP --sku Standard --zone 1 2 3
{
  "publicIp": {
    "ddosSettings": {
      "protectionMode": "VirtualNetworkInherited"
    },
    "etag": "W/"5c43cd7b-402a-436c-bf44-9a355d1b8c65"",
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/publicIPAddresses/myBastionIP",
    "idleTimeoutInMinutes": 4,
    "ipAddress": "4.255.88.0",
    "ipTags": [],
    "location": "eastus",
    "name": "myBastionIP",
    "provisioningState": "Succeeded",
    "publicIPAddressVersion": "IPv4",
    "publicIPAllocationMethod": "Static",
    "resourceGroup": "Create_PUB",
    "resourceGuid": "ffc197dd-77ef-4991-889e-7b623f1663ae",
    "sku": {
      "name": "Standard",
      "tier": "Regional"
    },
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": [
      "1",
      "2",
      "3"
    ]
  }
}

praneet [ ~ ]$ az network vnet subnet create \
  --resource-group Create_PUB \
  --name AzureBastionSubnet \
  --vnet-name myVNet \
  --address-prefixes 10.1.1.0/27
{
  "addressPrefix": "10.1.1.0/27",
  "delegations": [],
  "etag": "W/"b82b13ab-a49d-4151-aa9e-59af5879a0ee"",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/virtualNetworks/myVNet/subnets/AzureBastionSubnet",
  "name": "AzureBastionSubnet",
  "privateEndpointNetworkPolicies": "Disabled",
  "privateLinkServiceNetworkPolicies": "Enabled",
  "provisioningState": "Succeeded",
  "resourceGroup": "Create_PUB",
  "type": "Microsoft.Network/virtualNetworks/subnets"
}

praneet [ ~ ]$ az network bastion create \
  --resource-group Create_PUB \
  --name myBastionHost \
  --public-ip-address myBastionIP \
  --vnet-name myVNet \
  --location eastus
The command requires the extension bastion. Do you want to install it now? The command will continue to run after the extension is installed. (Y/n): Y
Run 'az config set extension.use_dynamic_install=yes_without_prompt' to allow installing extensions without prompt.
Default enabled including preview versions for extension installation now. Disabled in May 2024. Use '--allow-preview true' to enable it specifically if needed. Use '--install-stable version only'.
This command is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
[]/ Running ..
```

```

praneet [ ~ ]$ array=(myNicVM1 myNicVM2)
for vmnic in "${array[@]}"
do
    az network nic create \
        --resource-group Create_PUB \
        --name $vmnic \
        --vnet-name myVNet \
        --subnet myBackEndsSubnet \
        --network-security-group myNSG
done
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
  "NewNIC": {
    "auxiliaryMode": "None",
    "auxiliarySku": "None",
    "disableTcpStateTracking": false,
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDomainNameSuffix": "sjy55sjc5jyelkmqggblgcbwe.bx.internal.cloudapp.net"
    },
    "enableAcceleratedNetworking": false,
    "enableIPForwarding": false,
    "etag": "W/"0ecc341d-a564-4f9b-9bf1-e54699e2369e"",
    "hostedWorkloads": [],
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/networkInterfaces/myNicVM1",
    "ipConfigurations": [
      {
        "etag": "W/"0ecc341d-a564-4f9b-9bf1-e54699e2369e"",
        "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/networkInterfaces/myNicVM1/ipConfigurations/ipconfig1",
        "name": "ipconfig1",
      }
    ]
  }
}

praneet [ ~ ]$ az vm create \
    --resource-group CreatePubLBQS-rg \
    --name myVM1 \
    --nics myNicVM1 \
    --image win2019datacenter \
    --admin-username azureuser \
    --zone 1 --no-wait
Admin Password:
Confirm Admin Password:
Passwords do not match.
Admin Password:
Confirm Admin Password:
praneet [ ~ ]$ az network bastion create \
    --resource-group Create_Pub \
    --name myBastionHost \
    --public-ip-address myBastionIP \
    --vnet-name myVNet \
    --location eastus
This command is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
}

praneet [ ~ ]$ az vm create \
    --resource-group Create_PUB \
    --name myVM1 \
    --nics myNicVM1 \
    --image win2019datacenter \
    --admin-username azureuser \
    --zone 1 --no-wait
Admin Password:
Confirm Admin Password:
Passwords do not match.
Admin Password:
Confirm Admin Password:
praneet [ ~ ]$ az vm create \
    --resource-group Create_PUB \
    --name myVM2 \
    --nics myNicVM2 \
    --image win2019datacenter \
    --admin-username azureuser \
    --zone 2 --no-wait
Admin Password:
Confirm Admin Password:
praneet [ ~ ]$ 

```

**Create\_PUB** Resource group

Subscription (move) : [Incedo\\_AZ\\_Training](#) Subscription ID : 59c106e1-c9e3-4440-961d-8bd04fe978a7 Tags (edit) : [Add tags](#)

Deployments : 4 Failed 3 Succeeded Location : East US

**Resources Recommendations**

Showing 1 to 12 of 12 records  Show hidden types [Add filter](#)

Name	Type	Location
myBastionHost	Bastion	East US
myBastionIP	Public IP address	East US
myNicV1	Network interface	East US
myNicV2	Network interface	East US
myNSG	Network security group	East US
myPublicIP	Public IP address	East US
myVM1	Virtual machine	East US
myVM1_disk1_7ee8f70861094e4eb24e371d9ff2c323	Disk	East US
myVM2	Virtual machine	East US
myVM2_OsDisk_1_837db74aab74ac281e457c94277e60e	Disk	East US
myVNet	Virtual network	East US
praneet_lb	Load balancer	East US

**Bash**

```

Requesting a Cloud Shell.Succeeded.
Connecting terminal...

praneet [ ~ ]$ array=(myNicV1 myNicV2)
for vmnic in "${array[@]}"
do
    az network nic ip-config address-pool add \
        --address-pool myBackendPool \
        --ip-config-name ipconfig1 \
        --nic-name $vmnic \
        --resource-group Create_PUB \
        --lb-name praneet_lb
done
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI\_refstatus
{
  "etag": "W/"bde2dd3b-3ae6-485e-9ac6-568e48a73db9"",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/networkInterfaces/myNicV1/ipConfigurations/ipconfig1",
  "loadBalancerBackendAddressPools": [
    {
      "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/loadBalancers/praneet_lb/backendAddressPools/myBackEndPool",
      "resourceGroup": "Create_PUB"
    }
  ],
  "name": "ipconfig1",
  "primary": true,
  "privateIPAddress": "10.1.0.4",
  "privateIPAddressVersion": "IPv4",
  "privateIPAllocationMethod": "Dynamic",
  "provisioningState": "Succeeded",
  "resourceGroup": "Create_PUB",
  "subnet": {
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/virtualNetworks/myVNet/subnets/myBackEndSubnet",
    "resourceGroup": "Create_PUB"
  },
  "type": "Microsoft.Network/networkInterfaces/ipConfigurations"
}
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI\_refstatus
| Running ...

praneet [ ~ ]$ Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI\_refstatus
{
  "etag": "W/"1fcf1eca-5e15-485d-869b-8b26e115f102"",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/networkInterfaces/myNicV2/ipConfigurations/ipconfig1",
  "loadBalancerBackendAddressPools": [
    {
      "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/loadBalancers/praneet_lb/backendAddressPools/myBackEndPool",
      "resourceGroup": "Create_PUB"
    }
  ],
  "name": "ipconfig1",
  "primary": true,
  "privateIPAddress": "10.1.0.5",
  "privateIPAddressVersion": "IPv4",
  "privateIPAllocationMethod": "Dynamic",
  "provisioningState": "Succeeded",
  "resourceGroup": "Create_PUB",
  "subnet": {
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/virtualNetworks/myVNet/subnets/myBackEndSubnet",
    "resourceGroup": "Create_PUB"
  },
  "type": "Microsoft.Network/networkInterfaces/ipConfigurations"
}

```

```

praneet [ ~ ]$ az network public-ip create \
    --resource-group Create_PUB \
    --name myNATgatewayIP \
    --sku Standard \
    --zone 1 2 3
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
  "publicIp": {
    "ddosSettings": {
      "protectionMode": "VirtualNetworkInherited"
    },
    "etag": "W/\\"1e2944fb-077f-4021-bad1-39f2fb90954a\\\"",
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/publicIPAddresses/myNATgatewayIP",
    "idleTimeoutInMinutes": 4,
    "ipAddress": "4.156.36.154",
    "ipTags": [],
    "location": "eastus",
    "name": "myNATgatewayIP",
    "provisioningState": "Succeeded",
    "publicIPAddressVersion": "IPv4",
    "publicIPAllocationMethod": "Static",
    "resourceGroup": "Create_PUB",
    "resourceGuid": "b00b552e-581a-4f0b-9bb6-7e1a6c31f6de",
    "sku": {
      "name": "Standard",
      "tier": "Regional"
    },
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": [
      "1",
      "2",
      "3"
    ]
  }
}
praneet [ ~ ]$ 

praneet [ ~ ]$ az network nat gateway create \
    --resource-group Create_PUB \
    --name myNATgateway \
    --public-ip-addresses myNATgatewayIP \
    --idle-timeout 10
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
  "etag": "W/\\"878a1e50-d4aa-4c38-a95a-21cc0f19f4a8\\\"",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/natGateways/myNATgateway",
  "idleTimeoutInMinutes": 10,
  "location": "eastus",
  "name": "myNATgateway",
  "provisioningState": "Succeeded",
  "publicIPAddresses": [
    {
      "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/publicIPAddresses/myNATgatewayIP",
      "resourceGroup": "Create_PUB"
    }
  ],
  "resourceGroup": "Create_PUB",
  "resourceGuid": "ea424e02-d252-4ef5-8234-305d6c5035e3",
  "sku": {
    "name": "Standard"
  },
  "type": "Microsoft.Network/natGateways"
}
praneet [ ~ ]$ 

praneet [ ~ ]$ az network vnet subnet update \
    --resource-group Create_PUB \
    --vnet-name myVNet \
    --name myBackendSubnet \
    --nat-gateway myNATgateway
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
  "addressPrefix": "10.1.0.0/24",
  "delegations": [],
  "etag": "W/\\"145ae79c-fefc-4488-bcfe-0d03069590b5\\\"",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/virtualNetworks/myVNet/subnets/myBackendSubnet",
  "ipConfigurations": [
    {
      "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/CREATE_PUB/providers/Microsoft.Network/networkInterfaces/MYNICVM1/ipConfigurations/IPCONFIG1",
      "resourceGroup": "CREATE_PUB"
    },
    {
      "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/CREATE_PUB/providers/Microsoft.Network/networkInterfaces/MYNICVM2/ipConfigurations/IPCONFIG1",
      "resourceGroup": "CREATE_PUB"
    }
  ],
  "name": "myBackendSubnet",
  "natGateway": {
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Network/natGateways/myNATgateway",
    "resourceGroup": "Create_PUB"
  },
  "privateEndpointNetworkPolicies": "Disabled",
  "privateLinkServiceNetworkPolicies": "Enabled",
  "provisioningState": "Succeeded",
  "resourceGroup": "Create_PUB",
  "type": "Microsoft.Network/virtualNetworks/subnets"
}

```

```

praneet [ ~ ]$ array=(myVM1 myVM2)
for vm in "${array[@]}"
do
    az vm extension set \
        --publisher Microsoft.Compute \
        --version 1.8 --name CustomScriptExtension \
        --vm-name $vm --resource-group Create_PUB \
        --settings '{"commandToExecute":"powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\\\inetpub\\\\wwwroot\\\\Default.htm\" -Value $($env:computername)"}'
done
{
    "autoUpgradeMinorVersion": true,
    "enableAutomaticUpgrade": null,
    "forceUpdateTag": null,
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Compute/virtualMachines/myVM1/extensions/CustomScriptExtension",
    "instanceView": null,
    "location": "eastus",
    "name": "CustomScriptExtension",
    "protectedSettings": null,
    "protectedSettingsFromKeyVault": null,
    "provisionAfterExtensions": null,
    "provisioningState": "Succeeded",
    "publisher": "Microsoft.Compute",
    "resourceGroup": "Create_PUB",
    "settings": {
        "commandToExecute": "powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\\\inetpub\\\\wwwroot\\\\Default.htm\" -Value $($env:computername)"
    },
    "suppressFailures": null,
    "tags": null,
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "typeHandlerVersion": "1.8",
    "typePropertiesType": "CustomScriptExtension"
}
\ Running ...
}

praneet [ ~ ]$ array=(myVM1 myVM2)
for vm in "${array[@]}"
do
    az vm extension set \
        --publisher Microsoft.Compute \
        --version 1.8 --name CustomScriptExtension \
        --vm-name $vm --resource-group Create_PUB \
        --settings '{"commandToExecute":"powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\\\inetpub\\\\wwwroot\\\\Default.htm\" -Value $($env:computername)"}'
done
{
    "autoUpgradeMinorVersion": true,
    "enableAutomaticUpgrade": null,
    "forceUpdateTag": null,
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Compute/virtualMachines/myVM1/extensions/CustomScriptExtension",
    "instanceView": null,
    "location": "eastus",
    "name": "CustomScriptExtension",
    "protectedSettings": null,
    "protectedSettingsFromKeyVault": null,
    "provisionAfterExtensions": null,
    "provisioningState": "Succeeded",
    "publisher": "Microsoft.Compute",
    "resourceGroup": "Create_PUB",
    "settings": {
        "commandToExecute": "powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\\\inetpub\\\\wwwroot\\\\Default.htm\" -Value $($env:computername)"
    },
    "suppressFailures": null,
    "tags": null,
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "typeHandlerVersion": "1.8",
    "typePropertiesType": "CustomScriptExtension"
}
{
    "autoUpgradeMinorVersion": true,
    "enableAutomaticUpgrade": null,
    "forceUpdateTag": null,
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/Create_PUB/providers/Microsoft.Compute/virtualMachines/myVM2/extensions/CustomScriptExtension",
    "instanceView": null,
    "location": "eastus",
    "name": "CustomScriptExtension",
    "protectedSettings": null,
    "protectedSettingsFromKeyVault": null,
    "provisionAfterExtensions": null,
    "provisioningState": "Succeeded",
    "publisher": "Microsoft.Compute",
    "resourceGroup": "Create_PUB",
    "settings": {
        "commandToExecute": "powershell Add-WindowsFeature Web-Server; powershell Add-Content -Path \"C:\\\\inetpub\\\\wwwroot\\\\Default.htm\" -Value $($env:computername)"
    },
    "suppressFailures": null,
    "tags": null,
}
praneet [ ~ ]$ az network public-ip show \
    --resource-group Create_PUB \
    --name myPublicIP \
    --query ipAddress \
    --output tsv
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI\_refstatus
20.106.239.152
  ↻  ↽  ⚡  Not secure  20.106.239.152  ✖

```

myVM1

## After Deletion of VM1.

The screenshot shows the Azure portal interface for the resource group 'Create\_PUB'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Deployments, Security, Deployment stacks, Policies, Properties, Locks, Cost Management, Cost analysis, Cost alerts (preview), Budgets, Advisor recommendations, Monitoring, Insights (preview), and Alerts. The main content area displays the following details:

**Subscription (move) :** Inedo\_AZ\_Training  
**Subscription ID :** 59c106e1-c9e3-4440-961d-8bd04fe978a7  
**Tags (edit) :** Add tags

**Essentials**  
Deployments : 4 Failed 3 Succeeded  
Location : East US

**Resources**    Recommendations

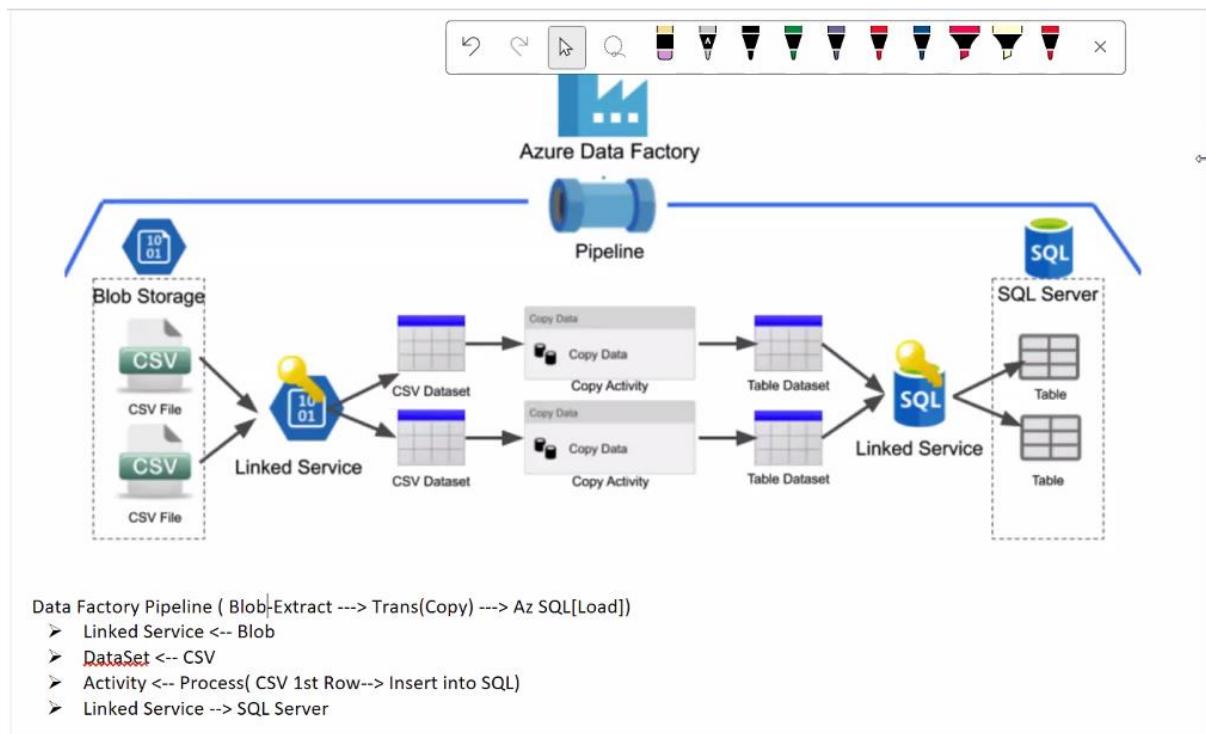
Showing 1 to 13 of 13 records.  Show hidden types

Name	Type	Location
myBastionIP	Public IP address	East US
myNATgateway	NAT gateway	East US
myNATgatewayIP	Public IP address	East US
myNicVM1	Network Interface	East US
myNicVM2	Network Interface	East US
myNSG	Network security group	East US
myPublicIP	Public IP address	East US
myVM1_disk1_7ee8f70861094e4eb24e371d9ff2c323	Disk	East US
myVM2	Virtual machine	East US
myVM2_OsDisk_1_837db74abb74ac281e457c94277e60e	Disk	East US
myVnet	Virtual network	East US
praneet_lb	Load balancer	East US

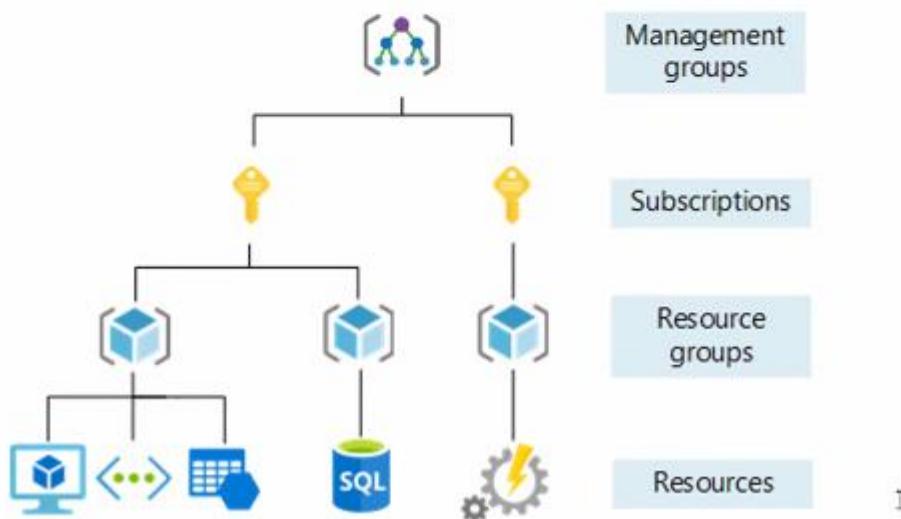
The screenshot shows a browser window with two tabs: 'myNSG - Microsoft Azure' and 'Create\_PUB - Microsoft Azure'. The address bar shows the IP address 20.106.239.152. The content of the page is 'myVM2'.

<https://github.com/NubeEra-Samples/AzureServices>

2<sup>nd</sup> FEB



1. Source(Storage Account --> Container= Inp)
2. Sink(Storage Account --> Container = Output)
3. Create ADF Studio & Pipeline
  - a. Launch Studio
  - b. PIPE LINE
    - i. Manage --> Linked Service
    - ii. Author --> Data Sets(Input & Output)
    - iii. Pipeline --> Activity(Copy) --> Select Source & Sink
    - iv. Publish
    - v. Debug



## Creating data factory

- o Create inside resource group
- o Create one more storage account (remember to heirarchical namespace should be enabled)
- o Create 2 containers inside it (input & output)
- o Input any data inside preferably csv file insite input container
- o Create 3 things (linked service, dataset,
- o Go to adf.azure.com
- ?
- Studio ➔ managed ➔ linked services ➔ azure data lake storage gen2
- ?
- Create the linked service and test the connection as LSinput
- ?
- Create one more as LSoutput
- ?
- Go to author and

The screenshot displays two Azure management interfaces. The top window shows the 'Data Factory' blade for a resource group named 'dfpraneet'. It features a left sidebar with 'Factory Resources' and 'Pipelines' sections, and a central workspace for managing a pipeline named 'pipeline1'. The pipeline contains a single 'Copy data' activity named 'Copy data1', which is currently successful ('Succeeded'). The bottom window shows the 'Containers' blade for an Azure Storage account named 'input'. It lists a single blob named 'industry.csv' with details such as modified time (2/2/2024, 3:50:48 PM), access tier (Hot (Inferred)), and size (749 B). Both windows have a header bar with the Microsoft Azure logo, search bars, and user information.

The screenshot shows the Microsoft Azure Storage Explorer interface. The left sidebar has a tree view with 'Containers' expanded, showing 'output' selected. Under 'output', there are links for 'Overview', 'Diagnose and solve problems', and 'Access Control (IAM)'. The main content area shows a table of blobs. A search bar at the top says 'Search resources, services, and docs (S+J)'. Below it, there are buttons for 'Upload', 'Add Directory', 'Refresh', 'Rename', 'Delete', 'Change tier', 'Acquire lease', 'Break lease', and 'Give feedback'. A section titled 'Authentication method: Access key (Switch to Microsoft Entra user account)' shows 'Location: output'. A search bar below it says 'Search blobs by prefix (case-sensitive)'. A toggle switch is set to 'Show deleted objects'. The table lists one blob:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
industry.csv	2/2/2024, 3:59:59 PM	Hot (Inferred)		Block blob	749 B	Available



The screenshot shows the Microsoft Azure Storage Container list for the 'storepraneeet' account. The left sidebar shows navigation options like Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, and Storage browser. Under Data storage, 'Containers' is selected. The main area displays three containers: '\$logs', 'input', and 'output'. Each container has a checkbox next to its name, indicating it can be deleted. The columns show the container name, last modified date (2/2/2024), anonymous access level (Private), lease state (Available), and three vertical ellipsis buttons for more actions.

Name	Last modified	Anonymous access level	Lease state
\$logs	2/2/2024, 3:48:24 PM	Private	Available
input	2/2/2024, 3:45:14 PM	Private	Available
output	2/2/2024, 3:45:23 PM	Private	Available

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' sidebar lists 'Pipelines' (1 item: 'pipeline1'), 'Change Data Capture (preview)' (0 items), 'Datasets' (2 items), 'Data flows' (0 items), and 'Power Query' (0 items). The main workspace shows a pipeline named 'pipeline1' with activities: 'Copy data' and 'Data flow'. A modal window titled 'Activities' is open, listing other options like 'Synapse', 'Azure Data Explorer', etc. To the right, a 'Notifications' panel displays three successful events: 'Publishing completed' (Successfully published, 2 minutes ago), 'Successfully created' (Successfully created AzureDataStorage\_Output (Linked service), 7 minutes ago), and another 'Successfully created' (Successfully created AzureDataStorage\_Input (Linked service), 7 minutes ago).

VNET 4<sup>TH</sup> FEB:

The image shows two adjacent Bash terminal windows. The top window displays the creation of a resource group named 'rgpraneet' in the 'eastus' location. The bottom window shows the creation of two virtual networks, 'Vnet1' and 'Vnet2', each with a single subnet ('Subnet1' and 'Subnet2' respectively) and specific IP address ranges.

```
Requesting a Cloud Shell. Succeeded.  
Connecting terminal...  
  
praneet [ ~ ]$ az group create --name rgpraneet --location eastus  
{  
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet",  
  "location": "eastus",  
  "managedBy": null,  
  "name": "rgpraneet",  
  "properties": {  
    "provisioningState": "Succeeded"  
  },  
  "tags": null,  
  "type": "Microsoft.Resources/resourceGroups"  
}  
  
praneet [ ~ ]$ az network vnet create --name Vnet1 --resource-group rgpraneet --address-prefixes 10.0.0.0/16 --subnet-name Subnet1 --subnet-prefix 10.0.0.0/24  
{  
  "newVNet": {  
    "addressSpace": {  
      "addressPrefixes": [  
        "10.0.0.0/16"  
      ]  
    },  
    "enableDdosProtection": false,  
    "etag": "W/\\"32201a8f-e915-4db2-b192-df6006707d27\\\"",  
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworks/Vnet1",  
    "location": "eastus",  
    "name": "Vnet1",  
    "provisioningState": "Succeeded",  
    "resourceGroup": "rgpraneet",  
    "resourceGuid": "1fb8f72e-d203-40c8-8b4f-f79d01b39258",  
    "subnets": [  
      {  
        "addressPrefix": "10.0.0.0/24",  
        "delegations": [],  
        "etag": "W/\\"32201a8f-e915-4db2-b192-df6006707d27\\\"",  
        "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworks/Vnet1/subnets/Subnet1",  
        "name": "Subnet1",  
        "privateEndpointNetworkPolicies": "Disabled",  
        "privateLinkServiceNetworkPolicies": "Enabled",  
        "provisioningState": "Succeeded",  
        "resourceGroup": "rgpraneet",  
        "type": "Microsoft.Network/virtualNetworks/subnets"  
      }  
    ]  
  }  
}  
  
praneet [ ~ ]$ az network vnet create --name Vnet2 --resource-group rgpraneet --address-prefixes 10.1.0.0/16 --subnet-name Subnet2 --subnet-prefix 10.1.0.0/24  
{  
  "newVNet": {  
    "addressSpace": {  
      "addressPrefixes": [  
        "10.1.0.0/16"  
      ]  
    },  
    "enableDdosProtection": false,  
    "etag": "W/\\"c44c1678-fdf9-4ae1-8e78-3daab23d6c3a\\\"",  
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworks/Vnet2",  
    "location": "eastus",  
    "name": "Vnet2",  
    "provisioningState": "Succeeded",  
    "resourceGroup": "rgpraneet",  
    "resourceGuid": "ed75c60b-2050-4f76-ab27-c11388d046fe",  
    "subnets": [  
      {  
        "addressPrefix": "10.1.0.0/24",  
        "delegations": [],  
        "etag": "W/\\"c44c1678-fdf9-4ae1-8e78-3daab23d6c3a\\\"",  
        "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworks/Vnet2/subnets/Subnet2",  
        "name": "Subnet2",  
        "privateEndpointNetworkPolicies": "Disabled",  
        "privateLinkServiceNetworkPolicies": "Enabled",  
        "provisioningState": "Succeeded",  
        "resourceGroup": "rgpraneet",  
        "type": "Microsoft.Network/virtualNetworks/subnets"  
      }  
    ],  
    "type": "Microsoft.Network/virtualNetworks",  
    "virtualNetworkPeerings": []  
}  
  
praneet [ ~ ]$ vNet1Id=$(az network vnet show --resource-group rgpraneet --name Vnet1 --query id --out tsv)  
praneet [ ~ ]$ vNet2Id=$(az network vnet show --resource-group rgpraneet --name Vnet2 --query id --out tsv)
```

```

praneet [ ~ ]$ az network vnet peering create --name Vnet1-to-Vnet2 --resource-group rgpraneet --vnet-name Vnet1 --remote-vnet $vNet2Id --allow-vnet-access
{
  "allowForwardedTraffic": false,
  "allowGatewayTransit": false,
  "allowVirtualNetworkAccess": true,
  "doNotVerifyRemoteGateways": false,
  "etag": "W/\\"b6e23f96-caa2-4fe3-9acf-aec9725d7566\\\"",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworkPeerings/Vnet1-to-Vnet2",
  "name": "Vnet1-to-Vnet2",
  "peeringState": "Initiated",
  "peeringSyncLevel": "RemoteNotInSync",
  "provisioningState": "Succeeded",
  "remoteAddressSpace": {
    "addressPrefixes": [
      "10.1.0.0/16"
    ]
  },
  "remoteVirtualNetwork": {
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworks/Vnet2",
    "resourceGroup": "rgpraneet"
  },
  "remoteVirtualNetworkAddressSpace": {
    "addressPrefixes": [
      "10.1.0.0/16"
    ]
  },
  "resourceGroup": "rgpraneet",
  "resourceGuid": "f2cd3125-f253-0fbe-2068-368e8963d4a6",
  "type": "Microsoft.Network/virtualNetworks/virtualNetworkPeerings",
  "useRemoteGateways": false
}
praneet [ ~ ]$ az network vnet peering create --name Vnet2-to-Vnet1 --resource-group rgpraneet --vnet-name Vnet2 --remote-vnet $vNet1Id --allow-vnet-access
{
  "allowForwardedTraffic": false,
  "allowGatewayTransit": false,
  "allowVirtualNetworkAccess": true,
  "doNotVerifyRemoteGateways": false,
  "etag": "W/\\"35f0aa7d-1559-44db-b65b-b629b0b2dda3\\\"",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworkPeerings/Vnet2-to-Vnet1",
  "name": "Vnet2-to-Vnet1",
  "peeringState": "Connected",
  "peeringSyncLevel": "FullyInSync",
  "provisioningState": "Succeeded",
  "remoteAddressSpace": {
    "addressPrefixes": [
      "10.0.0.0/16"
    ]
  },
  "remoteVirtualNetwork": {
    "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Network/virtualNetworks/Vnet1",
    "resourceGroup": "rgpraneet"
  },
  "remoteVirtualNetworkAddressSpace": {
    "addressPrefixes": [
      "10.0.0.0/16"
    ]
  },
  "resourceGroup": "rgpraneet",
  "resourceGuid": "f2cd3125-f253-0fbe-2068-368e8963d4a6",
  "type": "Microsoft.Network/virtualNetworks/virtualNetworkPeerings",
  "useRemoteGateways": false
}
praneet [ ~ ]$ az network vnet peering show --name Vnet1-to-Vnet2 --resource-group rgpraneet --vnet-name Vnet1 --query peeringState
"Connected"
praneet [ ~ ]$ az network vnet peering show --name Vnet1-to-Vnet2 --resource-group rgpraneet --vnet-name Vnet1 --query peeringState
"Connected"
praneet [ ~ ]$ ssh-keygen -t rsa -b 2048 -f ~/.ssh/praneet_lock
Generating public/private rsa key pair.
Created directory '/home/praneet/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/praneet/.ssh/praneet_lock
Your public key has been saved in /home/praneet/.ssh/praneet_lock.pub
The key fingerprint is:
SHA256:UAxjs4HyY+a61Z5CCtwKZlw28Fx0eZXyzBxoXRoAnRQ praneet@SandboxHost-638426161207079816
The key's randomart image is:
+---[RSA 2048]----+
| .B*E+
| o ... *+ . .
| . * o o =
| B B .+ *
| .= o oSB .
| o ..+ . =
| ...+o..
| .o.....
| ... .o
+---[SHA256]----+
praneet [ ~ ]$ 
praneet [ ~ ]$ az vm create --resource-group rgpraneet --name myVm1 --image Ubuntu2204 --public-ip-sku Standard --vnet-name Vnet1 --subnet Subnet1 --ssh-key-value ~/.ssh/praneet_lock.pub
{
  "fqdn": "",
  "id": "/subscriptions/59c106e1-c9e3-4440-961d-8bd04fe978a7/resourceGroups/rgpraneet/providers/Microsoft.Compute/virtualMachines/myVm1",
  "location": "eastus",
  "macAddress": "00-90-3A-98-40-79",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "20.102.85.185",
  "resourceGroup": "rgpraneet",
  "zones": ""
}
praneet [ ~ ]$ 

```

```

praneet@myVm1:~$ ping 10.0.0.4 -c 4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.039 ms

--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3077ms
rtt min/avg/max/mdev = 0.021/0.035/0.041/0.008 ms
connection to 10.0.0.4 closed.

praneet [ ~ ]$ ping 10.0.0.4 -c 4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.

--- 10.0.0.4 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3058ms

```

## POWERSHELL

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\praneet.manda> Write-Host 'Hello, World!'
>> 'Hello, World!' | Write-Host
Hello, World!
Hello, World!
PS C:\Users\praneet.manda> Write-Output 'Hello, World!'
>> 'Hello, World!' | Write-Output
Hello, World!
Hello, World!
PS C:\Users\praneet.manda> $wh = 'Hello, World!' | Write-Host
>> $wo = 'Hello, World!' | Write-Output
>>
>> Get-Variable wh
>>
>> Get-Variable wo
Hello, World!

Name                           Value
----                           -----
wh                            Hello, World!

```

```

> Windows PowerShell
PS C:\Users\praneet.manda> Write-Host 'Hello, World!'
>> 'Hello, World!' | Write-Host
Hello, World!
Hello, World!
PS C:\Users\praneet.manda> Write-Output 'Hello, World!'
>> 'Hello, World!' | Write-Output
Hello, World!
Hello, World!
PS C:\Users\praneet.manda> $wh = 'Hello, World!' | Write-Host
>> $wo = 'Hello, World!' | Write-Output
>>
>> Get-Variable wh
>>
>> Get-Variable wo
Hello, World!

Name          Value
----          -----
wh
wo          Hello, World!

PS C:\Users\praneet.manda> $a = 1337           # System.Int32
>> $b = "Swifty"                 # System.String
>> $c = 31337, "Swifty"         # array of System.Int32, System.String
>> $d = Get-ChildItem C:\Windows # FileInfo and DirectoryInfo types
>> New-Variable -Name e -Value 1337   # System.Int32
PS C:\Users\praneet.manda> $number = "1337"    # The string is converted to an integer.
>> $number.GetType()           # Get the type of the variable.

IsPublic IsSerial Name          BaseType
-----  -----  --          -----
True     True    String        System.Object

PS C:\Users\praneet.manda> [int]$number = 1337
>> $number = "One Thousand, Three Hundred and Thirty-Seven" #This will give an error
Cannot convert value "One Thousand, Three Hundred and Thirty-Seven" to type "System.Int32". Error: "Input string was
not in a correct format."
At line:2 char:1
+ $number = "One Thousand, Three Hundred and Thirty-Seven" #This will ...
+ ~~~~~
+ CategoryInfo          : MetadataError: (:) [], ArgumentTransformationMetadataException
+ FullyQualifiedErrorId : RuntimeException

```

Let's try and change it.

```

> Windows PowerShell
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\praneet.manda> New-Variable -Name myVar -Value 1337 -Option ReadOnly
>> $myVar
1337
PS C:\Users\praneet.manda> $myVar = 31337
Cannot overwrite variable myVar because it is read-only or constant.
At line:1 char:1
+ $myVar = 31337
+ ~~~~~
+ CategoryInfo          : WriteError: (myVar:String) [], SessionStateUnauthorizedAccessException
+ FullyQualifiedErrorId : VariableNotWritable

PS C:\Users\praneet.manda> $myvar           # Should output 1337
>> New-Variable -Name myVar -Value 31337 -Option ReadOnly -Force
>> $myVar       # Should output 31337
1337
31337
PS C:\Users\praneet.manda> 

>>>>
PS C:\Users\praneet.manda> New-Variable -Name myConst -Value "This CANNOT be changed" -Option Constant
>> $myConst
This CANNOT be changed
PS C:\Users\praneet.manda> New-Variable -Name myConst -Value "I'm going to change it" -Option Constant -Force
New-Variable : Cannot overwrite variable myConst because it is read-only or constant.
At line:1 char:1
+ New-Variable -Name myConst -Value "I'm going to change it" -Option Co ...
+ ~~~~~
+ CategoryInfo          : WriteError: (myConst:String) [New-Variable], SessionStateUnauthorizedAccessException
+ FullyQualifiedErrorId : VariableNotWritable,Microsoft.PowerShell.Commands.NewVariableCommand

PS C:\Users\praneet.manda> 

```



```
PS C:\Users\praneet.manda> $value = 5
>>
>> if ($value -gt 1) {
>>     Write-Host "value is greater than 1"
>> }
value is greater than 1
PS C:\Users\praneet.manda> $value = 5
>>
>> if ($value -gt 10) {
>>     Write-Host "value is greater than 10"
>> }
>> else {
>>     Write-Host "value is $value"
>> }
>>
>> if ($value -gt 10) {
>>     Write-Host "value is greater than 10"
>> }
>> elseif ($value -lt 10) {
>>     Write-Host "value is less than 10"
>> }
>> else {
>>     Write-Host "value is 10"
>> }
value is 5
value is less than 10
PS C:\Users\praneet.manda>
```

```

PS C:\Users\praneet.manda> switch ((Get-Date).Day) {
>>     { $_ -le 10 } { "Day of the month is lower than 10" }
>>     { $_ -gt 10 -and $_ -le 25 } { "Day of the month is between 10 and 25" }
>>     { $_ -gt 25 } { "Day of the month is greater than 25" }
>> }
Day of the month is lower than 10
PS C:\Users\praneet.manda> $month = 3
>>
>> if ($month -eq 1) { Write-Host "January" }
>> elseif ($month -eq 2) { Write-Host "February" }
>> elseif ($month -eq 3) { Write-Host "March" }
>> elseif ($month -eq 4) { Write-Host "April" }
>> elseif ($month -eq 5) { Write-Host "May" }
>> elseif ($month -eq 6) { Write-Host "June" }
>> elseif ($month -eq 7) { Write-Host "July" }
>> elseif ($month -eq 8) { Write-Host "August" }
>> elseif ($month -eq 9) { Write-Host "September" }
>> elseif ($month -eq 10) { Write-Host "October" }
>> elseif ($month -eq 11) { Write-Host "November" }
>> elseif ($month -eq 12) { Write-Host "December" }
>> else { Write-Host "Invalid month" }
>>
>> # Instead we can write the above as
>> switch ($month) {
>>     1 { Write-Host "January" }
>>     2 { Write-Host "February" }
>>     3 { Write-Host "March" }
>>     4 { Write-Host "April" }
>>     5 { Write-Host "May" }
>>     6 { Write-Host "June" }
>>     7 { Write-Host "July" }
>>     8 { Write-Host "August" }
>>     9 { Write-Host "September" }
>>     10 { Write-Host "October" }
>>     11 { Write-Host "November" }
>>     12 { Write-Host "December" }
>>     default { Write-Host "Invalid month" }
>> }
March
March
PS C:\Users\praneet.manda> ■

```

```

value is less than 10
PS C:\Users\praneet.manda> $msg = "Error, the action failed"
>> switch -Wildcard ($msg) {
>>     "Error*" { "Action error" }
>>     "Warning*" { "Action warning" }
>>     "Successful*" { "Action succesfull" }
>> }
>>
>> ## Or use it in the conditions
>> $msg = "Error, the action failed"
>> switch ($msg) {
>>     { $_ -like "Error*" } { "Action error" }
>>     { $_ -like "Warning*" } { "Action warning" }
>>     { $_ -like "Successful*" } { "Action succesfull" }
>> }
Action error
Action error
PS C:\Users\praneet.manda> switch ((Get-Date).Day) {
>>     { $_ -le 10 } { "Day of the month is lower than 10" }
>>     { $_ -gt 10 -and $_ -le 25 } { "Day of the month is between 10 and 25" }
>>     { $_ -gt 25 } { "Day of the month is greater than 25" }
>> }
Day of the month is lower than 10
PS C:\Users\praneet.manda> ■

```

## Windows PowerShell

```
March
PS C:\Users\praneet.manda> $values = @("One", "Two", "Three", "Four", "Five")
>> $values
>> $values.GetType()
One
Two
Three
Four
Five

IsPublic IsSerial Name                                     BaseType
-----  -----  -----
True     True    Object[]                                System.Array

PS C:\Users\praneet.manda> $values = "Six", "Seven", "Eight", "Nine", "10"
>> $values
Six
Seven
Eight
Nine
10
PS C:\Users\praneet.manda> [int[]]$values = 6, 7, 8, 9, 10
>> $values
6
7
8
9
10
PS C:\Users\praneet.manda> [array]$values = 11, 12, 13, 14, 15
>> $values
11
12
13
14
15
PS C:\Users\praneet.manda> $values = @("One", "Two", "Three")
>> $values
>>
>> # Counting the items in the array using the Count property
>> Write-Host "Items in array $($values.Count)"
>>
>> # Add a value to the array using the + operator
>> $values += "Four"
>> $values
>>
>> Write-Host "Items in array $($values.Count)"
>>
>> # Change a value in the array using the index
```

```
➤ Windows PowerShell
PS C:\Users\praneet.manda> $values = @("One", "Two", "Three")
>> $values
>>
>> # Counting the items in the array using the Count property
>> Write-Host "Items in array $($values.Count)"
>>
>> # Add a value to the array using the + operator
>> $values += "Four"
>> $values
>>
>> Write-Host "Items in array $($values.Count)"
>>
>> # Change a value in the array using the index
>> $values[0] = "Five"
>> $values
One
Two
Three
Items in array 3
One
Two
Three
Four
Items in array 4
Five
Two
Three
Four
PS C:\Users\praneet.manda>
PS C:\Users\praneet.manda> [array]$values = 1, 2, 3, 4, 5
>>
>> # Access the third item in the array
>> Write-Host "Item at index 2: $($values[2])"
Item at index 2: 3
PS C:\Users\praneet.manda> $nameArray = @("Erik", "Penny", "Randy", "Sandy", "Toby", "Uma", "Vicky", "Will", "Xavier", "Yvette", "Zach")
>> for ($i = 0; $i -lt $nameArray.Length; $i++) {
>>     Write-Host $nameArray[$i]
>> }
Erik
Penny
Randy
Sandy
Toby
Uma
Vicky
Will
Xavier
Yvette
Zach
PS C:\Users\praneet.manda> $nameArray = @("Erik", "Penny", "Randy", "Sandy", "Toby", "Uma", "Vicky", "Will", "Xavier", "Yvette", "Zach")
>> for ($i = 0; $i -lt $nameArray.Length; $i++) {
>>     Write-Host $nameArray[$i]
>> }
```

```
➤ Windows PowerShell
PS C:\Users\praneet.manda> $array = @(1, 2, (1, 2, 3), 3, 4, (10, 11, 12), 5)
>> $array[0]
>> $array[1]
>> $array[2]
>> $array[2][0]
>> $array[2][1]
>> $array[5]
1
2
1
2
3
1
2
10
11
12
PS C:\Users\praneet.manda> $array = New-Object 'object[,]' 5,8
>> $array[2,5] = 'Hello'
>> $array[3,7] = 'World!'
>> $array
Hello
World!
PS C:\Users\praneet.manda>
```

## Windows PowerShell

```
PS C:\Users\praneet.manda> $employees = @{}
PS C:\Users\praneet.manda> $employees = @{}
>>
>> # Adding values using integers
>> $employees.Add(1, "John")
>> $employees.Add(2, "Mary")
>> $employees.Add(3, "Bob")
>> $employees.Add(4, "Sam")
>>
>> $address = @{}
>>
>> # Adding values using strings
>> $address.Add("John", "123 Main Street")
>> $address.Add("Mary", "456 North Street")
>> $address.Add("Bob", "789 West Street")
>> $address.Add("Sam", "321 South Street")
>>
>> # Creating the hashtable in one go with values
>> $zipCodes = @{
>>     "John" = "12345"
>>     "Mary" = "54321"
>>     "Bob" = "98765"
>>     "Sam" = "32145"
>> }
PS C:\Users\praneet.manda> $employees[4]
>> $address["Mary"]
>> $zipCodes.Sam
Sam
456 North Street
32145
PS C:\Users\praneet.manda> $employees.keys | Sort-Object $_ | ForEach-Object {
>>     Write-Host "Employee ID $_ : $($employees[$_])"
>> }
>>
>> foreach ($key in $address.Keys) {
>>     Write-Host "$($key) lives at $($address[$key])"
>> }
Employee ID 1 : John
Employee ID 2 : Mary
Employee ID 3 : Bob
Employee ID 4 : Sam
Bob lives at 789 West Street
John lives at 123 Main Street
Sam lives at 321 South Street
Mary lives at 456 North Street
PS C:\Users\praneet.manda> $employeeAddress = @{
>>     Name = "Mary"
>>     Address = "456 North Street"
>>     Zipcode = "54321"
>> }
>>
>> $employeeAddress
Name           Value
----           -----
Name           Mary
Zipcode        54321
Address        456 North Street
```

```
PS C:\Users\praneet.manda> $addresses = @()
>>
>> $addresses += [ordered]@{Name = "John"; Address = "123 Main Street" }
>> $addresses += [ordered]@{Name = "Sam"; Address = "321 South Street" }
>>
>> $addresses += @{Name = "Mary"; Address = "456 North Street" }
>> $addresses += @{Name = "Bob"; Address = "789 West Street" }
>>
>> $addresses

Name                Value
----              -----
Name                John
Address             123 Main Street
Name                Sam
Address             321 South Street
Name                Mary
Address             456 North Street
Name                Bob
Address             789 West Street

PS C:\Users\praneet.manda> $addresses = @()
>>
>> $addresses += @{Name = "John"; Address = "123 Main Street" }
>> $addresses += @{
>>   Name = "Sam"
>>   Address = "321 South Street"
>> }
>> $addresses += @{Name = "Mary"; Address = "456 North Street" }
>> $addresses += @{Name = "Bob"; Address = "789 West Street" }
```

```
PS C:\Users\praneet.manda> function writeHelloWorld() {
>>     Write-Host "Hello World!"
>> }
>>
>> writeHelloWorld
Hello World!
PS C:\Users\praneet.manda> function writeMessage {
>>     param(
>>         [string]$Message
>>     )
>>
>>     Write-Host "Message: $Message"
>> }
>>
>> writeMessage "Hello World!"
>> writeMessage -message "Who is there?"
Message: Hello World!
Message: Who is there?
PS C:\Users\praneet.manda> function writeMessage {
>>     param(
>>         [Parameter(Mandatory = $true, Position = 1, HelpMessage = "The message to write")]
>>         [string]$Message
>>     )
>>
>>     process {
>>         Write-Host "Message: $Message"
>>     }
>> }
>>
>> writeMessage "Hello World!"
>> writeMessage
Message: Hello World!

cmdlet writeMessage at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Message: function writeMessage {
Message: function writeMessage {
PS C:\Users\praneet.manda>     [CmdLetBinding()]
>>     param(
>>         [Parameter(Mandatory = $true, Position = 1, HelpMessage = "The message to write")]
>>         [string]$Message
>>     )

cmdlet  at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Message: Praneet
PS C:\Users\praneet.manda>
PS C:\Users\praneet.manda>     begin {
>>         Write-Verbose "Beginning of script"
>>         if (($null -eq $Message) -or ($Message -eq "")) {
```

## Windows PowerShell

```
PS C:\Users\praneet.manda> $var = "bla"
>>
>> $var
bla
PS C:\Users\praneet.manda> $global:varOne = "bla"                                # Assign a variable in the global scope
>>
>> Write-Host "Variable One:" $global:varOne          # Print the variable
>>
>> # Function to demonstrate local and global scope
>> function MyFunc() {
>>     $global:varOne = "bla bla"
>>     $varTwo = "boo"
>>     return $varTwo
>> }
>>
>> Write-Host "Variable Two:" $varTwo          # Print the variable
>>
>> $varTwo = MyFunc                                # Call the function and change the variable
>> Write-Host "Variable One:" $varOne            # Print the variable
>>
>> Write-Host "Variable Two:" $varTwo          # Print the variable
Variable One: bla
Variable Two:
Variable One: bla bla
Variable Two: boo
PS C:\Users\praneet.manda> # Make sure that the variable is gone
>> Remove-Variable -Name var1 -ErrorAction SilentlyContinue
>>
>> $var1 = "This is a variable"
>>
>> Write-Host "Var1 = '$var1'"
>>
>> function test1 {
>>     Write-Host "Inside function, var1 = $var1"
>> }
>>
>> test1
>>
>> # Now let's do it privately
>> Remove-Variable -Name var1 -ErrorAction SilentlyContinue
>>
>> $Private:var1 = "This is a variable"
>>
>> function test2 {
>>     Write-Host "Inside function with private, var1 = $var1"
>> }
>>
>> test2
var1 = 'This is a variable'
Inside function, var1 = This is a variable
Inside function with private, var1 =
PS C:\Users\praneet.manda> function myFunc {
>>
>>     $Script:VarOne = "Script Scoped"
>>     $Var2 = "Function Scoped"
>> }
>>
>> myFunc
>> Write-Host "Var 1: $VarOne"
>> Write-Host "Var 2: $Var2"
Var 1: Script Scoped
Var 2:
PS C:\Users\praneet.manda> -
```

## Select Windows PowerShell

```
?S C:\Users\praneet.manda> $var = "Hello World!"  
>>  
>> "Lorem ipsum dolor sit amet..."  
>> $value1 = "Ut enim ad minim veniam... $var"  
>> $value2 = "Duis aute irure dolor in... $var"  
>> [string]$value3 = "Excepteur sint occaecat cupidatat non proident..."  
>>  
>> $value1, $value2, $value3  
>> write-host $value1, $value2, $value3  
>> write-host $value1 $value2 $value3  
Lorem ipsum dolor sit amet...  
Ut enim ad minim veniam... Hello World!  
Duis aute irure dolor in... $var  
Excepteur sint occaecat cupidatat non proident...  
Ut enim ad minim veniam... Hello World! Duis aute irure dolor in... $var Excepteur sint occaecat cupidatat non proident...  
Ut enim ad minim veniam... Hello World! Duis aute irure dolor in... $var Excepteur sint occaecat cupidatat non proident...  
?S C:\Users\praneet.manda> $firstName = "John"  
>> $middleName = "Hubert"  
>> $lastName = "Doe"  
>>  
>> $fullName = $firstName + ' ' + $middleName + ' ' + $lastName  
>> $fullName  
>>  
>> $fullName = "$firstName $middleName $lastName"  
>> $fullName  
John Hubert Doe  
John Hubert Doe  
?S C:\Users\praneet.manda> # Join array  
>> $list = @('a', 'b', 'c', 'd', 'e')  
>> $list -join ','  
>>  
>> # Join array without declaring variable  
>> 'f', 'g', 'h', 'i', 'j' -join '-'  
>>  
>> # Join array without separator  
>> -join ('k', 'l', 'm', 'n', 'o')  
>>  
>> # Using .Net string.Join  
>> [string]::Join('|', 'p', 'q', 'r', 's', 't')  
a,b,c,d,e  
f-g-h-i-j  
k|l|m|n|o|p|q|r|s|t  
?S C:\Users\praneet.manda> $string1 = "a,b,c,d,e"  
>> $string2 = "f-g-h-i-j"  
>> $string3 = 'monday,tuesday,wednesday,thursday,friday,saturday,sunday'  
>> $string4 = 'one1two2three3four4five5six6seven7'
```

```
PS C:\Users\praneet.manda> $string1 = "a,b,c,d,e"
>> $string2 = "f-g-h-i-j"
>> $string3 = 'monday,tuesday,wednesday,thursday,friday,saturday,sunday'
>> $string4 = 'one1two2three3four4five5six6seven7'
>>
>> # Splitting based on comma with .Split()
>> $result1 = $string1.split(",")
>> Write-Host "Count of items split: " $result1.Count
>> Write-Host $result1
>> ""
>>
>> # Splitting based on hyphen with -split()
>> $result2 = $string2 -split "-"
>> Write-Host "Count of items split: " $result2.Count
>> ""
>>
>> # Splitting based on a part of the string with -split
>> $string3 -split "day"
>>
>> # Split the string based on a try/catch block
>> # and limit the amount of items returned (3)
>> $string4 -split {
>>     try {
>>         [int]$_ -gt 1
>>     }
>>     catch {
>>         # Just ignore it
>>     }
>> }, 3
Count of items split: 5
a b c d e

Count of items split: 5

mon
,tues
,wednes
,thurs
,fri
,satur
,sun

one1two
three
four4five5six6seven7
```

```

PS C:\Users\praneet.manda> $firstName = "John"
>> $middleName = "Hubert"
>> $lastName = "Doe"
>>
>> # Format the names
>> "{0} {1} {2}" -f $firstName, $middleName, $lastName
>>
>> # Format the names but also do something else
>> "{0}.{1}@awesomedcorp.com" -f $firstName.Substring(0, 1), $lastName
John Hubert Doe
J.Doe@awesomedcorp.com
PS C:\Users\praneet.manda> # Create a simple object
>> $person = @{
>>     FirstName = 'John'
>>     LastName = 'Doe'
>> }
>> # Let's see what we have
>> $person
>>
>> # Use the object by referencing a property
>> Write-Host "`nHello, " $person.FirstName
>>
>> # Setup the full name of the person using the properties
>> $fullName = $person.FirstName + ' ' + $person.LastName
>> Write-Host "`nHello, " $fullName
>>
>> # Try to add the properties to a string
>> $fullName = "$person.FirstName $person.LastName"
>> Write-Host "`nHello, " $fullName # This will fail
>> # Let's add them the proper way
>> $fullName = "$($person.FirstName) $($person.LastName)"
>> Write-Host "Hello, " $fullName # This will work

Name                Value
----              -----
LastName            Doe
FirstName           John

Hello, John

Hello, John Doe

Hello, System.Collections.Hashtable.FirstName System.Collections.Hashtable.LastName
Hello, John Doe

```

```

PS C:\Users\praneet.manda> # Setup the query
>> $query = "SELECT * FROM [_SCHEMANAME_].[_TABLENAME_] WHERE id = _ID_";
>>
>> # Replace the templated values using the .Replace() method
>> $query = $query.Replace("_SCHEMANAME_", "dbo");
>> $query = $query.Replace("_TABLENAME_", "tbl_test");
>> $query = $query.Replace("_ID_", "1");
>> $query
>>
>> # Replace the delimiter for the CSV using the -replace operator
>> $csv = "a,b,c,d,e,f,g,h,i"
>> $csv -replace ",", ";"
SELECT * FROM [dbo].[tbl_test] WHERE id = 1
a;b;c;d;e;f;g;h;i
PS C:\Users\praneet.manda>

```

```
Windows PowerShell
Get-Item : Cannot find path 'C:\Users\PRANEE~1.MAN\AppData\Local\Temp\System.Diagnostics.Process (powershell)-0.txt' because it does not exist.
At line:12 char:5
+   Get-Item -Path $_
+   ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\PRANEE...wershell)-0.txt:String) [Get-Item], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetItemCommand

Get-Item : Cannot find path 'C:\Users\PRANEE~1.MAN\AppData\Local\Temp\System.Diagnostics.Process (powershell)-1.txt' because it does not exist.
At line:12 char:5
+   Get-Item -Path $_
+   ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\PRANEE...wershell)-1.txt:String) [Get-Item], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetItemCommand

Get-Item : Cannot find path 'C:\Users\PRANEE~1.MAN\AppData\Local\Temp\System.Diagnostics.Process (powershell)-2.txt' because it does not exist.
At line:12 char:5
+   Get-Item -Path $_
+   ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\PRANEE...wershell)-2.txt:String) [Get-Item], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetItemCommand

Get-Item : Cannot find path 'C:\Users\PRANEE~1.MAN\AppData\Local\Temp\System.Diagnostics.Process (powershell)-3.txt' because it does not exist.
At line:12 char:5
+   Get-Item -Path $_
+   ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\PRANEE...wershell)-3.txt:String) [Get-Item], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetItemCommand

PS C:\Users\praneet.manda>
```

```
Windows PowerShell
Get-Item : Cannot find path 'C:\Users\PRANEE~1.MAN\AppData\Local\Temp\System.Diagnostics.Process (powershell)-3.txt' because it does not exist.
At line:12 char:5
+   Get-Item -Path $_ -ErrorAction SilentlyContinue
+   ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\PRANEE...wershell)-3.txt:String) [Get-Item], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetItemCommand

PS C:\Users\praneet.manda>

Windows PowerShell
Confirm
Cannot find path 'C:\Users\PRANEE~1.MAN\AppData\Local\Temp\System.Diagnostics.Process (powershell)-0.txt' because it does not exist.
[Y] Yes  [A] Yes to All  [H] Halt Command  [S] Suspend  [?] Help (default is "Y"):
```

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

Loading personal and system profiles took 752ms.

```
PS C:\Users\praneet.manda> try {
>>     # This will generate an error
>>     1/0
>>     Write-Host "This is executed after the error"
>> } catch {
>>     # Catch all errors
>>     Write-Host "Oh oh! Error occurred.`n$@"
>> }
Oh oh! Error occurred.
Attempted to divide by zero.
PS C:\Users\praneet.manda> try {
>>     # This will generate an error
>>     1/0
>>     Write-Host "This is executed after the error"
>> } catch [System.DivideByZeroException] {
>>     # Catch all errors
>>     Write-Host "Divide by zero error occurred.`n$@"
>> } catch {
>>     # Catch all errors
>>     Write-Host "Oh oh! Another error occurred.`n$@"
>> }
Divide by zero error occurred.
Attempted to divide by zero.
PS C:\Users\praneet.manda> try {
>>     # This will generate an error
>>     1/0
>>     Write-Host "This is executed after the error"
>> } catch [System.DivideByZeroException] {
>>     # Catch all errors
>>     Write-Host "Divide by zero error occurred.`n$@"
>> } catch {
>>     # Catch all errors
>>     Write-Host "Oh oh! Another error occurred.`n$@"
>> } finally {
>>     Write-Host "Finally!"
>> }
Divide by zero error occurred.
Attempted to divide by zero.
Finally!
```

PS C:\Users\praneet.manda>

```
Windows PowerShell
PS C:\Users\praneet.manda> $url = "https://gist.githubusercontent.com/sanderstad/7b9593ff7f3abb9ff17f9026c74ed9c68/raw/d4406c4chb427e15fc0d6d92fbfcf3c72a1e70a/samplefile1.txt"
>>> $filePath = Join-Path -Path $env:temp -ChildPath "samplefile1.txt"
>>>
>>> Invoke-WebRequest -Uri $url -Outfile $filePath
>>> $url = "https://gist.githubusercontent.com/sanderstad/f59996889fc3ec794d325ad2162648f8/raw/5353480009bd714f9764a093b52f0fabff1078fd/samplefile2.csv"
>>> $filePath = Join-Path -Path $env:temp -ChildPath "samplefile2.txt"
>>>
>>> Invoke-WebRequest -Uri $url -Outfile $filePath
PS C:\Users\praneet.manda> $content = Get-Content -Path (Join-Path -Path $env:temp -ChildPath "samplefile1.txt")
Utilitatis causa amicitia est quae sit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. collatio igitur ista te nihil iuvat. Honesta oratio, Socratica, Platonis etiam. Primum in nostrane potestate est, quid meminerimus?
Duo Reges: constructio interrete. Quid, si etiam lucunda memoria est praeteritorum malorum? Si quidem, inquit, tollerem, sed relinquo. An nisi populari fama?
Quamquam id quidem licet iis existimare, qui legerint. Summum a vobis bonum voluptas dicitur. At hoc in eo M. Refert tamen, quo modo. Quid sequatur, quid repugnet, vident. Iam id ipsum ab
surdum, maximum malum neglegi.
PS C:\Users\praneet.manda> $content = Get-Content -Path (Join-Path -Path $env:temp -ChildPath "samplefile1.txt")
>>>
>>> $content.GetType()
>>> $content.Count
IsPublic IsSerial Name                                     BaseType
----- ----- -----
True     True    Object[]                                System.Array
4

PS C:\Users\praneet.manda> $content = Get-Content -Path (Join-Path -Path $env:temp -ChildPath "samplefile1.txt")
>>>
>>> $content | Select-Object -First 2
Utilitatis causa amicitia est quae sit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. collatio igitur ista te nihil iuvat. Honesta oratio, Socratica, Platonis etiam. Primum in nostrane potestate est, quid meminerimus?
Duo Reges: constructio interrete. Quid, si etiam lucunda memoria est praeteritorum malorum? Si quidem, inquit, tollerem, sed relinquo. An nisi populari fama?
PS C:\Users\praneet.manda> Import-Csv -Path (Join-Path -Path $env:temp -ChildPath "samplefile2.txt")

Month : May
Average : 0.1
2005 : 0
2006 : 0
2007 : 1
2008 : 1
2009 : 0
2010 : 0
2011 : 0
2012 : 2
2013 : 0
```

```
Windows PowerShell
This is just a test
This is just another test
PS C:\Users\praneet.manda> ^C
PS C:\Users\praneet.manda> Add-Content -Path (Join-Path -Path $env:TEMP -ChildPath "test2.txt") -Value "Test1"
>> Add-Content -Path (Join-Path -Path $env:TEMP -ChildPath "test2.txt") -Value "Test2" -NoNewline
>> Add-Content -Path (Join-Path -Path $env:TEMP -ChildPath "test2.txt") -Value "Test3" -NoNewline
PS C:\Users\praneet.manda> "Lorem ipsum dolor sit amet, consectetur adipiscing elit" | Out-File -FilePath c:\temp\output1.txt
>> Get-Content C:\temp\output1.txt
Out-File : Could not find a part of the path 'c:\temp\output1.txt'.
At line:1 char:61
+ ... consectetur adipiscing elit" | Out-File -FilePath c:\temp\output1.txt
+
+ CategoryInfo          : OpenError: () [Out-File], DirectoryNotFoundException
+ FullyQualifiedErrorId : FileOpenFailure,Microsoft.PowerShell.Commands.OutFileCommand

Set-Content : Cannot find path 'C:\temp\output1.txt' because it does not exist.
At line:2 char:1
+ Get-Content C:\temp\output1.txt
+
+ CategoryInfo          : ObjectNotFound: (C:\temp\output1.txt:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetContentCommand

PS C:\Users\praneet.manda> "Lorem ipsum dolor sit amet, consectetur adipiscing elit" | Out-File -FilePath c:\temp\output2.txt
>> "Ut enim ad minim veniam, quis nostrud" | Out-File -FilePath c:\temp\output2.txt -Append
>> Get-Content C:\temp\output2.txt
Out-File : Could not find a part of the path 'c:\temp\output2.txt'.
At line:1 char:61
+ ... consectetur adipiscing elit" | Out-File -FilePath c:\temp\output2.txt
+
+ CategoryInfo          : OpenError: () [Out-File], DirectoryNotFoundException
+ FullyQualifiedErrorId : FileOpenFailure,Microsoft.PowerShell.Commands.OutFileCommand

Out-File : Could not find a part of the path 'C:\temp\output2.txt'.
At line:2 char:43
+ ... enim, quis nostrud" | Out-File -FilePath c:\temp\output2.txt -Append
+
+ CategoryInfo          : OpenError: () [Out-File], DirectoryNotFoundException
+ FullyQualifiedErrorId : FileOpenFailure,Microsoft.PowerShell.Commands.OutFileCommand

Get-Content : Cannot find path 'C:\temp\output2.txt' because it does not exist.
At line:3 char:1
+ Get-Content C:\temp\output2.txt
+
+ CategoryInfo          : ObjectNotFound: (C:\temp\output2.txt:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetContentCommand

PS C:\Users\praneet.manda>
```

## Windows PowerShell

```
PS C:\Users\praneet.manda> Get-Date  
05 February 2024 11:36:35  
  
PS C:\Users\praneet.manda> $today = Get-Date  
>> Write-Host "Full Date: `t$today"  
>> Write-Host "Date: `t`t $($today.Date)"  
>> Write-Host "Year: `t`t $($today.Year)"  
>> Write-Host "Month: `t`t $($today.Month)"  
>> Write-Host "Day: `t`t $($today.Day)"  
>> Write-Host "Day of the Week: $($today.DayOfWeek)"  
>> Write-Host "Day of the Year: $($today.DayOfYear)"  
>> Write-Host "Hour: `t`t $($today.Hour)"  
>> Write-Host "Minute: `t`t $($today.Minute)"  
>> Write-Host "Second: `t`t $($today.Second)"  
>> Write-Host "Millisecond: $($today.Millisecond)"  
Full Date: 02/05/2024 11:36:40  
Date: 02/05/2024 00:00:00  
Year: 2024  
Month: 2  
Day: 5  
Day of the Week: Monday  
Day of the Year: 36  
Hour: 11  
Minute: 36  
Second: 40  
Millisecond: 688  
PS C:\Users\praneet.manda> $date = Get-Date  
>> $date  
>> $date -f "yyyyMMdd"  
  
05 February 2024 11:36:46  
02/05/2024 11:36:46  
  
PS C:\Users\praneet.manda> $today = Get-Date  
>> $yesterday = $today.AddDays(-1)  
>> $tomorrow = $today.AddDays(1)  
>> Write-Host "Today: `t`t$today"  
>> Write-Host "Yesterday: `t`t$yesterday"  
>> Write-Host "Tomorrow: `t`t$tomorrow"  
Today: 02/05/2024 11:36:52  
Yesterday: 02/04/2024 11:36:52  
Tomorrow: 02/06/2024 11:36:52  
PS C:\Users\praneet.manda> ■
```

```
PS C:\Users\praneet.manda> $date = Get-Date -Date "2022-01-02 03:04:05"
>> $date
02 January 2022 03:04:05

PS C:\Users\praneet.manda> # Assume with this example that the current date is 2022-03-14
>> $date = get-date -Month 5
>> $date
05 May 2024 11:37:58

PS C:\Users\praneet.manda>
```

```
Windows PowerShell
PS C:\Users\praneet.manda> $names = @("Muffin", "Romeo", "Noodle", "Zoe", "Jack", "Luna", "Gracie", "mittens", "Phoebe", "Peanut", "Harley", "Jake")
>>
>> $names | Sort-Object
Gracie
Harley
Jack
Jake
Luna
mittens
Muffin
Noodle
Peanut
Phoebe
Romeo
Zoe
PS C:\Users\praneet.manda> $names = @("Muffin", "Romeo", "Noodle", "zoe", "Jack", "Luna", "Gracie", "mittens", "Phoebe", "Peanut", "Harley", "Jake")
>>
>> $names | Sort-Object -Descending
Zoe
Romeo
Phoebe
Peanut
Noodle
Muffin
mittens
Luna
Jake
Jack
Harley
Gracie
PS C:\Users\praneet.manda> $names = @("Muffin", "muffin", "Noodle", "Zoe", "zoe", "Luna", "Gracie", "peanut", "Phoebe", "Peanut", "Harley", "jake")
>>
>> $names | Sort-Object -CaseSensitive
Gracie
Harley
Jake
Luna
muffin
Muffin
Noodle
peanut
Peanut
Phoebe
zoe
zoe
PS C:\Users\praneet.manda>
```

```

PS C:\Users\praneet.manda> $names = @("Muffin","muffin","Noodle","Zoe","zoe","Luna","Gracie","peanut","Phoebe","Peanut","Harley","Jake")
>>
>> $names | Sort-Object -Unique
Gracie
Harley
Jake
Luna
muffin
Noodle
Peanut
Phoebe
zoe
PS C:\Users\praneet.manda> $files = Get-ChildItem -Path $env:TEMP
>> $files | Sort-Object -Property LastWriteTime | Where-Object{$_.PSIsContainer -eq $false}

Directory: C:\Users\praneet.manda\AppData\Local\Temp

Mode                LastWriteTime      Length Name
----                -----        ---- 
----- 22-11-2021    01:48        42164600 wctB953.tmp
----- 17-09-2022    05:35        64854   wct10DA.tmp
----- 17-09-2022    05:35        64854   wct9E3C.tmp
----- 17-09-2022    05:35        64854   wct9E2D.tmp
----- 17-09-2022    05:35        64854   wctF586.tmp
----- 02-12-2023    01:14        65858576 wctC3D7.tmp
----- 15-12-2023    21:33         899   wctFF7C.tmp
----- 15-12-2023    21:33         899   wctAF83.tmp
----- 15-12-2023    21:33         899   wctBABE.tmp
----- 15-12-2023    21:33         899   wct89EE.tmp
----- 15-12-2023    21:33         899   wct91C9.tmp
----- 15-12-2023    21:33         899   wctFE55.tmp
----- 15-12-2023    21:33         899   wctE195.tmp
----- 15-12-2023    21:33         899   wctC9D1.tmp
-a--- 15-12-2023    21:33         899   wctCEA4.tmp
-a--- 15-12-2023    21:33         899   wctC00E.tmp
-a--- 15-12-2023    21:33         899   wctC0B9.tmp
-a--- 15-12-2023    21:33         899   wctE364.tmp

```

## Windows PowerShell

```

PS C:\Users\praneet.manda> # Old style of creating an object
>> $object1 = New-Object PSObject
>>
>> Add-Member -InputObject $object1 -MemberType NoteProperty -Name prop1 -Value "value1"
>> Add-Member -InputObject $object1 -MemberType NoteProperty -Name prop2 -Value "value2"
>>
>> $object1
>> $object1.GetType()

prop1  prop2
----- -----
value1 value2


PS C:\Users\praneet.manda> # New style of creating an object
>> $object2 = [PSCustomObject]@{
>>     prop1 = "value1"
>>     prop2 = "value2"
>> }
>>
>> $object2

prop1  prop2
----- -----
value1 value2


PS C:\Users\praneet.manda> ■

```

```

>S C:\Users\praneet.manda> Get-ChildItem -Path $env:TEMP -Include "*.txt" -Depth 2 -Recurse

Directory: C:\Users\praneet.manda\AppData\Local\Temp\acrobat_sbx

Mode                LastWriteTime          Length Name
----                -----        ----  --
-a---    04-02-2024      20:44           30270 acroNGLLog.txt

Directory: C:\Users\praneet.manda\AppData\Local\Temp\outlook logging

Mode                LastWriteTime          Length Name
----                -----        ----  --
----- 08-01-2024      12:41           85460 Prof_001_OUTLOOK_3ae0_WsdetectBefore_2024.01.08_07.11.53.txt
----- 08-01-2024      12:41           85516 Prof_002_OUTLOOK_3ae0_WsdetectAfter_2024.01.08_07.11.53.txt

Directory: C:\Users\praneet.manda\AppData\Local\Temp

Mode                LastWriteTime          Length Name
----                -----        ----  --
-a---    05-02-2024      11:30            607 samplefile1.txt
-a---    05-02-2024      11:30            500 samplefile2.txt
----- 23-01-2024      15:12           832959 Setup Log 2024-01-23 #001.txt
-a---    05-02-2024      11:35             96 test1.txt
-a---    05-02-2024      11:35             34 test2.txt

>S C:\Users\praneet.manda> $params = @{
>>   Path = $env:TEMP
>>   Include = "*.txt"
>>   Depth = 2
>>   Recurse = $true
>> }
>> Get-ChildItem @params

Directory: C:\Users\praneet.manda\AppData\Local\Temp\acrobat_sbx

Mode                LastWriteTime          Length Name
----                -----        ----  --

```

```

>S C:\Users\praneet.manda> $params = @{
>>   Path = $env:TEMP
>>   Include = "*.txt"
>>   Depth = 2
>>   Recurse = $true

```

## Windows PowerShell

```
PS C:\Users\praneet.manda> class Tree {
>>     [int]$Height
>>     [int]$Age
>>     [string]$Color
>> }
>>
>> $tree1 = new-object Tree
>> $tree2 = [Tree]::new()
>>
>> $tree1.Height = 10
>> $tree1.Age = 5
>> $tree1.Color = "Red"
>>
>> $tree2.Height = 20
>> $tree2.Age = 10
>> $tree2.Color = "Green"
>>
>> $tree1
>> $tree2

Height Age Color
----- 
    10   5 Red
    20  10 Green

PS C:\Users\praneet.manda> class Tree {
>>     [int]$Height
>>     [int]$Age
>>     [string]$Color
>>
>>     Tree() {
>>         $this.Height = 1
>>         $this.Age = 0
>>         $this.Color = "Green"
>>     }
>>
>>     Tree([int]$Height, [int]$Age, [string]$Color) {
>>         $this.Height = $Height;
>>         $this.Age = $Age;
>>         $this.Color = $Color;
>>     }
>> }
>>
>> $tree1 = [Tree]::New()
>> $tree2 = New-Object Tree 5, 2, "Red"
>>
>> $tree1
>> $tree2

Height Age Color
----- 
      1   0 Green
      5   2 Red
```

```

Windows PowerShell

> C:\Users\praneet.manda> class Tree {
>     [int]$height
>     [int]$age
>     [string]$color
>
>     # Initialize the tree by setting default values
>     Tree() {
>         $this.Height = 1
>         $this.Age = 0
>         $this.Color = "Green"
>     }
>
>     # Create a constructor with parameters a.k.a. constructor overloading
>     Tree([int]$height, [int]$age, [string]$color) {
>         $this.Height = $height
>         $this.Age = $age
>         $this.Color = $color
>     }
>
>     [void]Grow() {
>         # Get a random height because plants and trees don't grow the same each year
>         $heightIncrease = Get-Random -Min 1 -Max 5;
>         $this.Height += $heightIncrease;
>         $this.Age += 1
>     }
>
>     $tree = [Tree]::New()
>
>     # Let the tree grow for 10 years
>     for ($i = 0; $i -lt 10; $i++) {
>         $tree.Grow()
>         $tree
>     }
>
Height Age Color
-----  ---  -----
5   1  Green
7   2  Green
9   3  Green
13  4  Green
16  5  Green
18  6  Green
21  7  Green
25  8  Green
27  9  Green
31  10 Green

> C:\Users\praneet.manda> class Tree {
>     [int]$height
>     [int]$age
>     [string]$color
>
>     Tree() {
>         $this.Height = 1;
>         $this.Age = 0
>         $this.Color = "Green"
>     }
>
>     Tree([int]$height, [int]$age, [string]$color) {
>         $this.Height = $height;
>         $this.Age = $age;
>         $this.Color = $color;
>     }
>
>     [void]Grow() {
>         # Get a random height because plants and trees don't grow the same each year
>         $heightIncrease = Get-Random -Min 1 -Max 5;
>         $this.Height += $heightIncrease;
>         $this.Age += 1
>     }
>
>     class AppleTree : Tree {
>         [string]$species = 'Apple'
>     }
>
>     $tree = [AppleTree]::new()
>     $tree
>
Species Height Age Color
-----  ---  -----
Apple      1    0  Green

> C:\Users\praneet.manda> -

```

```
Windows PowerShell
PS C:\Users\praneet.manda> $url = "https://gist.githubusercontent.com/sanderstad/1c47c1add7476945857bfff4d8dc2be59/raw/d12f30e4aa9d2ee18e4539b394a12e63dea0c9c/SampleJSON1.json"
>>> $json = (New-Object System.Net.WebClient).DownloadString($url)
>>>
>>> $data = $json | ConvertFrom-Json
>>>
>>> $data.colors

color category type      code
----- --     --      --
black hue    primary   @{rgba=System.Object[]; hex="#000}
white value   primary   @{rgba=System.Object[]; hex="#FFF}
red   hue    primary   @{rgba=System.Object[]; hex="#F00}
blue  hue    primary   @{rgba=System.Object[]; hex="#00F}
yellow hue   primary   @{rgba=System.Object[]; hex="#FF0}
green hue    secondary @{rgba=System.Object[]; hex="#0F0}

PS C:\Users\praneet.manda> # Make sure you run the previous example first before running this one
>>>
>>> $data | ConvertTo-Json | Out-File $env:temp\json.txt -Force
>>> Get-Content $env:temp\json.txt
[{"colors": [
    {
        "color": "black",
        "category": "hue",
        "type": "primary",
        "code": "@{rgba=System.Object[]; hex="#000}"
    },
    {
        "color": "white",
        "category": "value",
        "type": "primary",
        "code": "@{rgba=System.Object[]; hex="#FFF}"
    },
    {
        "color": "red",
        "category": "hue",
        "type": "primary",
        "code": "@{rgba=System.Object[]; hex="#F00}"
    },
    {
        "color": "blue",
        "category": "hue",
        "type": "primary",
        "code": "@{rgba=System.Object[]; hex="#00F}"
    },
    {
        "color": "yellow",
        "category": "hue",
        "type": "primary",
        "code": "@{rgba=System.Object[]; hex="#FF0}"
    },
    {
        "color": "green",
        "category": "hue",
        "type": "secondary",
        "code": "@{rgba=System.Object[]; hex="#0F0}"
    }
]}
```

## Windows PowerShell

```
PS C:\Users\praneet.manda> # Creating a simple XML document
>> $xmlDocument = New-Object System.Xml.XmlDocument
>>
>> # Adding a root element
>> $rootElement = $xmlDocument.CreateElement("Root")
>> $xmlDocument.AppendChild($rootElement)
>>
>> # Adding child elements
>> $childElement1 = $xmlDocument.CreateElement("Child1")
>> $childElement1.InnerText = "Value1"
>> $rootElement.AppendChild($childElement1)
>>
>> $childElement2 = $xmlDocument.CreateElement("Child2")
>> $childElement2.InnerText = "Value2"
>> $rootElement.AppendChild($childElement2)
>>
>> # Save the XML to a file
>> $xmlDocument.Save("C:\Path\To\Your\File.xml")

Name          : Root
LocalName     : Root
NamespaceURI :
Prefix        :
NodeType      : Element
ParentNode    : #document
OwnerDocument : #document
IsEmpty       : True
Attributes   : {}
HasAttributes : False
SchemaInfo   : System.Xml.XmlName
InnerText     :
NextSibling   :
PreviousSibling:
Value         :
ChildNodes    : {}
FirstChild   :
LastChild    :
HasChildNodes : False
IsReadOnly    : False
OuterXml     : <Root />
BaseURI       :
PreviousText  :

#text : Value1
#text : Value2

Exception calling "Save" with "1" argument(s): "Could not find a part of the path 'C:\Path\To\Your\File.xml'."
At line:18 char:1
+ $xmlDocument.Save("C:\Path\To\Your\File.xml")
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
+ FullyQualifiedErrorId : DotNetMethodException
```

```
PS C:\Users\praneet.manda> # Load XML from a file
>> $xmlFilePath = "C:\Path\To\Your\File.xml"
>> $xmlContent = Get-Content -Path $xmlFilePath
>> $xmlDocument = [xml]$xmlContent
>>
>> # Accessing elements
>> $rootValue = $xmlDocument.Root.InnerText
>> $child1Value = $xmlDocument.Root.Child1.InnerText
>> $child2Value = $xmlDocument.Root.Child2.InnerText
>>
>> # Displaying values
>> Write-Host "Root Value: $rootValue"
>> Write-Host "Child1 Value: $child1Value"
>> Write-Host "Child2 Value: $child2Value"
Get-Content : Cannot find path 'C:\Path\To\Your\File.xml' because it does not exist.
At line:3 char:15
+ $xmlContent = Get-Content -Path $xmlFilePath
+ ~~~~~~ + CategoryInfo          : ObjectNotFound: (C:\Path\To\Your\File.xml:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetContentCommand

Root Value:
Child1 Value:
Child2 Value:
PS C:\Users\praneet.manda>
```

```
Root Value:
Child1 Value:
Child2 Value:
PS C:\Users\praneet.manda> # Load XML from a file
>> $xmlFilePath = "C:\Path\To\Your\File.xml"
>> $xmlContent = Get-Content -Path $xmlFilePath
>> $xmlDocument = [xml]$xmlContent
>>
>> # Modify a child element's value
>> $xmlDocument.Root.Child1 = "New Value"
>>
>> # Save the modified XML
>> $xmlDocument.Save($xmlFilePath)
Get-Content : Cannot find path 'C:\Path\To\Your\File.xml' because it does not exist.
At line:3 char:15
+ $xmlContent = Get-Content -Path $xmlFilePath
+ ~~~~~~ + CategoryInfo          : ObjectNotFound: (C:\Path\To\Your\File.xml:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetContentCommand

The property 'Child1' cannot be found on this object. Verify that the property exists and can be set.
At line:7 char:1
+ $xmlDocument.Root.Child1 = "New Value"
+ ~~~~~~ + CategoryInfo          : InvalidOperationException: () [], RuntimeException
+ FullyQualifiedErrorId : PropertyNotFound

You cannot call a method on a null-valued expression.
At line:10 char:1
+ $xmlDocument.Save($xmlFilePath)
+ ~~~~~~ + CategoryInfo          : InvalidOperationException: () [], RuntimeException
+ FullyQualifiedErrorId : InvokeMethodOnNull
```