**Swap OS disk/Snapshot**

* Initially create a virtual machine >> login to it >> install some softwares or make some changes in it
* In the VM blade>>disks>>open disk>>create snapshot and login back to the server and make some more modifications on it after taking the snapshots for the revert back the changes
* Stop the VM>>in seach type snapshot >>create a disk with it t>>save>>
* VM>>disks>>swaposdisk >> OK>>start the VM (we need to observe the 1st modifications present on it)

**Capture custom Image/SYSPrep**

* Login to vm >> cmd>>cd sysprep >>sysprep /generilized (OOBE generilize & shutdown)
* In the console after stopped >>stop (to deallocate it) >>capture >>OK
* After deploying>>create VM >> start (few txt files may miss bcz it might thing of the log file)

**Public Ip disabling**

* Open vm blade>>networking>>network interface >>ip configurations >>edit the below ip >>disable>>save

**Custom Route table routing**

* Need to establish a custom route from public to private using DMZ (custom traffic routing)
* Create a 3 Vm for 1st give public IP and for the second private ip and for DMZ also private ip only
* Create 3 subnets in vnet (public, private, DMZ)
* Login to public ip enabled vm and try to ping the private ip vm and dmz ip vm ip’s
* Create route table >> routes>>add>>public2privateviaDMZ address as destination ip (private ip) note dont give exact ip give the subnet CIDR range virtual appliance ( using as a media -router) need to give the exact dmz private ip only
* Vm with public ip >> subnet>>click existing subnet>> select the created route table
* Virtual appliance (using as a media -router) then we need to enable ip forwarding to thet DMZ VM in NIC card
* Login to the DMZ vm >> regedit>>Localmachine\system\currentcontrolset\services\tcpip\parameters >> change the regedit value to 1 for ipenabledrouter>>restart the vm
* Use the tracert privateip (you can see the communication goes via DMZ)

**VNET Peering**

* Vnet>>peerings>>add>>select the dest VNET and give details of 2-way peering we can enable or disable one-way or two-way peering here>>add
* Login to VM1>> ping the vm2 ip (able to ping bcz of peering) or use RDP (using private ip’s)do it wise versa

**Global VNET Peering**

* Vnet>>peerings>>add>>select the dest VNET (in different region) and give details of 2-way peering we can enable or disable one-way or two-way peering here>>add  
  If destination vm of different region is ot in our subscription then use resource id.
* Login to VM1>> ping the vm2 ip (able to ping bcz of peering) or use RDP (using private ip’s)do it wise versa

**Service Endpoint & Private Endpoint**

* Vnet>>Service endpoint >>add>>give service name(exp Azure storage) network private vnet.
* Create a storage account >> Networking >>selected network>>add the existing private subnet address
* private VM >> NSG>>add outbound rule(service tag storage region) on 443 .
* Storage acc >>private endpoint >>add PE (region)>>resource type(storage), resource (sub resource)>>private vnet >> create.  
  (a new network interface card will be created to access storage account on private ip)
* Storage acc>> container/fileshare >>connect>> powershell script>> login to private vm>>open powershell>>run the given powershell script >> you can able to see network drive will be mapped in file explorer.

**Load Balancer**

* Create 2VM assign Private Vnet and Availability set
* LoadBalancer type(internal use or for public)
* Login to those vm using some public ip vm and install IIS webserver in both the machines
* Load balancer>>Backend pools>>add>>associate to VM’s
* LB>>Health probe >>add (protocol & interval)
* LB>>LBrule>>add (port, pool,probe, persistence,timeout)
* Hit the public ip to access the webserver (website’s of VM1 & VM2)

**Application gateway**

* Create an application gateway>>add (tire,SKU,instances,autoscaling max100 firewall status, firewall mode, HTTP/V2 mandatory we need to create subnet , public ip, add backendpools(VM1 &VM2) add routing rules(listeners & backend targets)>>create.

Traffic Manager

* Traffic Manager need to define routing method (performance,weighted,priority,geographical)

**Point-to-site VPN**

* Vnet>>subnet>>create a gatewaysubnet
* Create Virtual network gateway (gateway type VPN)>>point2site config>>add config (address pool,tunnel type,authentication type,self signed certificates)>>download vpn client>>install it >>check the ipconfig of clientM

**Site-to-Site VPN**

* We need to create another VM either in different cloud or on onprem.
* Create VM in AWS>>login to vm >> add remote access feature>>subnets>>edit route table>>add route>>dest IP is the azure private VM’s IP target VM(NIC) >> login to VM >>remote access>>deploy VPN only>>select serverhost name right click >>configure custom config,VPN access>>Networkinterface in the logged in VM  
  VPN only,name,IKEv2 protocol,Virtualnetwork gateway IP in azure, add the dest IP details of Azure machine
* In azure console>>Local gateway>>add give aws public ip & aws CIDR create>> add connection associate to virtual network gateway and presharedkey & ike2 protocol >>login to VM>>remote access>>network interface>>newly created one right click set the same preshared key& connect.

**Sample Azure Dev0ps project**

* Azure Dev0ps projects>>select .Net (runtime)>> select the asp.net.core (framework)>>windows web app (service)>> create
* If we open Azuredevops we can find the new project with the specified name and it starts building
* If we do the modification in the any source code file once we commit again the build process and release everything will start and will be created as the version 2

**Deploying VM and few Resources using ARM Template**

* In Azure quick start templates we can get the ARM templates in the github
* We need to download the **main json** file and the **paraments json** file we can give and modify the parameters json file and save it >> open powershell >> connect-Azaccount (which connects the azure account) >>   
  **New-AzResourceGroup -Name ARM-RG01 -Location EastUS   
  New-AzResourceGroupDeployment -ResourceGroupName ARM-RG01 -TemplateFile D:\ArmTemplates\_25122020azuredeploy.json -TemplateParameter File D:\FirstPorject01\azuredeploy.parameters.json** >>upon executing the above commands now we can see the resources group and all other resources can be created under RG in the azure GUI console.

**Providing Access policy to key vault**

* Create a VM and key vault Resource group go to the vm on the azure console >> identity blade>>status ON >>add azure role assignments >> give the reader access to the key vault
* Key vault >> add access policy >> secret manager >> select priciple as VM name and define access policy->>save  
    
  How to view the secret content using powershell by placing URI ?  
  **$response Invoke-RestMethod -Uri 'http://169.254.169.254/metadata/identity/oauth2/token?api-version-2018-02-01&resource=https%3A%2F%2Fvault. azure.net' -Method GET -Headers @[Metadata="true"}  
  $response. Access\_token  
  $result = Invoke-RestMethod -Uri https://nextopskv1.vault.azure.net/secrets/WinAdmin Pass/0ce16b4b4df3482e8bbcb1b0b5b45996?api-version-2016-10-01 -Headers @{"Authorization" = "Bearer $($response. access\_token)"}  
  $result | select id, value, contentType**