

AZ-303 NOTES

STORAGE ACCOUNT

STORAGE ACCOUNT TYPE

- BLOB
- TABLE
- QUEUE
- FILE

CREATING A STORAGE ACCOUNT:

Basic Tab: -

- Name
- Region
- Performance- standard or premium
- Premium acc. Type
- Account kind-gen2 or gen1
- Redundancy-LRS, GRS, RA-GRS,ZRS

Advanced Tab:

- leave as it is

Networking tab

- leave

Data Protection tab

- Soft delete-remain for period of days we choose after deletion

Tags Tab:

- leave

Review And Create Tab:

- Leave
- Open storage acc. Deployment

WORKING WITH BLOB SERVICE:

- From sidebar open Blob tab
- Container option
- CREATE NEW CONTAINER:
 - Name
 - Public Access level--Private (no anonymous access)
 - Blob (anonymous read access for blobs only)

- Container (anonymous read access for containers and blobs)
 - Open the container:
 - We can upload our object in this tab
 - Browse we can upload image or do etc.
 - Open the file we uploaded
 - Overview of file
 - Edit the file
 - We can make changes to the file and download the file.
 - We also get a URL for the object in the overview section, we can copy it on browser and go to the data. we will not be able to access as we are anonymous user. change the public access level of the container to access the contents of the container. Now we will be able to access and see the contents using URL.
 - If image file is greater than 2MB it will not be able to see the image on that board.
 - We can have different formats or extensions like JavaScript file.
 - We can also create folders inside folders and so on.
 - We can change our access level tab

Refresh Tab

Delete Tab

Change Tier

USE CASE SCENARIO FOR BLOB

Suppose we have an application running in our env. And our users use this app to upload images, files, and videos, instead of storing these objects onto the app which they are using we can store it into storage acc. Using container and app can store videos using container. Instead of locally storing the videos we can use the storage acc. As a separate service.

FILE SERVICE

- We can have file share that is use by users, or VM's and VM's can map a drive and access the file share.
- In Blob the object gets a URL but in file share not.
- For sharing resources across users and VM's.

CREATING FILE SHARES

- Open SA
- from sidebar find file share
- Create file share
- Name
- Quota-mentions size in GB
- Tier-Premium, Transaction optimized, Hot, Cool

GO TO FILE SHARE:

- We can add directories: - select any file by browsing
- We can upload a file
- We can delete directory

Refresh Tab

- When we open a file that we uploaded:
- In overview tab there is an URL
- But we will not be able to open
- So, in file share we have an option of connect
- We will have a script copy that in our PowerShell and press enter
- Now it will map a drive to our local system, we can see in the File explorer we will have a shared drive and
- We will have all the folders and file in that drive.
- We can make a new doc and then we can refresh the file share, we will see that file also that we made above.

TABLE SERVICE

- Suppose we have an app storing tables using SQL server, but in SA it is NO SQL Store.
- In normal we can query data using SQL
- In NOSQL datastore everything is stored as entity, there is no predefined schema, it follows schemeless design.
- In this we can query on different entity but in SQL we need to use Join and all to query data. So, each have its pros and cons.
- Everything is stored in SQL as key value pairs
- In NO SQL we have a partition key and a row key.

Partition Key- used to define the partition to store the entities

Row key-Helps to uniquely identify an entity in the partition.

- To find an item in item we can use the partition and it becomes easy.
- Like we have partition based on partition so, we can search a course acc. To in which course ID it will be available.

CREATING TABLE SERVICE

- From sidebar open table service
- Creating a table:
 1. Create table and name it
 2. Storage explorer ke through jao in created table and add krdo details.
 3. We need to have our partition key and row key
 4. We will see tabs to fill values

5. Then we can run query on the table and see the result by running query.
6. We can use storage explorer to work with various storage services like blob, file....

INTO TO STORAGE EXPLORER:

- Install krlo
- Open kro
- Use subscription to sign in and use azure acc. to login.
- We will see all subscriptions and we can choose one on which we want to work.
- We will see all SA's and all services that we have in our subscription.

DIFFERENT AUTHORIZATION TECHNIQUES:

- Access Keys
- Shared Access Signatures
- Azure AD

HOW TO ACCESS USING ACCESS KEYS:

- Go to azure SA and there we have an access key option, and we can see 2 keys.
- Copy any key
- Go to storage explorer
- Click on add account
- Choose SA option
- Place the acc. key
- Copy acc. name
- And then connect
- That we will have access to all services as access key give permissions to all services.

HOW TO ACCESS USING SAS:

- We can generate SAS at SA level or at Blob level
- At Blob Level go to containers in the data container
- Change access level to private
- Go on to the object there we have an option for generate SAS
- We need 1 key and permissions(read/write)
- Start and expiry date and time
- Allowed IP address
- Click generate SAS and we will get the URL and copy and paste to browser we will see object.
- We can give access to particular user and to particular object, more secure
- At SA Level, we will have an option to click on the service to which we want to give access like blob, file...

- Choose Resource type (Service, Container, Object)
- Allowed permissions (Read, Write, List, Delete, Add, Create, Update, Process)
- Start and expiry
- IP address
- Access key
- Generate SAS token
- We will get a URL but will not open as it is for whole container, so use SA
- Go to SE than add acc. using SAS
- We will be login and we will see the Blob only as this service was only selected and we will not be able to del. as no write permissions.

So, this is more granular level.

STORED ACCESS POLICY:

- Go to the container and there from sidebar we have an option of access policy
- This will have more control using SAS we can use it.
- Add policy
- Policy name
- Permissions (above of SA same)
- Start time and expiry time
- Click ok
- Reason for having the policy, suppose the SAS get into wrong hands but it will expire only at a time we mentioned, so we can use the stored access policy and make the SAS invalid after some time.
- Create a SAS again using the storage explorer tab in SA in azure, in that blob--> data-->get SAS-->choose policy name and click ok
- Now we will get URI and copy that
- And again, to SE than using SAS login kro and we will see the blob and all.
- Goes onto wrong hands
- Go back to policy and edit the policy and remove all permissions and again open SE and then we will see will not be able to authenticate as if we will try to login using SAS again using the URI or we could refresh also SE.

How to access using Azure AD:-

- Go to AD, create a new user, fill the details.
- After that copy the id of that user after creating.
- Go to SA, go to access control-in this we give role assignments
- Create add role assignment
- Choose the role(reader.....)
- Assign access like to user, group, or Service principal.
- Select the user and click save.

- Now we have given reader role to a user for SA.
- Again, add role assignment--. we can see various roles, storage blob reader
- Choose user
- Go to SE
- Add an acc.--> sign in as a new user we created.
- We need to create pswd for him
- Apply
- We will see that we have access to our only SA to which have given access
- And to blob only read access
- We will not be able to delete anything.

Most Secure Way Is Using

AD-->SAS-->ACCESS KEY

strong----->weak

- When audit access needed, we should use SAS as that want for limited time.

STORAGE REPLICATION:

- Azure made multiple copies of data that is stored.
- this helps to protect against planned and unplanned events-transient hardware failures, network, or power outages.
- LRS- least cost option, we have 3 copies of your data in a particular location.
- ZRS- helps to protect against data center level failures.
- GZRS- data replicated to another region.
- RA-GZRS access data in both regions, but 1 is read only.
- to access the secondary region, we need to append the URL for the blob container with secondary.

STORAGE ACCOUNT-ACCESS TIERS

- Pricing for blob service-see table online
- Hot-Accessed frequently. Can be Enable at SA Level.
- Cool-accessed infrequently and stored for at least 30 days. Can be Enable at SA level.
- Archive-this is used for data that is rarely accessed and stored for at least 180 days. Only be enable at Blob level.
- In the SA account we have, got to configuration to change the access tier.
- now whenever we will upload an object to this SA it will have the tier we have choose or set at SA level.
- We can change the access tier of individual object at the blob level.

- Archive is the least cost option among all, but we must rehydrate the file or object to access the file, to access that, and it takes time for rehydration. After archiving we will not be able to download anything or no option for it.
- Standard priority- here request is processed in order received and can take up to 15 hrs.
- High priority- here the request is prioritized and could finish in under 1 hr for object under 10 GB in size.

COSTING ASPECTS OF SA'S

- Cost is also based on the operations like for write and read etc., for hot it is less and cool, archive is more.
- Storage cost is less in archive and access cost is less in hot.

LIFECYCLE POLICIES

- Using lifecycle mgmt. rules, we can set a condition and change the access tier acc. to that.
- Go to SA-->lifecycle mgmt.

Creating Rule:

- Rule name
- Rule Scope
- Blob Type
- Blob subtype (Base Blobs, Snapshots, or versions)
- If condition then delete, change tier etc.
- We can have multiple rules and we can disable and enable rules.

CHANGE REPLICATION:

- From LRS to ZRS, we need to perform either manual migration or live migration.
- Manual migration, we need to create another SA with ZRS type, and then copy from source onto destination.
- We can also request Microsoft for live migration. this ensure that you have no app downtime during migration.
- From LRS to GZRS or RA_GZRS, first we need to switch to GRS or RA_GRS and then request live migration.
- From GRS or RA-GRS to ZRS, first switch to LRS and then request live migration.

Changing:

- Go to SA--> configuration--> we can change replication (LRS<GRS<RA-GRS) for ZRS too at migration process.
- If we have object in archive tier, we will not be able to change replication, we need to change access tier firstly.

QUICK NOTE:

- Page Blobs are used for VHD or virtual hard disk files.
- Block Blob- we get premium performance.

AZURE CLOUD SHELL

- For writing scripts
- We can PowerShell cmds and CLI cmds
- We need to have a SA.
- Run cmds in cloud shell to create SA.
- Will give result in json
- We can use this tool to work with resources.

THE VIRTUAL MACHINE SERVICE

CONCEPT

- We know why we will create VM in azure, we don't need to manage underlying Infra like physical server.

DEPLOYING

- Number of resources that get deploy when we deploy VM
- VM
- Virtual Network.
- Disks
- N/w Interface
- Public or Private Address
- NSG

CREATING VM IN AZURE

- Create a resource or on tab of VM

TABS INCLUDED:

Basic Tab

- Subscription
- resource grp
- VM name
- Region
- Availability options
- Availability zone
- Image (OS type)

- azure spot instance
- Size (CPU's and memory and estimated cost0
- username and pswd
- inbound Port Rules
- public
- select port (RDP,HTTP)

Disk Tab

- Disk Type (SSD)
- Encryption Type
- Data disks if we want additional disk.

Networking Tab

- VNet (create new or use existing)
- Subnet(auto)
- Public IP (auto)
- NIC (none, Basic, advanced)
- Public inbound ports (none, allow selected ports)
- Load balancing

Management Tab

- Monitoring
- Boot Diagnostics
- Identity
- Auto Shutdown
- Backup
- Guest OS Update

Advanced Tab

Tags Tab

Review And Create Tab (Total Charge And Review All Settings)

Connecting To VM

- Go to VM resource we created
- we can see all details in overview

Connect Button Overview Tab We Have 3 Options:

- RDP-will download RDP and we will enter username and pswd and we will logon on our VM.
- SSH

- Bastion

In VM, Open server mgr. to install the Web Server (IIS) role.

now we can open Internet explorer and type http://localhost we could see IIS services.

ANOTHER WAY TO CONNECT

- Using Public IP but it has some rules. so go to networking section in VM
- Add inbound port rule:
- Source-Any
- Source port ranges
- Destination-Any
- Destination Port Ranges (80)
- Protocol-ANY/TCP/UDP/ICMP
- Action-Allow or Deny
- Priority
- Name
- Now we can copy IP than we can type IP to browser we will see the IIS service page.

STATE OF VM

- **Managed Disk:** - managed by Azure, designed for high Availability. size-127GB
- **Temporary Disk:** size varies depending on instance size we choose. If there is an maintenance event the data will be lost on the temporary disk, so try not to store imp data on this disk.

RESTART AND STOPPING VM:

- If you restart VM from VM or from azure portal, the Public IP address. will remain as it is. also, the data on temporary disk remains as it is.
- If we shut down VM from VM itself, in azure it will show status as topped but public IP remain as it is, and we will be incur charge for the compute. If we don't want to incur from azure portal stop krdo and the file will also remain as it is.
- If you Stop/Deallocate the VM from azure portal, the Public IP add. will be lost. The data on temporary disk also gets erased and we will not incur any cost.
- If we will start VM again it, will get new Public IP and need to download new RDP file and file will also not be there

DEPLOYING LINUX VM'S

- Same details as above
- Select image ubuntu
- It will allow port SSH other than RDP
- Same for other tabs

- Review and create krdo.

To connect to Linux VM we need to download putty tool.

- Open tool-->IP address copy krdo
- CMD will open, username and pswd dalo on cmd.
- install nginx server
- create an inbound rule same as above to allow traffic at port 80 from azure portal.
- now if we copy and paste IP on our browser, we will nginx server page.

TERMINATING RESOURCES

- Delete resources if we are not using to save on costs.
- id we will delete the VM, it will not delete the resources related to it, we need to delete them by selecting them.

DISKS FOR AZURE VM'S:

Un Managed Vs Managed Disk:

These are talking about data disks

- to use unmanaged disk, then you need to manage the storage acc. that would be used to store the disks.
- the SA can be gen v1 or gen v2 but must be premium type.
- Microsoft recommends use of managed disks.
- With managed disks the durability and availability of disks are managed by Azure. the mgnd disks are designed for 99.999% availability.
- Refer documentation of Microsoft for seeing the different type of disks (Ultra, Premium SSD, Standard SSD, Standard HDD).

ADDING DATA DISKS

- Go to the VM than go to disk section.
- Click Create and attach new disk
- Fill name, size, storage type and max IOPS and Max TPS.
- Go to the VM and initialize the disk
- and we will have a new volume in the file explorer we could see.

SECONDARY NETWORK INTERFACE

- We get 1 NIC attached to our VM

NEED OF SECONDARY NIC:

- Suppose we have a VNet (10.0.0.0/24)
- And in that we have 2 subnets
- SubnetA (10.0.0.0/24)

- SubnetB (10.0.1.0/24)
- In SubnetA we have 1 VM with both Private/Public IP
- In SubnetB we have 2 VM's with Private IP
- And these 2 VMs are hosting apps, now if suppose these VM's want to download updates from internet. So, they can go through the appliance can be VM in subnetA and can get updates from the internet and then the VM in subnetA is helping in providing updates to the VM in VNet in different subnet. and the VM can have 2 NIC, 1 interface have both Public and Private IP, Public to get update from internet, Private of the VM itself.
- And secondary interface having a Private IP, o relay info to other VM's in another subnet. It can be bcoz of security reasons as well.

ADDING SECONDARY N/W INTERFACE:

- Creating a new VM and new VNet with 2 subnets subnet a and subnet B as above example.
- Stop VM first
- Go to networking section
- Add new network interface
- Create
- Name
- Choose the subnet
- Click create
- To route traffic from 1 N/w interface to other n/w interface we need to have routing mechanism.

RESIZING THE VM:

- Go to the VM --> go to size section--> choose size acc., and click ok, machine will be resized.
- We can see usage and quotas as each region has a limit of size we can create a VM's

CREATING A VM CUSTOM IMAGE

- Like if we have an VM, on which we have install the apps and al on it and now we want that VM to deploy using azure, we can use do this.
- After creating an image of the VM deployed in Azure the VM can't be used further than, it will be waste after that.

CONNECT TO VM

- File Explorer-->Windows C:--> System32-->Sysprep-->click on the application
- select generalize, shutdown
- it is destructive process as after that we will not be able to use VM.
- in Azure Cloud shell stop VM write command

- press capture button in VM tab we will see.
- name image
- RG
- VM name
- now search for images the name we have given in azure.
- go onto image and create VM click button and give name and give i=other details
- we will also have IIS services installed on this VM as we capture image
- we can copy and paste the IP and we will see the IIS service page.
- So, we can create our own custom image and create VM out of it.

VM SLA

- For any single instance of VM using premium storage for all Operating system disks and data disks we guarantee you will have VM connectivity of at least 99.9%
- for all 2 or more instances deployed in same availability set we guarantee you will have VM connectivity of at least one instance at least 99.95% of the time.
- for all 2 or more instances deployed in same availability zones we guarantee you will have VM connectivity of at least 99.99% of the time.

AVAILABILITY SET

- if we have set of VM's and these VM's are hosting the apps, and as we know when we create the VM in azure it will create n Physical server in a rack, if both on same physical server, but if server goes down and then our both VM's go down and our whole App goes down.
- That is why we can make use of Availability zone
 - **Fault Domain-** in this 1 or more VM is connected to different power and network. source. So, that of Power goes down not all VM's goes down. we can have 3 or more fault domains depending on requirement. It works vertically.
 - **Update Domain-** if Microsoft update the physical server and it require restart and then VM will not be available at that time. it will work horizontally and then 3 VM's will not work in a line but VM in another line will work. So, instead of updating all at a time, it will update each at a time, as physical server is place horizontally.

Further aspects in Availability Sets:

- Underlying VM's can be use either managed or unmanaged disks
- If the VM's are part of AS, then Microsoft recommends using Managed disks for underlying VM.
- Managed disks provide better reliability for availability sets by ensuring that the disks of VM's in an Availability set are sufficiently isolated from each other to avoid single point of failure.

- Most of regions have a maximum of 2 or 3 fault domains.
- Configure each app tier as part of different availability set. Like web tier different and App tier different set.

How to work with Availability Set

- Create an availability set separately simply searching in azure.

Basics Tab

- Subscription
- RG
- name
- region
- fault domains
- update domains
- use managed disks (no or yes)

Advanced tab

- Choose proximity grp- the VM will place close to each other.

Tags Tab

Review and Create

NOW CREATE A NEW VM

- Choose the availability set when filling details, should be in same region same region m kro create VM.
- Other details as above fill in
- In availability set we will see the fault and update domain as zero
- Create another VM as the existing VM cannot be assigned to an availability set.
- In availability set we will see the fault and update domain for another VM will be 1 for both.

USE CASE SCENARIO

- You must move an o-premises app onto azure subscription. the app will be hosted on several Azure VM's
- you must ensure that the app will always be running on at least 4 VM's during a planned azure maintenance period.

Number of Update Domains we need?

- We need to have 5 Update Domain for 4 VM's to be working.
- Azure maintenance Period-Update Domains
- Faults to the underlying hardware-Fault domains.

VM SCALE SETS

USE CASE SCENARIO

- We have an app, and it is use to process jobs and the CPU percentage goes to above 90% and the jobs increasing
- Then we need to have a multiple VM's to control CPU usage, but this is manual process.
- Now if we want to do it auto. we could use VM Scale Sets acc. to scaling condition set.
- The VM Scale set will auto the no. of VM's based on demand.
- You define the configuration of the VM's would be part o the scale set.
- You then define the scaling conditions.

VM SCALE SETS Points

- This service allows you to create and manage a group of identical virtual machines
- you can also place scale set behind a load balancer to distribute the traffic across VM"s.
- the number of VM's instances Automatically increases or decreases based on demand on VM scale set.
- the use of VM's helps provide better redundancy and improved performance for your apps.

CREATING VM SS

Basic Tab

- Search for VM ss
- Subscription
- RG
- Name of ss
- Region
- Availability zone-no
- Image
- Size
- Username and pswd

Disks Tab

- disks leave as it is

Networking Tab

- Create a new Vnet
- Edit network interface

- Allow ports SSH and port
- Public Ip allow

Scaling Tab

- Instance count -1
- Scaling policy-custom

Scale out

- CPU threshold-percentage mention
- Duration
- Number of VM's increase by

Scale In

- CPU threshold-percentage mention
- Number of VM's decrease by

Management tab

- Turn on boot diagnostics

Health Tab

Advanced Tab

Tags Tab

Review And Create Tab

- When we will go to our scale set, we created we will see our VM up and running in that.
- Go onto the VM using Putty and Linux VM and then install tool stress on the VM, to increase a stress on the VM and we will see in the azure portal in Scale set there will more than 1 VM.
- If we go to overview tab in Scale set there, we can see the graph of CPU percentage.

USE CASE SCENARIO

- One aspect if we have a VM in VM Scale Set and it CPU goes above and we have an app running on that VM will the app be there on the other VM that will be created using Scale set, the answer is no.
- we need to install the app itself or we can use Custom Script extensions or create a custom image.
- Another aspect i how can we ensure that the number of users in each VM is also distributed equally. we need to use the Load balancer.

AVAILABILTY ZONES

- These are unique physical locations that are equipped with independent power, cooling, and networking.
- There are normally 3 availability zones in a region.
- It is collection datacenters
- Each zone is collection of 1 or more DC's
- DC's is collection of multiple racks and servers; it is a building.
- In each zone we can have 1 or more VM's
- These are used in the region failure conditions
- It gives better SLA
- No additional cost for creating availability set or zones.
- Communication across the VM in availability zones, it requires cost but not for Avail set.

CREATING AVAILABILITY ZONES

- Create a VM
- And choose number of Avail zones acc.
- We can create VM SS and add the avail zone as 3.
- and fill other details.
- If we check different scale set overview we can see the zone mentioned acc.

REVIEW OF AVAILABILITY SERVICES

AVAILABILITY SETS

- When you host your virtual machines in Azure, you sometimes need to cater to the following
- An unplanned event wherein the underlying infrastructure fails unexpectedly. The failures could be attributed to network failures, local disk failures or even rack failures.
- Planned maintenance events, wherein Microsoft needs to make planned updates to the underlying physical environment. In such cases, a reboot might be required on your virtual machine.
- You can increase the availability of your application by making use of availability sets. Each virtual machine that is assigned to the availability set is assigned a separate fault and update domain.
- Fault domains are used to define the group of virtual machines that share a common source and network switch. You can have up to 3 fault domains.
- Update domains are used to group virtual machines and physical hardware that can be rebooted at the same time. You can have up to 20 update domains.

AVAILABILITY ZONES

- This features help provides better availability for your application by protecting them from datacenter failures.
- Each Availability zone is a unique physical location in an Azure region.
- Each zone comprises of one or more data centers that has independent power, cooling, and networking
- Hence the physical separation of the Availability Zones helps protect applications against data center failures
- Using Availability Zones, you can be guaranteed an availability of 99.99% for your virtual machines. You need to ensure that you have 2 or more virtual machines running across multiple availability zones.

AZURE DEDICATED HOSTS

- Hardware Isolation- No other company VM's will be placed on the host
- You can control the maintenance events.

CREATING DEDICATED HOSTS

- Search for dedicated hosts in azure portal.

Basic Tab

- Subscription
- RG
- Name
- Location
- Host group-> name, availability zone and fault domain
- Size

Tags tab

Review and create tab

- We haven't created as it is meant for large org.

AZURE VIRTUAL NETWORK

- We need to have a subnet
- We need to define address space (10.1.0.0/16) and then we can define address for subnet (10.1.1.10,10.1.2.9) as it will subset of address space.
- Within VNet communication between VM's is possible.
- Like subnet is for Web app and other for DB and then we can communicate easily.
- NIC is attached to VM
- Private IP is the address which is taken from the address space and used for internal communication between VM's or outside if we have configured that.
- If 1 VM host Web server and other VM host DB server it will communicate using private IP and establish connection on private Ip.

Now if we want resources on internet to reach on the VM (Web server) that we need to have Public IP assigned to the NIC of the VM. This IP has nothing to do with the address space of the VNet.

Like if I want to connect to the VM (web server) from my personal lapy than, I will go to internet and using the RDP services will connect to that VM.

DB VM has no Public IP and we don't want to expose it to internet, to secure our data.

Notes on Azure networking:

- Azure VNet allow resources such as azure VM's to securely communicate with each other.
- You define subnets in an azure VNet. this helps segment the network into one or more sub-networks.
- A VNet is coped to a single region.
- Public IP- used for communication with internet.
- Private IP- This is used for communication within an azure Vnet or with on-premise network.
- There are 2 SKU"s when it comes tp public IP

Basic SKU- you can assign either static or dynamic IP address.

- NSG's can optionally be used to restricting traffic via the Public IP.
- there is no support for availability zones.

Standard SKU- you can assign static IP address.

- NSG's can need to be used to restrict traffic.
- they are zone redundant by default.

IP ADDRESSING

- VNet--10.0.0.0/24(address range)
- VM (host IP)-10.0.0.5 come from address range of VNet.
- /24 means there are total 32 bits so, 24 bits are reserved so, last 8 bits can bhi change and host can provide the address.
- /16 it means first 16 bits network ID and last 16 bits be host ID or names
- subnet is a subset of the adder range
- ex: Vnet-10.0.0.0/16
- SubnetA-10.0.0.0/24
- VM in subnet can have in range 10.0.0.0 to 10.0.0.255
- subnetB-10.0.1.0/24
- VM in subnet can have in range 10.0.1.0 to 10.0.1.255

WORKING WITH VNETS:

- We can add VNet from VNet section

Basics Tab

- Subscription
- RG
- Name
- Location
- IP Addresses tab
- Address space
- Add subnet (we will have default)

Security Tab

Tags tab

Review and Create tab

- Now we can add subnets to our VNet.
- We can change address space of the subnets
- Till we don't have any VM
- We can also change the address space for the VNet till it is superset of the subnet.
- To disable the Public IP, we can go to Network interface and go to IP address and disable it.

Network Security Groups

- It helps to control the traffic
- It attached to Network interface or the entire subnet.
- All the VM's will be effect in the subnet.

It consists of inbound and outbound security rules.

- **Inbound rules**-to control traffic that flow into the VM. (VM's in the same VNet)
- **Outbound traffic**-to control traffic that flow out of the VM. (

Info required to add security rule

- Priority
- Port
- Protocol
- Source
- Destination

LAB-NSG

- create a new VM that will be part of the VNet, how to limit traffic if we try to access through our workstation the VM.

- Create Karlo VM
- in networking tab, we allow only RDP in inbound rules.
- public IP needed as we want to access services on the internet.
- connect onto machine using RDP file
- install web server onto the VM.
- in portal in networking section on, open the NSG.
- NSG is a separate resource where we can add inbound and outbound rules.
- 3 rules are default in NSG, VM's in same VNET will allow ow communication, load balancing rule., and denies all other rules.
- we need to add HTTP rule so that we can use public Ip to see the IIS service page for the VM.
- in source we could choose service tag and it will connect internet auto,
- port range-*
- destination any
- action allow
- priority
- name
- now we will be able to connect to VM using public IP.

NSG-Priority setting:

- Will check first rule and if it meets the req. than it will not check further.

NSG-Subnet Considerations:

- If we have a NSG at subnet level and other NSG at NIC level on a particular VM
- If there is no rule at Subnet level for port 80 it will not check the NSG at NIC level.
- If NSG at subnet allow and at NIC deny in the end deny
- If at both allow than only will work or both deny.

APPLICATION SECURITY GROUPS

- It is extension of NSG when coms to traffic filtering.
- We will create a ASG as it is easier to define source and destination as if we want to connect web servers and DB Server. we will define NSG at web server and add members to it and then DB server's NSG will reference to ASG for connecting web server and it will be easy to define source and destination.

LAB-ASG

- Create 2 VM's one with web server (IIS) role installed and from internet install SQL mgmt. server studio.
- on other VM(DB), from internet install Microsoft SQL server 2019 and fill details and run the and install package.

- To connect to the SQL server, windows search for windows defender firewall rule and add inbound rule of TCP, port-1433(as Microsoft SQL listens on this port). name and all done.
- Now go to SQL server configuration, properties and IP address and enable it, and restart the SQL server.
- in the portal in DB VM add inbound port rule to deny all VNet traffic as it is default allow. We need to be more secure that is why.
- now move to Web server VM and open SQL Mgmt. Studio that we install, put, username, IP of DB and connect, we will not be able to connect.
- Now in DB VM in portal add inbound port rule for allowing traffic to web server VM, and now connect using Web server VM using SQL mgmt. studio, we will be able to connect.
- So, very hectic, now we will create a resource ASG, fill details, location same as Web VM.
- now in Web VM we will have a ASG option and then configure it and choose the ASG we created.
- Now in DB VM in portal networking section select, add inbound rule, instead if mentioning source address, we can choose that ASG.
- everything will work fine, try to connect now from SQL mgmt. studio.

USING JUMP SERVER AND BASTION HOST

Suppose we have a 2 subnets in a VNet, 1 host web server and other DB server and we want to connect DB server from our workstation, so we will need a Public IP, but Providing Public IP to DB server is not secure, so we could have a another subnet where we will have a VM with both Private and Public IP and public Ip can used to connect to our workstation using port rule in 3389(RDP) and then form that VM we could take RDP of the both VM in a VNet via a private IP address and now it is secure connection. this is idea of having bastion host and jump server.

LAB

- We will have a VNet having 2 subnets
- JumpSubnet (Public IP)
- SubnetA (No public IP)
- We should have a NSG for connecting to our workstation, on the jumpserver VM.
- Create Vnet-10.0.0.0/16
- And add subnet, subnetA-10.0.0.0/24
- Create A VM, same location and choose no for Public IP, and choose our VNet and SubnetA.
- Now add an additional subnet on the VNet we created for our jump server (10.1.0.0/24)

- Create a VM for the Jumpserver, it should have a Public IP, and in the subnet that we created. and add RDP rule and IP of your workstation.
- Now in the other VM, go to networking section, add RDP rule and mention Private IP of jump server.
- now connect JumpVM and open RDP services on the jump server and not copy private IP of other VM and connect and we will be able to connect.

AZURE BASTION HOST

- For connecting to VM from our workstation we need to have Public IP but company may feel risk of having Public IP for the VM, so we could use bastion host.
- In this we will have 2 VM's in different subnet.
- 1 VM will have Private IP
- Other VM will have Public IP and we connect from our workstation using the public IP on this VM and further will take RDP/SSH to the 1st VM via a Private IP.
- We can have separate NSG's on both VM to restricting traffic.
- Azure Bastion is deployed per VNet.
- Azure bastion is a managed service we need not to take IP and we can simply open in browser.
- Azure bastion host must have a AzureBastionSubnet in place.

LAB-AZURE BASTION

- Create a new VM, add none inbound port rule, in networking we will create a new VNet (10.1.0.0/16) and add subnet A(10.1.0.0/24) and AzureBastionSubnet(10.1.1.0/24).
- And choose subnet A for this VM and should not have any public IP.
- Connect using bastion, use bastion.
- It will give steps, name of host, and create a public IP and create bastion.
- Now type username and pswd, and connect, it will connect from the browser on VM without take RDP and all.

SERVICE ENDPOINTS

Suppose we want to access Azure SQL DB and Azure Storage Accounts services using private IP add. These both services are public endpoints, but we want to connect using private connectivity within the VNet from VM than we need to make use of service endpoints.

Using service endpoint all the traffic goes through the secure connection over the Azure Backbone network.

LAB

- You need to create a SA and inside container upload a any file.
- Create a VM in same location and install a SE on the VM.

- In SA, go to networking section, we can add firewall, don't add abhi.
- In VM in SE, connect on to the SA using access keys.
- It will not connect as no firewall is added.
- On portal in VM, go to VNet, service endpoints
- Add service endpoint-->service-Microsoft.storage-->choose you default subnet and add.
- Go back to SA-->Networking-->allow access to-->selected networks--add the VNet and a subnet-->add.
- Now on the VM in SE--> refresh--> and go to container-->go to fie, we will see that file.

PRIVATE ENDPOINTS

- It is a network interface that can be provisioned in your VNet.
- It allows you to connect privately and securely to a service.
- The service can be a storage account, azure cosmos DB.

LAB

- We have a SA, add the selected networks but we also an tab for private endpoints,
- Add private endpoint, same region, name of endpoint, choose target resource (Blob),choose VNet and a subnet and create it.
- Go to VM-->open SE-->refresh all and we will only be able to connect to blob does not file share, so securely connection to a resource.

CUSTOM ROUTING

- System route is defining in the VNet which route the traffic on different subnets.

In this if suppose we have 3 subnets and each has a VM, all in same VNet and 1 VM wants to connect or communicate to other VM but we want that request must pass on from 3rd VM to check if is reliable and then pass on the request to the 2nd VM, so, we need to route traffic from 3rd VM before going to 2nd VM, so we need to use custom route.

WE NEED TO CUSTOM ROUTE TABLE:

- Address prefix-10.1.0.0/1
- Next hop type-virtual appliance (drop down)
- Next hop address-10.1.1.0(Private IP of 3rd VM)
- And we will add this route to the 2 VM's

USER DEFINED ROUTES

- Create 3 VM in 3 different subnets
- SubnetA (10.1.0.0/24) demovm2

- subnetB (10.1.2.0/24) demovm1
- subnetC (10.1.1.0/24) demovm3
- Now we want to access IIS service on demovm1 from demovm2(can be done using private Ip on port 80). But we want to route the traffic from demovm2 to demovm3 than it reach to demovm1.
- Demovm3 might have a virtual appliance, so we will need to install a routing service role on this VM.
- demovm1 and 2 pe IIS install krdo.

CREATE A RESOURCE IN PORTAL ROUTE TABLE

- Name
- Subscription
- RG
- Location
- Gateway route enabled
- Create krdo
- Go to route table add a route
- Name
- Address prefix of entire VNet
- Next hop type (VNet gateway/VNet/Internet/V Appliance/None)
- Choose V appliance
- Next hope address of demovm3 kyunki ussey krwna route (koi bhi traffic from vent ab vm3 se jayega)
- Now we also must associate a route with the subnets. Route table m subnets section, associate
- And add krdo subnetA (demovm2 ka)
- Now demovm3 -->networking->Ip configuration-->Ip forwarding enable-->save
- demovm3 pe connect kro and install routing role (remote access) --> routing through server mgr.
- To configure-->deploy VPN-->LAN routing-->start service-->now it understands how to route request.
- Ab demovm2 pe on internet explorer type Private IP of demovm1 and we will be able to connect, and traffic is passing through the demovm3.

VIRTUAL NETWORK PEERING

- There are 2 Vnets and in 1 Vnet there are 2 VM's and other have 1 VM.
- Now we want 1 Vm to connect to other VM in other vnet, by default it is not possible, so we need to do this, we will use via virtual network peering and then both Vnets can communicate.
- The traffic between the Vnet will go through the Azure backbone network not through the internet, so it is good thing.

POINTS

- The traffic from Vnets transverse through the backbone network of azure.
- you can connect Vnet n same region or across regions.
- you can also connect VNets across different subscriptions
- you also must ensure that IP address don't conflict for the Vnets
- both of you Vnets are using classic resource deployment model then you can't use network peering. then you should make use of Vnet private connections.

LAB

- We have 2 VNets(stagingnet and testnet)
- Stagingnet (10.0.0.0/16) with 1 VM having IP(10.0.0.4) in a subnetA
- Testnet (10.1.0.0/16) with 1 VM having IP(10.1.0.5) in an subnetB
- Create the above scenario using azure portal.
- Go to either of the VM and download RDP file and connect to the VM and add web server role onto it.
- Connect to testvm also and connect onto it, in server mgr. in local server turn on IE enhanced service and the open internet explorer and type private IP of staging VM we will not be able to connect
- We need to have a rule that allow traffic on port 80 to connect using our browser using public IP.
- We need to create connections from both side.
- Got to staging network-->peering's-->add-->link name-->allow traffic-->choose the network-->testnet--> this will create connections for both side.
- Done
- We can see in testnet-->peering's-->we will see the peering
- Now open test VM and type private IP of staging VM we will see IIS service page.

POINT TO SITE CONNECTION

Server Side

- We need gateway subnet. (10.1.2.0/27)
- We need Virtual Network gateway, charger acc. To hourly basis.

Client Side

- We need to have certificates in place.
- We can generate our own self-signed certificate.
- Export the public key.
- On client machine we need user certificate with private key.
- VPN client connect to user certi.
- Check my written notes for this, for diagram and description.

NOTES ON POINT-TO-SITE VPN CONNECTION

A Point-to-Site VPN connection is used to establish a secure connection between multiple client machines and an Azure virtual network via the Internet.

Below is a diagram from the Microsoft documentation on a sample scenario

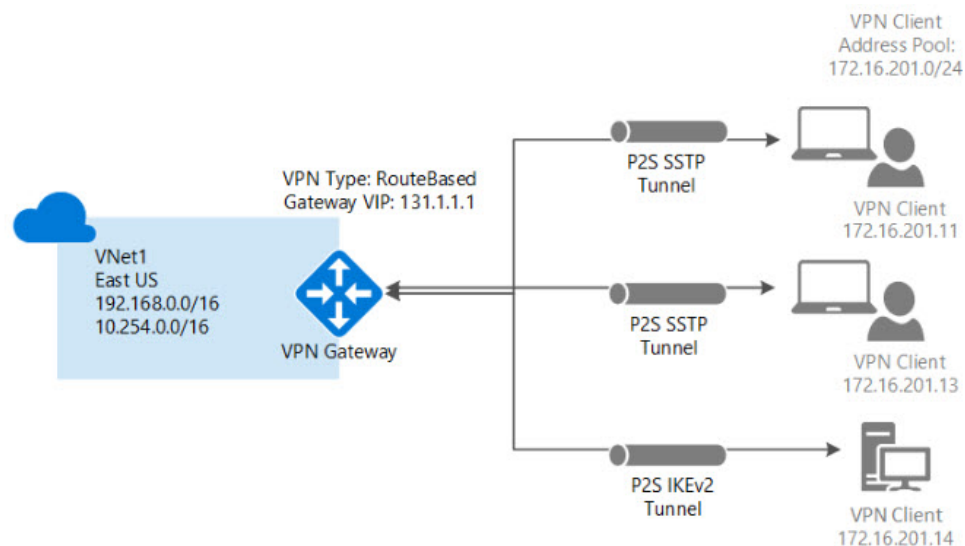


Image reference - <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-point-to-site-resource-manager-portal>

- This sort of connection is based off certificates for authentication.
- You need to have a root certificate in place that needs to be uploaded to Azure for the point-to-site connection.
- A client certificate needs to be generated from the root certificate. This client certificate needs to be on each client computer that needs to connect to the Azure virtual network via the Point-to-Site connection.
- To generate the certificates, you can use a Certificate authority or generate a self-signed certificate using PowerShell. Some commands are given below

// To generate the root certificate

```
$cert = New-SelfSignedCertificate -Type Custom -KeySpec Signature `
```

```
-Subject "CN=RootCertificate" -KeyExportPolicy Exportable `
```

```
-HashAlgorithm sha256 -KeyLength 2048 `
```

```
-CertStoreLocation "Cert:\CurrentUser\My" -KeyUsageProperty Sign -KeyUsage CertSign
```

// To generate the client certificate

```
New-SelfSignedCertificate -Type Custom -DnsName P2SChildCert -KeySpec Signature `
```

-Subject "CN=ClientCertificate" -KeyExportPolicy Exportable `

-HashAlgorithm sha256 -KeyLength 2048 `

-CertStoreLocation "Cert:\CurrentUser\My" `

-Signer \$cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")

Site To Site VPN Connections

- We want to connect azure network to on premise network instead of public we want secure connection using private IP address.so, we could do with the help of site-to-site connection.
- In On- premises side, we need to have a hardware router or software router for routing data to azure network on the datacenter side.
- In azure side, we need to have VM, gateway subnet, Virtual network gateway and a local gateway.
- Now refer notes for further diagram and lab setup.

Notes on Site-to-Site VPN Connection

A Site-to-Site VPN connection is used to establish a secure connection between an on-premises network and an Azure network via the Internet.

Below is a diagram from the Microsoft documentation on a sample scenario

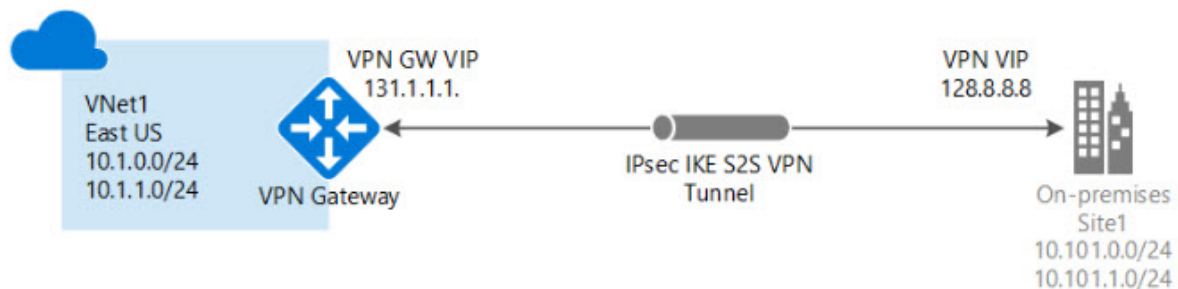


Image reference - <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-site-to-site-resource-manager-portal>

- On the on-premise side, you need to have a VPN device that can route traffic via the Internet onto the VPN gateway in Azure. The VPN device can be a hardware device like a Cisco router or a software device (e.g Windows Server 2016 running Routing and Remote services). The VPN device needs to have a publically routable IP address.
- The subnets in your on-premise network must not overlap with the subnets in your Azure virtual network
- The Site-to-Site VPN connection uses an IPsec tunnel to encrypt the traffic.

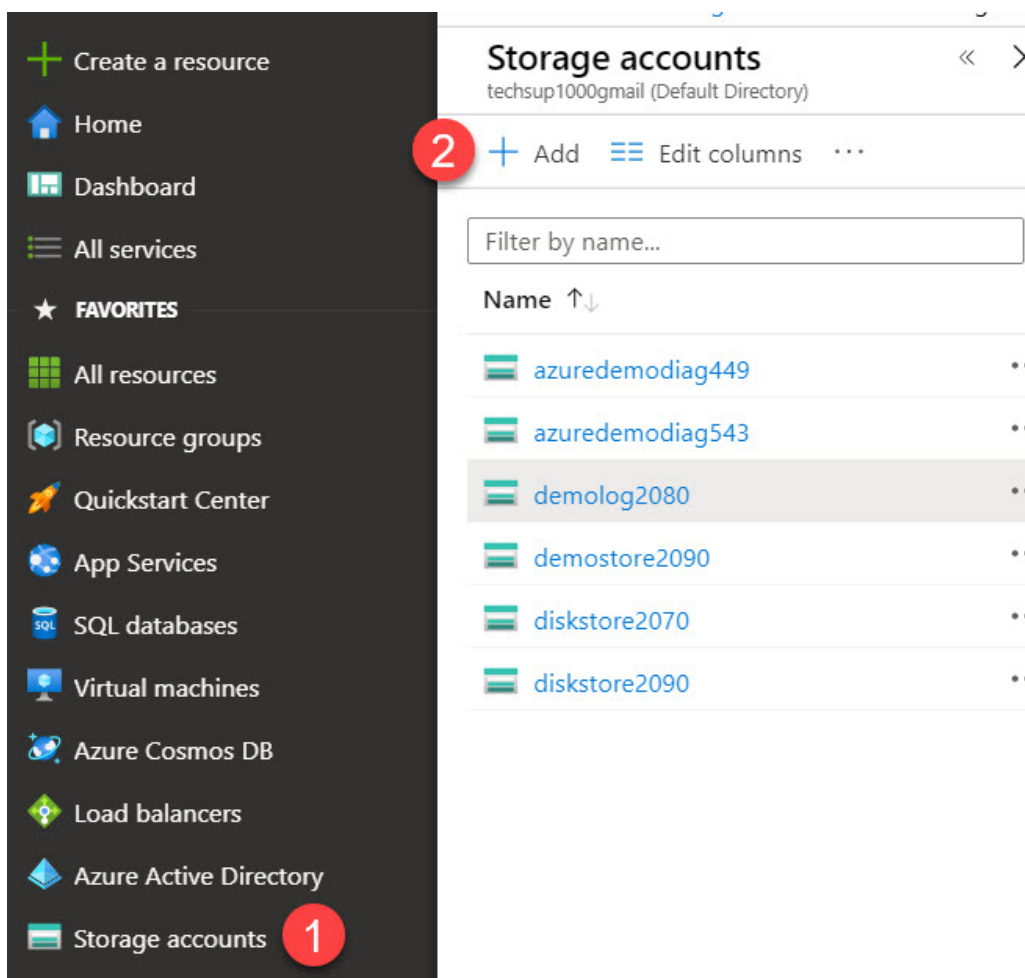
- The VPN gateway resource you create in Azure is used to route encrypted traffic between your on-premises data center and your Azure virtual network.
- There are different SKUs for the Azure VPN gateway service. Each SKU has a different pricing and attributes associated with it - Reference - <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings>

Creating a storage account

In the next chapter we are going to see how to use ARM templates to create a virtual machine. The storage account will be used to store the hard disks for the virtual machine.

Please ensure that you use the same storage account name in the ARM template in place of diskstore2070. Please view the next Lab and the resource attached to that chapter to see whether to replace the name of the storage account.

Step 1) Go to storage accounts and add a new storage account



2) Enter the following details in the screen for the properties of the storage account

Subscription * Pay-As-You-Go ▼

Resource group * 1 azuredemo ▼
[Create new](#)

Instance details

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

Storage account name * 2 diskstore8090 ✓

Location * 3 (US) Central US ▼

Performance 4 ☐ Standard ☒ Premium

Account kind 5 StorageV2 (general purpose v2) ▼

Replication 6 Locally-redundant storage (LRS) ▼

! Accounts with the selected kind, replication and performance type only support page blobs. Block blobs, append blobs, file shares, tables, and queues will not be available.

1. Choose the resource group
2. Give a unique name for the storage account
3. Ensure the location of the storage account is the same as that of the virtual machine you are going to create using the ARM template
4. Mark the Performance as Premium
5. Mark the Account kind as StorageV2
6. Mark the Replication as LRS

Step 3) You can go directly to Review and Create and then go ahead and create the storage account.

AZURE RESOURCE MANAGER TEMPLATES

- This provides ability to define infra as a code
- You create templates using json
- This defines the infra and configuration of the resources that need to be deployed.

DIFFERENT SECTIONS OF TEMPLATE:

- **Resources**-this is used to specify the resources that need to be deployed.
- **Variables**-these are values that can be reused in template.
- **Parameters**- this can be used to provide values during deployment phase.
- **Outputs**-this returns values from deployed resources.

If we go to any VM-->Export template-->this is a code for creating a VM in JSON format.

IN AZURE PORTAL SEARCH FOR TEMPLATE DEPLOYMENT.

- Create a new template
- Add code to create a virtual network.
- Resources section is very imp, in this define which resources we want, type as a VNet, name, location, properties (add. Space, address prefixes), subnets (name, properties (address prefixes))
- API version is imp, it decides how to mention properties and all.
- We can also use add resource button to simple create and easily get json script and understand that.

Lab-ARM Templates (Parameters)

- Create a new template
- Parameters help to give values at runtime.
- When we want to pass values at runtime than we can use it.
- Name of Vnet, address space etc.
- We need to provide a default value if user does not enter.
- When we save we will get option for mentioning name add. Etc which we define in parameters.

LAB-ARM TEMPLATES(VARIABLES)

- If we want to use values that we define in resources section, we can define in variable section.
- In resources section we can simply take reference from variable section.

ARM TEMPLATES FOR VM

- Check the template file(downloaded)

USE CASE SCENARIO

Resource Iteration

- You copy element in ARM template to make multiple copies of a resource.
- You can use the copyindex() function to return the current iteration on the loop. The value of copyindex() starts from 0.
- If you want to start from a particular index value-copyindex(1)
- You can also specify a mode for the deployment which can be serial or parallel.

- Add the template deployment.
- We are deploying storage acc. Using the template. We are using copy element and giving count=3 to create 3 SA.
- In resources section we will use copyindex() function also.
- Save and deploy krdo.
- We will see 3 SA's
- So, we can create multiple instances of a resource.

Lab - ARM Templates - Use case scenario - Resources

The following template can be used as a reference for the previous chapter

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": { },
  "resources": [
    {
      "name": "[concat(copyIndex(),'storage',uniqueString(resourceGroup().id))]",
      "type": "Microsoft.Storage/storageAccounts",
      "location": "[resourceGroup().location]",
      "apiVersion": "2019-06-01",
      "kind": "StorageV2",
      "sku": {
        "name": "Standard_LRS"
      },
      "dependsOn": [],
      "tags": { },
    }
  ]
}
```



```

    "properties": {},
    "copy":{
        "name":"storagecopy",
        "count":3,
        "mode":"Serial",
        "batchSize":1
    }
}
]
}

```

ARM TEMPLATE FOR IP ADDRESS:

- In this we have define location both in parameter and variable.
- In parameters we have defined a location parameter with value 'westus'
- In variable we are taking the location of resource grp same for IP, so in the end it will take the location of rh resource that we will specify and will create IP in that location only.

ARM TEMPLATES - USE CASE SCENARIO - RESOURCES

The following template can be used as a reference for the last chapter

```

{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "name":{
            "type":"String"

```

```
    },  
  
    "location":{  
        "defaultValue":"westus",  
        "type":"String"  
    }  
  
    },  
  
    "variables":{"location":"[resourceGroup().location]"},  
  
    "resources": [  
    {  
        "type":"Microsoft.Network/publicIPAddresses",  
        "apiVersion":"2019-11-01",  
        "name":"[parameters('name')]",  
        "location":"[variables('location')]",  
        "sku": {  
            "name":"Basic"  
        },  
        "properties":{  
            "publicIPAddressVersion":"IPv4",  
            "publicIPAllocationMethod":"Dynamic",  
            "idleTimeoutInMinutes":4,  
            "ipTags":[]  
        }  
    }  
    ]  
}
```

```
]
}
```

AZURE BACKUP SERVICE FOR AZURE VM

- We can use azure backup service to restore backup.
- Data of azure VM is stored in recovery services vault and it should be in same region of the VM.
- The first backup is taken from the required or the underlying disk only.
- But subsequent backups are become intelligent in nature and will only look at the changes which happened in the last backup and will backup that to RSV.
- We need to use the backup policy to configure the schedule for the backup process so, the backup can be taken at a particular time, and we will also mention the retention period.
- The recovery point is created when backup is created. For every time we create backup.
- When we want to recover data, few can choose any recovery point acc. To our need.
- We can recover files, VM and disk (replace or add new disk)

POINTS:

- This service is used to backup the disks allocated to the VM.
- During the first backup, an extension is installed on the VM.
- This extension is used to take a snapshot of the disks attached to the VM.
- For the windows-based v<'s the backup service works with windows volume shadow copy service to take an app-consistent snapshot of the VM.
- For Linux VM's the service takes a file consistent backup.

The different types of snapshots that can be taken by service

- **Application consistent**-here backup service captures the memory content pending I/O operations.
- **File System Consistent**-here the backup service takes snapshot of all the files a same time.
- **Crash consistent**-this happens the VM shuts down at the time of the backup process.

LAB

- In an VM, go to backup section.

- We need to create a recovery services vault; it will give option.
- And we need to create policy too (choose existing or new).
- We can also choose default policy
- Enable backup.

Backup Policy

- Policy name
- Backup schedule (frequency, time, timezone)
- Instant restore
- Retention range (daily weekly monthly etc.) (mention the day on which we want.
- Once backup is enabled
- We will have no restore points
- We can click backup now.
- We can see backup jobs as well.
- In vault we can see the backup items
- Once backup is complete, we can see the restore point now.
- We can restore VM and file recovery.

FILE RECOVERY

- We can choose our recovery point
- Download executable (it will create mount points)
- Click on executable type pswd, that is given
- It will open PowerShell
- And backup is complete, and we can check the file explorer
- Then unmount disks option on the portal after that.

RESTORE VM

- We can create new disks or replace the existing disks on the VM
- Restore type (restore disks)
- Mention RG
- And a storage acc.
- Click restore
- We will see the new disk and we can add this disk to our VM.

AZURE MARS AGENT

- Logon onto a VM or connect and open a azure portal on the Internet explorer.

- Go to RSV-->Backup-->on premises->files and folders (we can selectively backup using the agent)
- Click prepare infra
- Download the agent option click.
- Agents install hojayega
- No Microsoft update.
- Download vault file, in steps mention.
- Enter the details.
- Enter paraphrase as a secret it is needed.
- Ok and close.
- And it will launch Microsoft backup service and we can schedule our backup
- We can add items acc. We want, here we can choose selectively
- Choose the date and time
- Mention retention policy.
- Finish
- Then click backup now
- Next and finish
- Then click recover data, we can also recover the data onto other VM from this VM.

AZURE AD TOPICS

AZURE AD

- It is identity provider.
- We need a username and password to login in the company, it helps in authenticating.
- User can authenticate using Azure AD and can access azure resources.

POINTS

- It is Microsoft cloud-based identity and access mgmt. service.
- This allows users to sign in and access resources in the azure portal, Microsoft office 365 and other SaaS apps as well.
- There are many features available for azure ad in addition to just adding users and groups.
- You can also define apps that would need access to resources in azure.
- There are also different pricing models available for azure active directory.

LOOK AT AD:

- We get one instance of ad after creating account.
- We get a directory
- We can create users, groups and register devices.
- We can register apps

There are a lot of things in ad

- Creating users
- Go to AD→user's section
- Create new user
- Username
- First name
- Last name
- Tenant
- Username and pswd
- We can add user to a group or roles
- We can sign in using that user. It will ask to update pswd. And we will sign in, we will have no access to any resources as we haven't given any access.
- We can switch to our other user.
- We can give access to group and members in that will inherit all properties.

CUSTOM DOMAINS

- Username is appended with the directory name.
- To have a custom name we need to create a custom domain.
- We have bought different domain names.
- In AD-->custom domain name-->add domain
- We need details like host name, point to address, TTL, record type
- We can use GoDaddy to create a domain.
- And add domain to the azure ad above details fill krdo.

AZURE AD LICENCES

- Check documentation for the licences.
- We can use free trail for azure ad premium p2

Types of licenses: -

- Free
- Office 365
- Premium p1

- Premium p2

We can assign licenses to a user's or users.

We should define usage location than only we can assign licenses.

AZURE AD AND SUBSCRIPTIONS

- **Azure Tenant**- this is dedicated and trusted instance of azure AD.
- **Azure AD Directory**- Each azure tenant has dedicated and trusted Azure AD directory. This includes the tenant's users, groups and apps and is used for performing identity and access mgmt. onto resources.
- **Subscription** is grouping of resources or is use for billing purpose.
- 1 subscription can trust 1 directory. Users can have access to resources in a particular subscription.
- We can change directory any time.
- We can create new tenant and switch between tenant also.
- Tenant is a directory only.

MULTI-FACTOR AUTHENTICATION

- Additional layer of security
- We need username and pswd to login to the azure portal.
- It can be risky.
- So, we can add more security, like a SMS, email, call etc.

STATUS OF USER DURING MFA

- **Disabled**-the default for a user.
- **Enabled**-The user has been enrolled in azure MFA.
- **Enforced**-the use has completed the replication process.

LAB-MFA

- Go to AD-->Create a new user.
- Go on to security in a directory and we can go to MFA
- Go to users than where we will see all the users.
- We will see the status for each user, change the user status to enabled.
- Not sign in with that particular account, it will ask for more info
- It will ask how to contact you? We can use phone and message and give details.
- We will receive the code we will be verified, and we need to change the pswd as it is a new user and we will be signed in as that user.
- The status will change to enforced now.

MFA-TRUSTED DEVICES

- In MFA-->service settings
- We will have different methods for verification (message, mobile app etc.)
- We want that the user not go to the MFA process again as we trust the device.
- We can enable a setting in MFA in service setting of remember.
- Now try to login as a user, and we not asked for MFA process as it remembered device.

MFA-SKIP IPs

- We can have trusted IPs.
- That means the can use AD federation services. To skip MFA.
- Suppose we have a VM in east US location, we can take Public Ip and copy that paste in trusted IP setting in MFA.
- Then click save.
- Connect to that VM, and open internet Explorer, type azure.portal.com
- And then sign in as that user, we will not be asked for MFA.

CONDITIONAL ACCESS POLICIES

- Automate security for user authorization based on conditions

Different aspects to policy

- First aspect is "Who should the policy apply to".
- Second aspect cloud apps, select the apps means here we can say "Apply this policy to users who are trying to access Office 365 exchange online.

Next, we have conditions: azure tries to

SIGN IN RISK

- Determine if a legitimate user is trying to sign in.
- It then categorizes this based on severity levels
- For this you need to have azure ad identity protection in place.

DEVICE PLATFORMS

- So, here we are adding a condition based on platform.
- So here we are saying that if user is trying to log in from a windows device, then ensure to apply this policy.

LOCATIONS

- Here we are adding a condition-based n where user is trying to access azure from.

CLIENT APPS

- So here we are adding a condition based on whether the user is using a browser a mobile app or desktop clients.

FINALLY GRANT OF ACCESS

- So, here we are saying that if policy conditions are met, what authentication mechanisms should we apply.

LAB

- First create a user to use condition access policy
- Fill details and create.
- Login with that user to portal.
- Sign out krdo (user se)
- Fir main tenant, that we were using ussey AD-->condition access policy.

NEW POLICY

- **Assignments**-users or groups (we can include or exclude) can select acc. to roles. (Select the user we created)
- **Cloud apps or actions** (we can select apps, we want to apply when user use azure portal so need to use 'Microsoft azure Mgmt.)
- **Various Conditions**
 - Grant (enable MFA)
 - Session
 - Create krdo.
 - Now login as that user, it will ask for the more info, enter details and then verify and we will login to azure portal.)

So, with the policy with the condition we can create MFA etc.

AZURE AD IDENTITY PROTECTION

- This is a tool that automates and remediates identity-based risks.
- Here Microsoft uses its built intelligence system to detect any identity based risks.

Different types of risks that can be detected:

- **Anonymous IP address**-This happens when a user signs in from an anonymous IP address.
- **Atypical travel**- This happens if a user sign's in from a location that is not normally used by user for the sign-in process.
- **Unfamiliar sign in properties**- here the sign in properties is not the same as normally seen for the user.

We need Azure AD Premium P2 licence to use Azure AD identity Protection.

Search for identity protection in Azure Portal.:

- Overview will let us know if there are any risky users
- In reporting it helps us to give, and we can find risky users
- We can define policy we can define for all users, and we can define category (high, medium and low) and then we can acc. Block the access or allow acc.
- We have sign in risk policy (we can choose assignments and we can block or allow access and apply MFA)

ACCESS REVIEWS

- Now in Azure AD there can be many users and each user can have a role acc. User can be a part of User Group, Azure AD role, Azure Resource etc.
- Now it is very important that to review the access as a user can have change the dept. Suppose or move somewhere and then he does not need the access of the particular role that he has, so we can use access reviews for that.
- Check Microsoft doc for the access reviews.
- You need Azure AD Premium licences to work with Access Reviews.

LAB

- Define or create some users in AD and assign to a group or add to an new group.
- Search for identity governance-->access reviews
- New access review
- We can access review for 'Teams + Groups' or 'Applications'.
- Select which team or group
- Select the groups (we created)
- Select review scope (all users or guest users)
- Select reviewers (grp owner, selected user or group)-the reviewer should have azure AD premium licence

- Specify recurrence of review (duration (in days),start date and end date, weekly or.....,occurrences(specify a number)
- Next settings
- Upon completion settings
- Auto apply results (enable or disable)
- If reviewer does not respond (select what to do from list (Nochange,remove access approve access, take recommendations))
- Enable reviewer decision helpers
- No sign in within 30 days (enable or disable)
- Advanced settings
- Review and create access review.
- The reviewer will get an email and from where we can approve the review and deny review acc.
- We can stop the access review forcefully after giving the decision.
- After that we can apply the result that we have taken the decision.
- In the demogrp now we will have 1 user only as we have approved for 1 and deny for 1 that is why.

WORKING WITH MULTIPLE DIRECTORIES

To create a new tenant

- If we have may org. In company we can define users for each org differently.
- So, we can create different tenants
- We can switch the directories between the different tenants.
- But we need to have another subscription for the different tenants.
- We can delete directory or tenant acc. If we do not want.

AZURE AD CONNECT

- If a company has large number pf users defined in their AD, and it is difficult to move all users to map to Azure AD, so we can use Azure Ad connect to move all users or groups or workloads from company to Azure AD.
- We need to install Azure AD connect component on the server that have an AD and then sync it to Azure AD.

Types of Synchronization:

- **Password Hash Synchronisation**-the user is duplicated from the Ad to Azure AD and pswd also replicated and this is done by hash. then we will have a user and pswd on both the location. in this if user change pswd in AD it will replicated to the Azure AD.

- **Pass Through Authentication**- user login on both with same username and pswd, but authentication happens only on on-premises environment. first it will authenticate to on premise then to azure AD and then it will be able to access the resources of azure, it can be bcoz of security policy we want. This service is used to synchronize our users from local on-premises AD onto Azure AD not the other way around. in this if we want to change password on both sides, we need to enable **Password Writeback**.

POINTS

- The Azure AD Connect synchronisation service is used to synchronise identity data between your on-premises environment and Azure AD.
- There are 2 components for this service
 - **Azure AD Connect sync Component**- this is installed on the on-premises environment.
 - **Azure AD Connect sync Service**-this service runs in Azure AD
- To use Azure AD, connect, you need to have the following pre-requisites in place
 - An Azure AD tenants
 - You need to add and verify your domain in Azure AD.
 - Use the Idfix Tool to identify error such as duplicated and formatting problems in your on-premises directory.
 - The Azure AD connect Sync component must be installed on the windows server 2012 standard or better. The server must have the full GUI installed. The server must be domain joined. Ideally this component must not be installed on the domain controller.
- During the configuration of the Azure AD connect sync component, you need to use an
 - Azure AD Global Admin account for the Azure AD tenant. The account should be a school or organisation account and cannot be a Microsoft account.
 - An enterprise admin account for the on-premises AD.
- The Azure AD connect server needs the DNS resolution for both intranet and internet. The DNS server must be able to resolve names both to your on-premises AD and Azure AD endpoints.

Synchronisation Techniques

- **Password Hash Synchronisation**- Here Azure AD connect synchronises a hash, of the hash, of a user's password from an on-premises AD instance to a cloud based Azure AD instance.

The advantage is that you only need to maintain one pswd for authentication in you on premise env. and in cloud.

If you change a user's pswd on the premise ad the pswd will be synched onto azure AD.

- **Pass Through Authentication-** This is a kind of similar to password hash synchronisation, but here the user's pslds is directly validated against the on-premises AD. This allows orgs. to enforce their on-premises AD security and psld policies.

AD IMPLEMENTATION

- To setup AD domain.
- First Add a new VM, fill details (windows server 2019)
- Connect to VM using RDP file.
- Add roles and features, Add AD domain services.
- Promote server to DC, add new forest, give domain name, psld and then install.
- Restart the VM.
- In portal assign change the IP address to static. So, that is does not change for more secure.
- Now if we want other VM's to connect to the internal domain we created, we need to have a DNS server.
- So, Portal-VM-VNet-->DNS Servers section-->custom-->enter Private IP of our VM (as a custom DNS server) -->Save.
- Now restart VM from the portal-->connect t VM again-->use the domain name\user a username and enter the psld we set --> we will be able to login.
- Now create a new VM in same VNet.
- Connect to new VM using RDP
- And change the workgroup to domain for that machine, enter username and psld and we will connect that VM to our domain.
- In the domain VM we will see inside computer it, will show that VM.
- And in DNS also we will see the details.
- Now login to that new VM→ using a domain admin or user as it is connected to that domain.

AZURE AD CONNECT INSTALLATION

- Create a new VM, fill details, make sure it is in same VNet as the above 2 VM's.
- Connect to that VM using RDP.
- Local server-connect to the domain (we created)
- Go to the Domain VM-->AD users and computer-->create 2 new users
- Logon on to that 3rd VM using the domain credentials in RDP.
- Turn on IE enhanced config
- Go to Internet explorer-->download Ad connect-->run the installer.

- Now we need to have a Azure Global Admin t login in Azure AD connect credentials.
- In Portal we can adding a user to global admin and then enter those credentials
- Connect to the AD connect on the 3rd VM.
- Now enter the domain credentials and pswd,
- Now it will synchronisation and configure the synchronisation, start krdo
- Configuration complete.
- In portal go to azure ad we will see the users we created in our ad will be replicated or synched in azure ad.

AZURE AD PASSWORD HASH SYNCHRONISATION

- Spun up the windows 10 client machine and join this computer on to the domain.
- Connect to that client VM→ control panel-->system-->remote settings-->allow remote connect and ass the users we created in domain to the allow remote connection for that-->apply krdo-->disconnect krdo
- Ab vpi connect kro domain name\userb (we created in ad)
- Ab portal be sign in kro with the user b and login kro in azure.
- So, we have use the same pswd to login to portal also and to the client VM, so we can login with same pslds on both the environment.

GOING THROUGH AZURE AD CONNECT SETTINGS

- Logon on to the 3rd VM (AD Connect VM), azure ad connects Kholo, we can see synchronised devices, account, password hash sync enable h.
- Customize synch options-->pswd dalna pdega global admin ka
- We can see the directories that are synched
- We can see the domain
- We can see setting like pswd writeback and pswd hash sync, pswd writeback (disable krdo)
- Apply krdo
- For vpi Kholo azure Ad connect-->configure staging mode-->enable staging mode
- Staging mode--> it exports the data from AD that is going to be imported to Azure AD, but data does not get imported. It is used for testing, as org. Can want some users to sync only and not all, and it will tell which users will imported to azure AD, if our sync not happening check this setting.
 - (Process and sync ni hoga issey)
- Previous page --> choose change user sign in setting
- We can select sign on method-->pswd hash sync settings and pass through bhi h abhi ese hee chod do.

AZURE AD PASS THROUGH AUTHENTICATION

- In this request flows through Ad for accessing resource of an azure.
- When we want to replicate the security policies in our AD we want that it first check that policies and verify than can access resources than we use this.
- Logon onto AD Connect VM-->start Ad connect kholo-->change user sign in-->user sign in m select pass through authentication-->next --> enter the domain credentials.
- AD connect pe jao vpi in AD Connect VM only-->customize synch options-->pswd dalna pdega global admin ka
- We can see the directories that are synched
- We can see the domain
- We can see setting like pswd writeback and pswd hash sync, pswd writeback (disable krdo dono) save.
- Ab Logon to Domain VM-->AD users and computers--user b--Account tab-->deny logins for all the days for the user -->apply krdo-->Ok
- Login in portal using user b, it will give sign in failed as it is going from our AD

AZURE AD-SINGLE SIGN ON

- Logon permit krdena on domain VM on the userb that we changed above for all days
- Login the azure portal using the userb we will be able to login to portal.
- Sign out krdo portal se
- AD connect VM pe jao -->azure AD connect open kro-->user sign in-->choose single sign on-->forest credentials daal do -->configure krdo.
- Client VM pe jao with windows 10 -->internet options pe jao-->security-->local intranet-setting-->advanced-->sites-->enter URL for auto logon-->.add-->ok
- Open internet explorer-->login in azure portal using userb and we will login to azure portal, we will not need to enter the pswd, so, benefit of single sign-in.

PASSWORD WRITEBACK

- As an admin we can also change the user pswd from azure AD-->user-->rest pswd. But we need to replicate the pswd to Ad also so, that there is no mismatch.
- So, for to AD connect VM-->open azure ad connects software-->choose customize sync options-->global admin pswd-->next-->enable pswd writeback-->configure.
- Now whenever the pswd is change it will change for both sides.

DOMIAN-OU FILTERING

- Go to the domain VM-->Ad users and computer-->create a new OU-->and create new user inside it.
- Go to AD connect VM-->azure AD connect-->customise sync options-->global admin pswd-->domain OU filtering-->sync selected domains and choose the Only OU we created -->next-->configure
- Go to the Azure portal AD-->refresh we will not see any user we will only see the OU user we created inside an OU.

AZURE AD CONNECT HEALTH

- Portal-->Azure AD->azure AD connect section-->AD Connect health-->different issues how our azure AD is performing-->we can see sync errors. Or sync services.
- We can go to AD DS services we can download various agents-- and it will show the health and graph about our local AD how it is performing.

AZURE AD-DYNAMIC GROUPS

- We need to have azure AD premium licence for this.
- When we want users to add to a group automatically based on the user attributes, we can use dynamic groups.
- We can create user or device groups based on the user attributes or device attributes also automatically, using dynamic user and dynamic device.
- When we create a group there is an option in membership type (dynamic user, dynamic device, assigned)
- We can now add the dynamic query, we will see the attributes for the user and device.
- Like we choose the city contains Miami.
 - Now go to any user edit properties of users and change the city to 'Miami'
 - Now go to any user edit properties of users and change the city to 'miami'
 - Now go to any user edit properties of users and change the city to 'MIAMI'
- Go to groups-->go to your dynamic group-->we cannot manually add the user to this group
- We will see all the users will be added to group, it is not case sensitive.

AZURE AD-SELF SERVICE PASSWORD RESET

- Login with any user from the AD, we have an option for forget password, now instead of asking IT admin for resetting password we can use self-service password reset option, to reset password by our own self.
- So, for that we need to go to Azure AD-->password reset section
- Select the group or users for which we want to enable password reset.

- Go to authentication methods section-->we can various methods we can choose which methods we want user to enable pswd reset.
- We need to have Premium P2 licence for the enabling self-service reset pswd.
- We need to assign the licence to a user for which we are enabling the reset pswd setting for it. The user location should be defined than only we can assign the licence to that user or group.
- Now login using that user and it will ask for more info, we need to enter the email where we want the pswd to be sent and set the email.
- Then again sign in and choose forgot pswd and we will ask for the contact method to set new pswd.

AZURE MIGRATE SERVICE

AZURE MIGRATE SERVICE

- It can be used for assessment and migration of your on-premises workloads.
- Suppose company is using the virtualisation service like Hyper-V/VMWare for hosting VM"s and want to migrate to azure.
- We can use Azure Migrate service.
- Azure Migrate can be work with both Hyper-V and VMWare.
- With this we can do assessment first, means it will assess your on-premises workloads and, it will give us insights on what should you do like size, problem costs, whether it is on ready state or not the VM's and more.
- Then we can Perform a replication of the workloads
- You can then perform a test and final migration.

AZURE SITE RECOVERY SERVICE

- Suppose in primary region, we have workloads which are running on azure VM's or on physical servers or hyper or VMware.
- Now if the Site goes down we want to be ready with the secondary region.
- With the site recovery service, we can make sure that the data is being replicated to the secondary region.
- This can be done for physical servers or Azure VM's or any mix it is possible.
- We can also at a time do a test failover or planned failover from primary to secondary region.
- Main service used is azure site recovery service.

POINTS:

- When you setup Azure Site recovery for an Azure VM, the Azure VM continuously replicated onto a different target region.
- if an outage occurs, you can fail over the VM's onto secondary region.
- When all is fine in primary region, you can then fall back and continue operations in primary location.

What are all the components involved in disaster recovery for an Azure virtual machine.

- For the source region, you also need to create a SA. This becomes the cache SA. In the replication process, the VM changes are first stored in cache storage account before they are sent to target region.

The replication will create the following target resources: -

- A resource group that the target VM will belong to
- A new VNet that target VM will reside in
- If the source VM is using unmanaged disks, then a new storage account will be created in target region.
- If the source VM's are part of an availability set, a new availability set will be created in target region
- If the source VM is part of an availability zone, the same zone number will be allocated to the target VM in the target region.
- **Replication Policy**-The default policy has following settings:
 - **Recovery Point Retention**-Set to 24 hrs-this specifies how long the recovery services keep the recovery points.
 - **App-consistent snapshot**-set to every 4 hrs-this specifies how long the recovery service takes an application consistent snapshot.

PROCESS:

- Suppose we want to enable the site recovery for set of VM's
- And we have our VM's in east US region and east US goes down.
- and we want our VM in seconds up and running in central US location.
- For this we can use site recovery service.
- With backup service it will take time about 2 3 hrs., so we can't use this.
- Now 1 VM is using unmanaged storage to enable replication we need to have a cache storage also (all data changes are first send to cache storage then to secondary region)
- To enable replication.
- Now in target region we need to have a VNet also for hosting VM when doing failover.

So, in Source region what happens is:

- The Site recovery mobility service extension is installed on the source VM.
- Continuous replication then occurs via a cache storage Account.
- when the data is processed in target region, crash consistent reconvert points are generated every 5 minutes.
- Once we have retention point at target region then we can perform failover and then we will get VM's,

LAB

- Create VM with windows server, add role to thi VM-Web server role.
- In azure portal add the inbound port rule 80 for the VM.
- In VM-->go to notepad-->add some simple HTML.
- Go to file explorer-->inetpub-->wwwroot-->save that HTML files (all files as type).
- And now take public IP from Azure portal and type in browser-->IP/default.html(filename)
- It will show data of the html file.
- In VM-->site recovery section-->choose the target region(custom).
- Advanced settings (this section will create a subscription, RG, VNet, Availability, Storage settings (will create cache storage)
- Replication setting it will have own replication policy.
- Extension settings
- Agent will be installed on the VM
- Leave everything as it is.
- Go to the resource RSV, we will be replicated items-->it is replicating data to target region, at this time it is not creating a new VM, it is just replicating data.
- After the replication is complete in replicated items.
- Replicated items we will get option of test failover, now it will create a VM with the data which was replicated.
- We need to choose recovery point and the VNet-->then click ok
- It will now create a failover.
- Now in resources we can see we have 1 more VM in our resources.
- Go to that VM-->assign an IP address to that VM-->create new and associate krdo.
- Take IP/default.html -->we will see that text.
- Clean-up krdo test failover from replicated items ek brr.

AZURE MIGRATE FOR HYPER-V

- Suppose in our environment we have a Hyper-V environment in place based on windows 2016 and on this we have various machines.
- Now we want to migrate these VM's to the Azure VM's

- For solution we should first assess your on-premises workloads. So, in assessment will tell you what is running on the VM and acc. to that will tell what the specs of the VM are, you should create, for the replication and migration.
- It will also tell the approx. costs of the infra when we migrate to azure.

WHAT WE WILL DO IN LAB

- Now when it comes to Hyper -V we will deploy another VM on the hyper -V itself, it will be azure assessment appliance, it will take the inventory of all the hyper -v machine that are running on the hyper -v host, it will go ahead and give this info to azure migrate service and further the migrate service will give info such as what is sizing that you should consider when migrating the workloads from on premise env. and what is the proper cost.
- When we migrate a solution to azure, we need to install agent on hyper host and agent will replicate VM's from hyper v servers to azure. these all data will be stored in RSV in background.
- When we perform replication and migration to azure we need to have a SA and a VNet.

LAB

- Create a VM from azure Portal, with a appropriate size and connect to that VM.
- Now using PowerShell as admin install the hyper -v role on it, machine will be restarted
- After that from server mgr. open hyper v mgr.
- Using PowerShell create a V-switch
- Create an IP address using PowerShell.
- Create a new NAT
- Inside VM->local server-->IE enhanced Config--turn on
- Open Internet explorer and login to azure portal
- In azure portal go to azure migrate-->assess and migrate servers.
- Add tools-->fill subscription, RG, and project name
- Azure Migrate server mgr.
- Add tool
- Deploying project-->Azure Migrate server mgr.-discover-->choose the technology-->Hyper-V-->download migrate appliance-->save kro VM pe.
- Open that file and extract contents.
- Now go to hyper -v mgr.-->import VM option-->browse that appliance (select that folder) -->next -->next-->choose the V-switch we created-->create
- it will now create an assessment machine.
- Connect to that VM-->provide you own pswd.
- We will login into VM
- Go to control panel-->system-->remote settings-->allow remote connection.

- We need to assign an IP address to this assessment machine
- Control panel--net setting-->ethernet-->properties-->assign IP we created before-->and other details. -->close.
- Azure migrate service is running on this machine in background.
- It will install latest updates also.
- We need to register this machine with azure portal.
- Enter details username and pswd.
- In azure migrate service choose subscription and choose the migrate project-->enter appliance name-->register-->continue
- Give username of hyper v VM now and pswd. -save details
- Turn off the firewall for both VM's for private VM.
- Now in azure migrate service we need to add IP of our Hyper-V machine (copy krlo portal se) it will configure.
- Once discovery is complete
- In azure go to azure migrate service--> we will see discovered servers
- We will see our machine (it has detect the machine running on hyper v)
- Assess server krke details bhrdo and choose that VM.
- We will get an assessment--it will give cost estimate, estimate size and storage details etc.

AZURE MIGRATE -MIGRATION DEMO

- In the portal in azure migrate we will add migration tool--add
- Once added-->click discover
- Choose hyper v and target region
- Now it will create a site recovery service to replicate and store data.
- Now go to our VM where we have a hyper v role--open the azure migrate service on the browser-->servers-->download azure site recovery.exe--run krdo fir.
- Wizard opens and install krdo azure site recovery agent-->register kro completion ke bd
- Watch video.

Notes BY ALAN

AZURE MIGRATE AND AZURE SITE RECOVERY - REVIEW

AZURE MIGRATE

- The Azure Migrate tool provides a centralized solution to assess and migrate your on-premises servers, infrastructure, applications, and data to Azure.

- With Azure Migrate, you can assess and migrate your servers, databases, web applications, virtual desktops, and your data as well. The data can be moved using the Azure data Box set of products.

With the Azure Migrate Server Assessment tool you can do the following

- **Azure readiness** – You can assess whether the on-premises machines are ready for migration.
- **Azure sizing** – This helps to estimate the size of Azure VMs required for the migration.
- **Azure cost estimation** – Estimate the cost for running the servers in Azure.

You can migrate your On-premises VMware VMs, On-premises Hyper-V VMs and On-premises physical servers

AZURE SITE RECOVERY

- This is a business continuity and disaster recovery strategy.
- It can be used to protect your business applications and workloads to continue running in the case of outages.
- This service can be used to replicate your workloads from a primary region onto a secondary region.
- If your primary region fails for any reason, you should be able to switch over to the secondary region.
- You can replicate your Azure Virtual Machines.
- You can replicate your VMware VMs, Hyper-V VMs and your physical servers as well.

IMPORTANT EXAM POINTS WHEN YOU NEED TO CONSIDER USING AZURE SITE RECOVERY

The source virtual machine must conform to certain requirements

- Operating system – Windows Server 2019, Windows Server 2016 64-bit, Windows Server 2012 R2/ Windows Server 2012, Windows Server 2008 R2 with SP1 onwards, Windows 10, Windows 8.1, Windows 8 and Windows 7 with SP1 64-bit.
- Linux – Red Hat Enterprise, CentOS, Ubuntu, Debian, SUSE Linux, Oracle Linux.
- OS Disk – up to 2,048 GB.
- Data disk size – 8,192 GB when replicating to managed disk. 4,095 GB when replicating to storage accounts.
- Bit Locker must be disabled before replication is enabled.

AZURE TO AZURE DISASTER RECOVERY FOR AZURE VIRTUAL MACHINES

- When you setup Azure Site Recovery for an Azure virtual machine, the Azure VM continuously replicates onto a different target region.
- If an outage occurs, you can fail over the VM's onto a secondary region.

- When all is fine in the primary region, you can then fail back and continue operations in the primary location.

What are all the components involved in disaster recovery for an Azure virtual machine.

- For the source region, you also need to create a storage account. This becomes the cache storage account. In the replication process, the VM changes are first stored in the cache storage account before they are sent to the target region.
- The replication will create the following target resources
 - A resource group that the target VM will belong to
 - A new virtual network that the target VM will reside in
 - If the source virtual machine is using unmanaged disks , then a new storage account will be created in the target region.
 - If the source VM's are part of an availability set, a new availability set will be created in the target region.
 - If the source VM is part of an availability zone, the same zone number will be allocated to the target VM in the target region

YOU ALSO HAVE TO DEFINE THE REPLICATION POLICY

- **Replication policy** – The default policy has the following settings
- Recovery point retention – set to 24 hours – This specifies how long the recovery services keep the recovery points.
- **App-consistent snapshot** – set to every 4 hours – This specifies how long the recovery services takes an application consistent snapshot.
 - App-consistent snapshots are taken in accordance with the frequency you specify. This frequency should always be less than you set for retaining recovery points. For example, if you retain recovery points using the default setting of 24 hours, you should set the frequency at less than 24 hours.
 - They're more complex and take longer to complete than crash-consistent snapshots.
 - They affect the performance of apps running on a VM enabled for replication.
- **A crash consistent snapshot** captures data that was on the disk when the snapshot was taken. It doesn't include anything in memory.

AZURE LOAD BALANCER

AZURE LOAD BALANCER

- Suppose there is a VNet and there is a subnet, and we have 2 VM's inside a subnet. And we are hosting a web app on these VM's, we want the users to be equally distributed across both VM's.
- We can use Load balancer, now user instead of directly hitting the web app will hit the load balancer (distribution of traffic).
- Load balancer should be on a VNet.

We need to configure for LB:

- **Backend Pool**- it is VM's only where to direct the end user request after balancing the load.
- **Frontend IP**-it is IP, which is expose to the users, users will be hitting this IP
- **Health Probe**-it is used by LB to understand whether VM's are healthy or not. it is a type of heartbeat, which will tell LB to please at regular intervals, send the request to on a particular number if we get respond back in a interval than it is healthy. To determine VM is healthy or not.
- **Load Balancing Rules**-it is used to tell VM, that request coming from frontend Ip and we need to send it to backend pool, but on which port VM listens to, than we create Rule on that port, than direct traffic to VM. It is telling where to send the traffic.

ADDITIONAL POINTS

There are 2 types of SKUs:

- **Basic LB**- we can have single VM or machines at part of Availability set or machines are part of scale set.
- **Standard LB**- We can have multiple VM's, machines at part of Availability set or machines are part of scale set.
 - For Higher SLA-99.99% we can use standard LB.

TYPES OF LB

- We have web server in a subnet and DB server in another subnet.
- We can have a LB for web server, so that if users from internet try to access the web server, they can be distributed to VM's equally and load is distributed. This is an **External LB or Public LB**.
- As users are accessing from public facing internet or public domain that is why it is external LB.
- Now we can have a LB for DB server. the VM in web server can talk to DB server using the LB which is internal domain so, **Internal LB**.

LAB-BASIC SETUP

SCENARIO

- We will spin up 2 VM based on windows server 2019
- Both VM's need to be part of availability set.
- Install internet information services on both VM
- Put a simple default.html file as home page on web server.
- Create a Public IP which we need to assign to our LB.
- Create a LB
- We need to create a backend Pool (which are the VM's Only.
- Then we will create a health probe for the LB.
- Then create a load balancing rule, so that request can be routed to both the VM's.

LAB

- Create VM, also make it part of Availability set (name, fault domain (2), update domains (2)) enable port HTTP (for installing web server) and RDP, let us create a new VNet, and allow to create a public IP (we need this to install IIS services and run default.html file) -->create krdo.
- Create another VM, same region, choose availability set, enable same ports, same VNet and create this VM.
- Connect onto the demovm1-->install IIS Role from server manager.
- From portal take Public IP of the demovm1 and copy to browser we will see IIS service page.
- Open notepad add some HTML textTdemovm1) -->save as now browse-->in file explorer-->windows C-->inetpub-->wwwroot-->save default.html (all files krdo).
- Same set of steps on the demovm2, like for demovm1
- We can check Public IP/default.html for check krlo.
- Remove the Public IP from both VM's as we don't need as users will use LB public IP.
- Go to demovm1-->networking-->NIC-->Ip config section-->disassociate IP-->save.
- Same for the demovm2, remove Public IP.
- In Azure portal search for IP-->Create a new Public IP-->basic-->name-->static-->Subscription-->RG-->same location as VM-->create krdo.
- Add resource-->LB-->create-->Name-->same region as VM-->public-->choose our IP-->tags-->create.
- LB-->Frontend IP section-->we will have public IP
- Backend Pool section-->Name-->VNet-->IPv4-->VM's-->select the VM's-->add-->add krdo.
- Health Probes section-->create-->name-->Protocol (TCP)-->Port (80) -->interval (5 sec) -->threshold (2 failures) -->create

- Load balancing rule-->Create-->Name-->Frontend Ip-->Protocol-->Port (80) -->backend Pool (choose kro) -->health Probe(choose)-->Ok
- Now from overview section of LB take Public Ip and type on browser-->public IP of LB/Default.html
- It will forward request to either of the VM.
- We will also see the public IP at overview section of both VM's of the LB.

LAB-STANDARD SETUP

SCENARIO

- We will spin up 2 VM based on windows server 2019
- Both VM's need not to be part of availability set.
- Install internet information services on both VM
- Put a simple default.html file as home page on web server.
- Create a Public IP which we need to assign to our LB.
- Create a LB
- We need to create a backend Pool (which are the VM's Only).
- Then we will create a health probe for the LB.
- Then create a load balancing rule, so that request can be routed to both the VM's.

LAB

- Create VM, enable port HTTP (for installing web server) and RDP, let us create a new VNet, and allow to create a public IP (we need this to install IIS services and run default.html file)-->create krdo.
- Create another VM, same region, enable same ports, same VNet and create this VM.
- Connect onto the demovm3-->install IIS Role from server manager.
- From portal take Public IP of the demovm1 and copy to browser we will see IIS service page.
- Open notepad add some HTML textTdemovm3) -->save as now browse-->in file explorer-->windows C-->inetpub-->wwwroot-->save default.html (all files krdo).
- Same set of steps on the demovm4, like for demovm3
- We can check Public IP/default.html for check krlo.
- Remove the Public IP from both VM's as we don't need as users will use LB public IP.
- Go to demovm3-->networking-->NIC-->Ip config section-->disassociate IP-->save.
- Same for the demovm4, remove Public IP.
- In Azure portal search for IP-->Create a new Public IP-->SKU (standard)-->name-->static-->Subscription-->RG-->same location as VM-->create krdo.

- Add resource-->LB-->create-->Name-->same region as VM-->Standard SKU-->choose our IP-->tags-->create.
- LB-->Frontend IP section-->we will have public IP
- Backend Pool section-->Name-->VNet-->IPv4-->add VM's-->select the VM's-->add-->add krdo.
- Health Probes section-->create-->name-->Protocol (TCP)-->Port (80) -->interval (5 sec) -->threshold (2 failures) -->create
- Load balancing rule-->Create-->Name-->Frontend IP-->Protocol-->Port (80) -->backend Pool (choose kro) -->health Probe(choose)-->Ok
- Now from overview section of LB take Public IP and type on browser-->public IP of LB/Default.html
- It will forward request to either of the VM.
- We will also see the public IP at overview section of both VM's of the LB.

AZURE APPLICATION GATEWAY SERVICE

AZURE APPLICATION GATEWAY SERVICE

- This is a service which can load balance a traffic at OS layer 7.
- The users will hit application gateway service via frontend IP.
- Then we can create various listeners and rules to route traffic onto backend pool.
- Backend pool can have various endpoints like we can route traffic to VM, Azure App Service, On-premises server.
- We need to have an empty subnet as part of VNet to deploy the application gateway in which or as part of same VNet.

BENEFIT

- We want to ensure that there is a firewall to protect or can be denied our app from the attacks and all we can use Application gateway.
- It has ability to route traffic based on what is the URL of the request itself.

Example:

- Like user is making request on cloudportalhub.com and we have an app running on VM and 1 VM is responsible for processing the images and another VM is used for processing Videos, remember we will have different URLs for both, and we want those requests to go on a different VM's when you can do that with help of AG, if Images than 1st VM and for videos 2nd VM

POINTS

- This service is a web traffic LB that is used to distribute traffic to web apps.
- The web apps can reside on VM's, VMSS or even on-premises servers.

- The application gateway is an OS layer 7 LB.
- SECURE SOCKETS LAYER(SSL/TLS) TERMINATION
 - Here requests to the AG can be secure.
 - And then requests to backend pool resources can go unencrypted.
 - This can lift the burden of the backend pool for decrypting requests.
 - the decryption of requests can be left to AG resource.
- You can also enable autoscaling for your AG resource.
 - This allows the AG to scale up or down based on traffic load patterns.
 - You can also enable the Web Application Firewall feature for the Application gateway resource.
- You can also enable session affinity which allows a user session to be directed to the same server for processing. If the state of the user session is stored on the server, then this can be a useful feature.

DIFFERENT COMPONENTS OF AG:

- **Frontend IP address**- Users will hit the AG via a frontend IP.
- **Listener**-This is a logical entity that checks for incoming connection requests. There can be multiple instances attached to an AG.
- **There are 2 types of Listener Configurations:**
 - **Basic**-here the listener listens to a single domain site.
 - **Multi-site**-here the listeners map to multiple domain sites.
- **Routing Rules**-This is used to route traffic from listener to the backend pool.

There are 2 types of routing rules:

- **Basic**-Here all requests are routed to backend pool directly.
 - **Path-based**-here requests are routed to backend pool based on URL in the request.
- **Backend Pools**-These can be Network interface cards, VMSS, Public or internal IP, FQDN or backends such as App Service.
- **Health Probes**- This defines how the AG will monitor the health of the resources in the backend pool.

LAB

SCENARIO

- We will create 2 VM's and install IIS services on this page. And we will add pages in different folders, on the VM's.
- We will spin AG and we will implement URL based routing that will ensure to add VM's as part of Backend pool of the AG.

LAB-IMPLEMENT

- Create a new VM, add port RDP and HTTP for installing IIS services.
- create a new network in VM and add a subnet and we should have an empty subnet also, so we need to have an address pool of good range.
- Create a new VM, in same region and same VNet and allow both ports.
- Connect to appvm1 using RDP file, now add web server role and install it.
- start notepad-->text (this is a video server) -->in file explorer save the html file at inetpub-->wwwroot-->create a folder video-->save the notepad file in that folder.
- take Public IP of the appvm1 from portal and on browser type public IP/videos/default.html
- It will show us the HTML text.
- repeat the same steps for appvm2, but in this create a folder image and inside that folder save file default.html.
- take Public IP of the appvm2 from portal and on browser type public IP/images/default.html
- It will show us the HTML text (this is an image server)
- Now create an AG resource-->subscription-->RG-->region(same)-->tier (standard v2) -->we can enable autoscaling
- Go to the VNet in which VM's are created and add a subnet for application gateway.
- then select VNet and subnet in AG -->assign Public IP (frontend IP) -->create krdo vhin-->backends (add VM's to our pool) -->we will create 2 pool-->name (video pool) -->choose the VM (appvm1)-->also mention the NIC of the appvm1. and add another pool for images also.
- Now we need to add routing rule-->add-->name of listener-->select our frontend IP-->select port.
- Backend targets-->name -->port (80) -->add krdo.
- now add path -->/videos/*-->choose the target VM
- do same for images. path.
- we get all things now and create the AG.
- Go to resource-->copy Public IP-->in browser -->Public IP of AG/images/default.html
- we will see the text, check for both.

AG-WEB APPLICATION FIREWALL

- Go to the AG resource--web Application Firewall section-->configure krdo.
- Now create a WAF policy in portal search for it.

BASICS TAB

- Policy for(Global WAF(front door)/regional WAF(application Gateway)/Azure CDN(preview))
- Subscription
- RG
- Location

- Policy name

POLICY SETTING TAB

- leave as it is

MANAGED RULES

- leave as it is

CUSTOM RULES TAB

- Add custom rule
- Choose IP of our workstation
- Rule name
- Priority (100)

ASSOCIATION TAB

- Associate with our application gateway
- We can also add HTTP listeners(leave)

TAGS TAB

- Leave as it is

REVIEW AND CREATE TAB

- Create krdo

Now if we try to access the Public IP of AG/images/default.html on browser from our laptop will we denied access (403 forbidden error).

AZURE APPLICATION GATEWAY

- This service is a web traffic load balancer that is used to distribute traffic to web applications.
- The web applications can reside on Virtual Machines, Virtual Machine Scale sets or even on on-premises servers.
- The Application gateway is an OSI Layer 7 load balancer.

Some of the features of the Azure Application Gateway Service

SECURE SOCKETS LAYER (SSL/TLS) TERMINATION

- Here requests to the Application Gateway can be secure.
- And then the requests to the backend pool resources can go unencrypted.
- This can lift the burden of the backend pool for decrypting requests.
- The decryption of requests can be left to the Application gateway resource.

Autoscaling

- Here the Application Gateway can scale based on demand.
- You can also distribute the deployment of the Application Gateway across multiple zones to ensure better availability of the Azure Application Gateway service

WEB APPLICATION FIREWALL

- This feature provides protection of your web applications against common exploits and attacks from the Internet
- Here your application could be protected against SQL injection and cross-site scripting attacks
- The Web Application Firewall uses a set of rules to protect your web applications. These rules are based on the Open Web Application Security Project. These rules are automatically updated to ensure all the latest threats are included in the rules.
- You can also create your own custom policies for your Web Application Firewall

COMPONENTS OF THE AZURE APPLICATION GATEWAY

- **Frontend IP address** – Users will hit the Application Gateway via the Frontend IP address.
- **Listener** – This is a logical entity that checks for incoming connection requests. There can be multiple listeners attached to an application gateway.
There are 2 types of Listener configurations
 - **Basic** – Here the listener listens to a single domain site
 - **Multi-site** – Here the listeners maps to multiple domain sites.
- **Routing Rules** – This is used to route the traffic from the listener to the backend pool.
There are 2 types of routing rules
 - **Basic** – Here all requests are routed to backend pool directly.
 - **Path-based** – Here requests are routed to the backend pool based on the URL in the request.
- **Backend pools** – These can be Network Interface cards, Virtual Machine scale sets, Public or Internal IP addresses , FQDN or backends such as App Service.
- **Health Probes** – This defines how the application gateway will monitor the health of the resources in the backend pool.

DIFFERENCE BETWEEN AZURE LB AND AG

- AG-balance traffic at OS layer 7
- LB-balance traffic at OS layer 4
- OS layer description
- first from machine(source) to destination is goes acc. to OS layer model.
- Both source and destination we have same set of layers

- Application is browser where we type google.com
- then we get respond at session layer
- the request is divided into packets at TCP layer
- and these packets are sent to physical layer and then transferred to the physical layer of google server and all process happens again.
- This is raw representation of data transferring from source to destination.
- LB only looking at network connection layer and not looking at the data which is to send but the AG will look at this info also, we can see through the Wireshark tool.
- We can inspect page any page in google we can see that data in more understandable way and AG is able to route the request acc. to URL path also.
- LB is much faster than AG at it simply route request acc. to network layer but AG, if it needs to route request, it first needs to have info available at layer 7, data will be constructed from layer 1 to layer 7, so it takes time, but it is more intelligent than LB.

AZURE TRAFFIC MANAGER

AZURE TRAFFIC MANAGER

- It can be used to route request from users to the DNS routing.
- We have a VM on which our app is running in Us-central, we also have a azure web app running in different region, we also have another endpoint we app running in our on premise env.
- We can have different routing methods to route the request from the users.
 - **Priority**--we want to route all the users to be directed on the VM, we would create traffic manager profile, and we will create multiple endpoints for each application, and we will give priority 1 to that VM and users will be directed on the VM, if priority 1 fails we will route users to other endpoint having priority 2 and so on.
 - **Weighted**--we can say 30% of traffic to this endpoint and 30% to another and remaining 40% of users to this endpoint.
 - **Performance**--based on least latency, we have our endpoints located in different regions, so if users n us central we can route user to VM, and where the latency is less it will route acc.
- When we want to have a routing, solution based on DNS and based on different routing methods we can use traffic manager.

POINTS

- DNS based traffic LB.
- Allows for the distribution of traffic across multiple endpoints-The endpoints could be either in azure or outside of Azure. The endpoint just needs to be an internet facing service.
- Different Routing methods:

- Priority-This allows for another endpoint to become available if primary endpoint goes down.
- Weighted-This is when you want to distribute traffic based on weightage.
- Performance-This is when you want users to get the closest endpoint in terms of latency,
- Different Routing Methods
 - Geographic-here the users are directed to specific endpoints based on geographic location of the DNS query being made.
 - Multivalue-here all healthy endpoints are sent to the client.
 - Subnet-here a set of end user IP address are sent to specific endpoint.

LAB-TRAFFIC MGR

- Create an Azure Web app-->name-->Subscription-->RG-->region->App service plan-->standard plan-->create krdo.
- Create another Web app-->change the region bki sab same.
- In visual studio--create project-->ASP.Net app-->create krdo-->index file m-->welcome message m primary web app krdo-->publish app-->azure-->login in azure-->select subscription-->RG-->choose the azure web app-->publish krdo.
- Same for other we app, welcome msg (secondary we app) -->publish krdo-->we can see on browser the text coming and we app working for both.
- Create a traffic mgr. profile-->name-->routing method (Priority)-->subscription-->RG-->no location9as work at global level)
- Go to resource-->endpoints section-->add-->name-->type (azure endpoint) -->target resource type(App service)-->choose our app(staging(1st app))-->priority(1)--Ok
- Add for 2 and VM endpoint by following above step.
- Traffic mgr.-->config section-->protocol (TCP)--port (80) -->save
- Once endpoint become online
- In overview section of Traffic mgr. take the DNS name and paste in browser we will see our staging web app text.
- Staging app ko stop krdo fir ab vpis DNS name daalo of traffic mgr. and run kro ab secondary app ka ayega

AZURE TRAFFIC MGR-REAL USER MEASUREMENTS

- This is useful when using performance routing method.
- We can use these measurements to help endpoint with least latency, the azure traffic mgr. has built in feature in this but we could also help with this measurements, for better decision.
- Traffic mgr.-->real user measurements section-- take code and embed it.

AZURE FRONT DOOR SERVICE

AZURE FRONT DOOR SERVICE

- This is routing service that helps accelerate your app performance of your endpoints.
- This service works at layer 7 or HTTP/HTTPS.
- This service will route your client requests to the fastest and most available application backend.
- An application backend is any internet facing service that could be hosted inside or outside of Azure.
- URL Based Routing:
 - Here you can route traffic your backend servers based on URL paths of the request.
 - If you had a web app hosted via you domain URL of `http://cloudhublearning.com`, you could direct requests for `http://cloudhublearning.com/images/*` to one set of servers and other requests for `http://cloudhublearning.com/videos/*` to another set of servers.
- Multiple-Site Hosting-
 - Here you can configure more than on web site on the same front door configuration.
 - Session Affinity-here you can keep user session attached to the same application backend.
- SSL Termination
 - Here the SSL connections can be terminated at front door itself rather than being processed by backend servers.
- Web Application Firewall
 - You can use this feature to help protect your web application from internet-based attacks.

SCENARIO

- Create 2 azure web apps, in diff. regions
- Then publish one app using visual studio
- We will setup Azure front door service
- Need to implement Frontend IP.
- Users will now be hitting this IP of Azure front door service.
- We will define a backend pool which contains our web apps, we can also define here the weightage of apps or priority for both apps.
- Further we will create a routing rule which will tell when request come route it to the backend pool.

LAB

- Create 1 Azure web apps, choose basic app service plan, choose runtime stack as ASP.NET
- Create another web app but in different region.

- On visual studio publish app on azure using us acc. and mention welcome msg as this is productions app.
- publish the 2nd web app also using visual studio.
- Creating a Front door service-->subscription-->RG-->frontend IP-->add-->host name-->we can also enable WAF-->add krdo
- Adding backend Pool-->Pool name-->host type (App service) -->choose the production app-->HTTP port (80) -->HTTPS port (443) -->priority (1) -->weight(50)-->status(enabled).-->add.
- Add another backend also for staging app.-->priority (2) --Weight (100) -->add (bki same as above)
- Add health probes-->Path (/) -->Protocol (HTTP)-->interval (30) -->add
- add routing rule-->frontend IP select kro-->backend pool select krdo-->add
- Review and create
- Go to the front door resource
- and take the DNS name and copy it on browser we will get home page for production app.
- Ab ek ko stop krke check kr skte.
- and now check again the URL (staging app)

POINTS

Difference between front door service and Azure AG

- Front door service is combination of LB and AG.
- In AG it is regional resource whereas front door service is global service, we also need a VNet and empty subnet for AG for balancing load but in front door service we don't need to have any region or subnet
- Azure front door provides path-based routing at the global level and behind front door we can also have AG services and further that AG can do path based routing VNet level.
- You can also achieve 100% TLS/SSL offload.
- We can also have LB to sit behind the front door service, but we need Public IP for the LB.

AZURE FIREWALL

AZURE FIREWALL

- It is used to filter traffic from internet or flow outside the internet.
- It could be hardware service, or it could be install as a service.
- In azure we can use this service and can be install on the VM's and that can be used to filter traffic from internet to the VM's
- Main purpose of firewall is to have intelligent based system to understand if there are some attacks that can happen from internet.

- Like in NSG we can define inbound and outbound rule but these we define manually, but firewall are smart systems which help us to protect from attacks.
- NSG is one line of defense.
- In azure firewall we don't need to manage the underlying infra for deploying the firewall, it is completely managed service and is part of VNet.

POINTS:

- It has built in high availability.
- Can deploy the Azure firewall instance across 2 or more availability zones-99.99% SLA.
- You can filter traffic based on fully qualified domain names.
- You can also create network filtering rules -based on source and destination IP, port, and protocol.
- It is stateful in nature, so it understands what packets of data to allow.
- It has built in threat intelligence-here you can get alerts or deny traffic from/to malicious IP and domains

LAB

SCENARIO

- The VNet in which we want to create Firewall, it needs to have empty subnet and with name Firewall subnet.
- Private IP of firewall will be used to communicate to VM and Public IP will be used to connect to internet.
- The request from internet to VM will pass through firewall now.
- We need to create a route table that all the traffic from subnet needs to be routed via Azure Firewall Service.
- Create Route table and assign it to the subnet hosting the VM.

LAB-AZURE FIREWALL

- Create a VM-->fill details-->not allow any public ports and communication will be done with private IP.
- Create a VNet and choose address range of 10.2.0.0/16
- create 2 subnets subnetA910.2.0.0/24) and other AzureFirewallSubnet (10.2.1.0/24).
- not Assign the Public IP.
- Create krdo.
- Now create an azure firewall service-->create-->subscription-->RG-->region-->tier (Standard)-->choose the VNet-->create a public IP-->create the firewall.
- In portal search for Route table-->create-->RG-->region same as VNet-->Name-->create krdo.

- Go to route table resource-->add-->Route name-->address prefix (internet ka daaldo (0.0.0.0/0)) -->next hop type (virtual appliance) -->next hop address(Private IP of firewall)-->add krdo route table.
- Route table--Subnets-->associate-->VNet-->subnetA-->add.

LAB AZURE FIREWALL-NAT RULE

- Go to firewall-->Rules section-->add NAT Rule collection-->Rule name-->Source (IP)-->source IP (our workstation) -->destination address (public IP of firewall) -->destination ports(4000,koi bhi daaldo)-->Translated address(Private IP of VM)--translated Port(3389).
- Add that rule now.
- Now on PC open cmd-->/console--> to have RDP box-->type IP for the firewall(public):4000-->connect
- Add credentials of the VM now, we will be connected.

AZURE FIREWALL: APPLICATION RULES

- On the VM-->open Internet explorer-->we are not able to open any site as all requests are passing through firewall.
- Now we need to add rules to allow these things into firewall.
- Go to firewall resource -->Rules section-->Add Application Rule collection-->name-->Priority (100) -->Action(allow)-->Source type (IP)-->Source IP address (private IP of VM) -->Protocol: port(http, https)-->target FQDN's(www.microsoft.com)-->add
- Now on VM try the Microsoft site--> we will be able to login.

Notes on Azure Traffic Manager

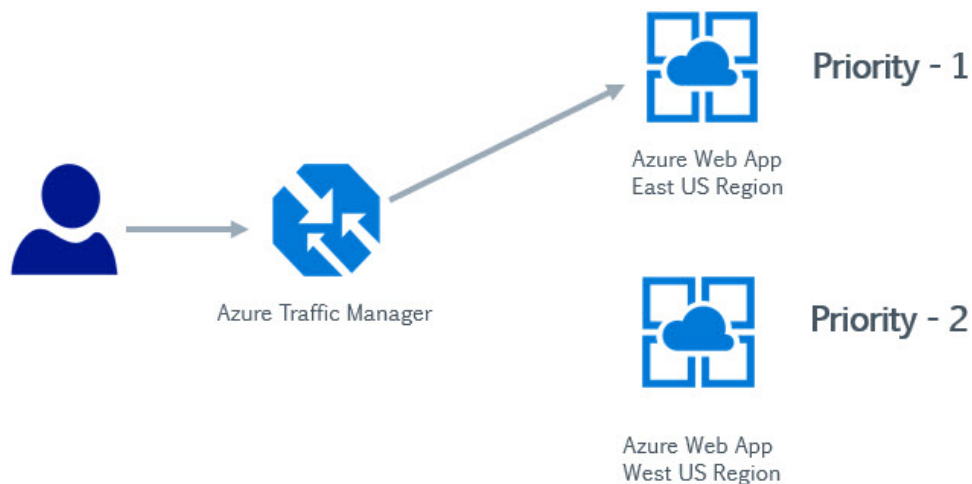
AZURE TRAFFIC MANAGER

- The Azure Traffic Manager service is a DNS-based traffic load balancer that distributes traffic across services that are distributed across different Azure regions.
- The Traffic Manager service is used to direct client requests to the most appropriate service endpoint that is based on a traffic-routing method and the health of the endpoints.
- The different traffic routing methods available for the Azure Traffic Manager are
 - **Priority** – Route traffic to another endpoint in case the primary fails.
 - **Weighted** – Route traffic to different endpoints based on weight.
 - **Performance** - you want end users to use the "closest" endpoint in terms of the lowest network latency.
 - **Geographic** - geographic location their DNS query originates from.
 - **Multivalue** – Here different endpoints are sent to the client. The client then selects the endpoint to send the request to.

- **Subnet** – This maps a set of end-user IP address ranges to a specific endpoint within a Traffic Manager profile.

Below is an example of the Priority routing method that can be used with the Azure Traffic Manager service

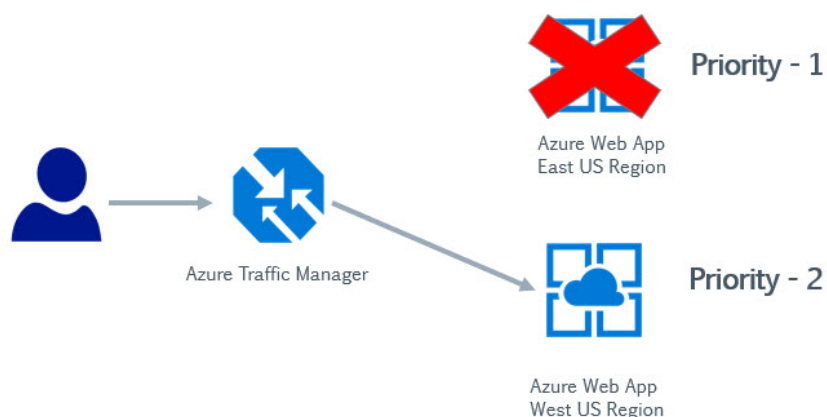
Here we are assuming that a company has similar web applications, both are running using the Azure Web App service. One web application is running in the East US Region and the other is running in the West US Region.



1. Here we create a Traffic Manager profile and create two endpoints. Each endpoint points to each Azure Web app respectively. We assign a priority of 1 to the service endpoint attached to the Azure Web App running in the East US region and a priority of 2 to the other service endpoint.

1. Here users would make requests to the Traffic Manager service.

2. The requests could be initially be directed to an Azure Web App located in the East US region, since there is a priority of 1 to the service endpoint attached to this endpoint.



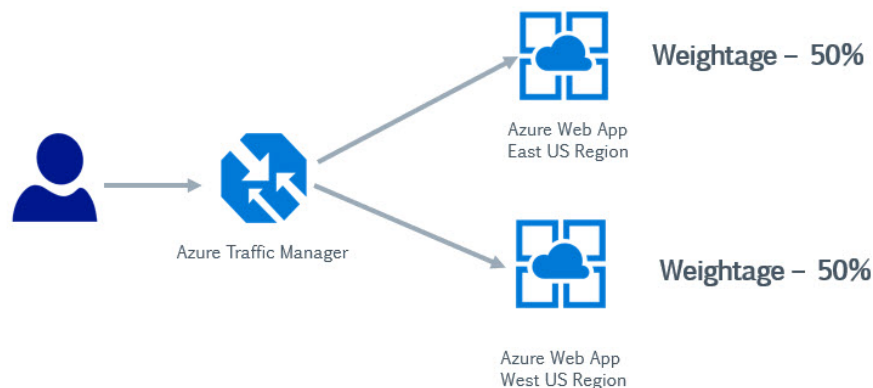
3. Now let's say there is an issue with the web application running in the East US region, Azure Traffic Manager would then understand that there is an issue with the web application running in this region.

It would then start redirecting user requests to the second endpoint which has the Priority of 2.

Hence over here you are adding a higher availability to your architecture by ensuring that user requests are always adhered to by redirecting requests if the primary service fails for any reason.

If you use the Weighted Routing method, you can actually load balance requests across multiple service endpoints

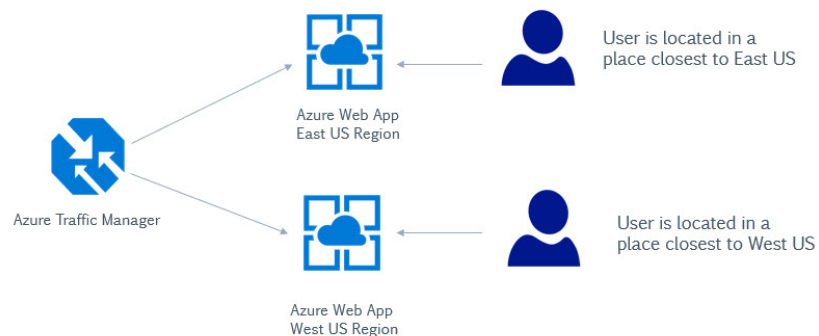
› Weighted



Over here, users' requests would be directed or load balanced across both web applications running in different regions.

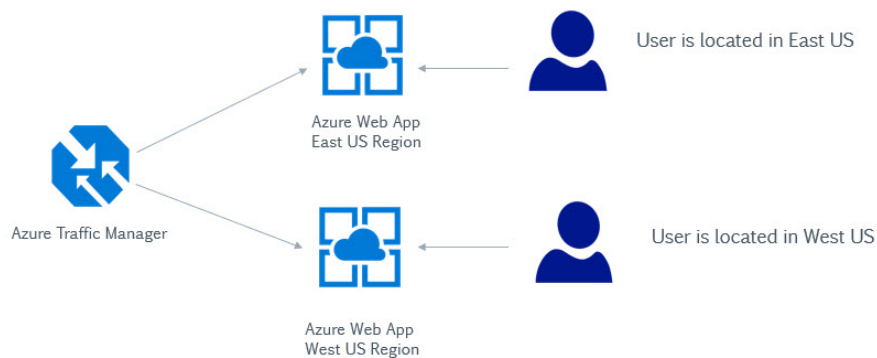
In the Performance routing method as shown below, users will be directed based on the least latency of an endpoint.

› Performance



And then we have the Geographic routing method wherein users would be directed to an endpoint based on their geographic location

› Geographic



SUBSCRIPTIONS AND MANAGEMENT GROUPS

SUBSCRIPTIONS AND MANAGEMENT GROUPS

- Company wants to ensure that they have staging assets and production assets are billed separately.
- Then in particular subscription they can have different RG and different Resources.
- And we want to have these 2 subscriptions for different departments
- than we can have a management group on top of the subscriptions.
- The hierarchy would be
 - Tenant root group-->group or departments-->subscription-->RG-->resources

LAB

- Search Mgmt. grp-->add krdo-->name-->subscription.
- further in particular mgmt. grp we can add policies, budgets, cost analysis, access control.
- These policies will be inherited by subscriptions as well.

ROLE BASED ACCESS CONTROL

ROLE BASED ACCESS CONTROL

- It is an authorisation system that is built on top of Azure Resource manager that allows to provide fine grained management of Azure resources.
- All user's login in azure portal using Azure AD, but we want to user access to particular VM or SA, we can do this with Role based Access control.
- We can assign role to user acc. to we want to give user access.
- We can assign role at resource level or at RG level or subscription level.
- Role is nothing but a JSON application, we can also create custom role, it includes action (Read, Write, etc.), in Sa we have data actions, in scope we define at which level we want to apply that role.
- If you have multiple roles assigned the combined with, be sum of the role assignments.

- If you give reader role and contributor role-final assignment is contributor role.

Check out this link

<https://docs.microsoft.com/en-us/azure/role-based-access-control/overview>

LAB

- Got to Azure AD-->existing user choose any.
- Go to a VM-->access control section-->Role assignments tab-->Add-->Select The Role (based on services and all) (reader)-->select user-->add.
- Login as that user to azure-->we will see that VM, and we will be able to see details and we will not be able to see the Public IP and all as we don't have permission.
- Go to azure portal with our main account-->open any RG-->access control--role assignment-->add-->reader-->add that user.
- login as that user-->we will see all resources which that RG has, and can see details of every resource, but cannot change anything.

CUSTOM ROLE BASED ACCESS CONTROL

- Go to RG-->access control-->role assignment-->add custom role
- **BASICS TAB**
 - Name
 - Description
 - Baseline permissions-> (clone role/start from scratch/start from JSON)
- **PERMISSIONS TAB**
 - Add permissions-->we will see all services-->search compute-->data actions-->login to VM-->availability set and all.
- **ASSIGNABLE SCOPE TAB**
 - Choose scope-->RG level
- **JSON TAB**
 - JSON definition
- **REVIEW AND CREATE TAB**
 - Next Create.
 - Now go to RG-->access control-->add role assignment-->choose your custom role.

NOTES ON ROLE BASED ACCESS CONTROL

Quick note on some of the roles

A quick note on the RBAC roles, just some important points from the exam perspective

All these notes use the reference - <https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles>

Contributor Role

- Remember that this role allows a user to manage all types of resources but does not allow the user to grant access to resources.
- To allow a user to have the ability to grant access to resources, the user must be granted either the User Access Administrator Role or the Owner Role

User Access Administrator Role

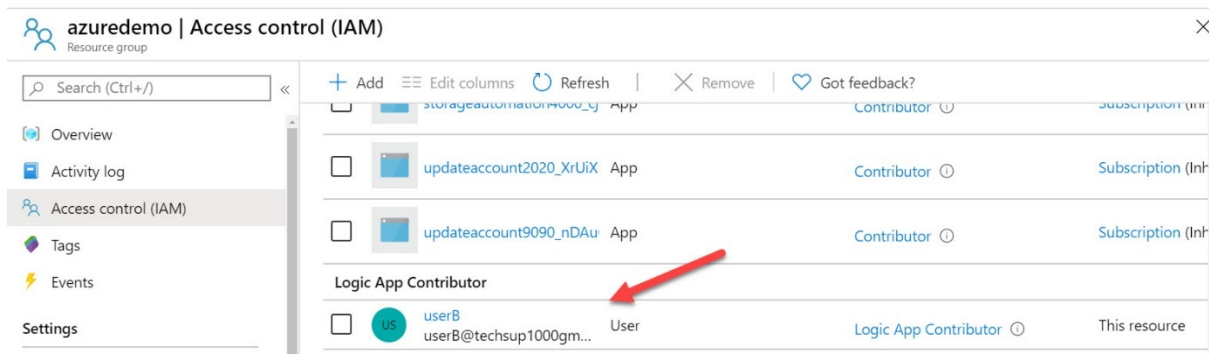
- Here the user can manage the access to resources. Here the user would be able to read all resources. But the user can't modify resources

Virtual Machine Contributor Role

- This allows to manage the properties of the Virtual Machine. This though will not provide access to the underlying virtual network, or the storage accounts the virtual machine is connected to.

Logic App Contributor Role

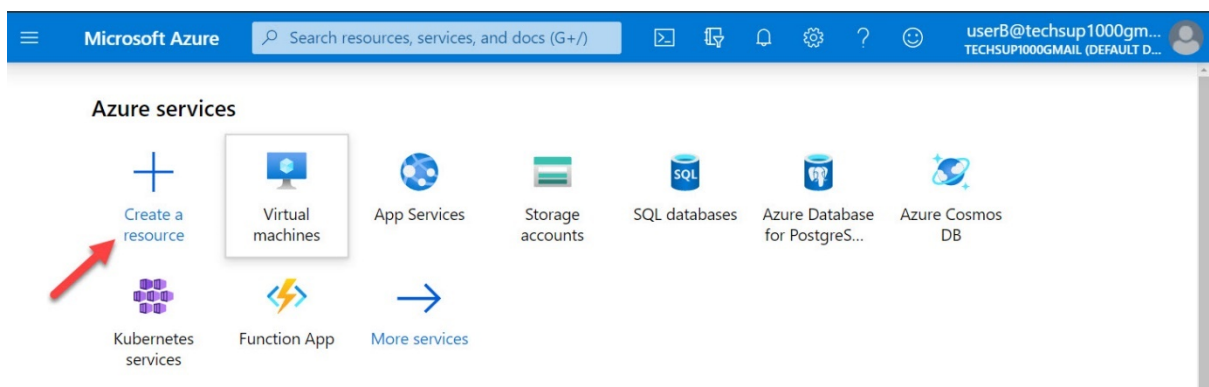
- This Role allows one to create and Manage Logic Apps. Azure Logic Apps is a workflow-based service. You can also actually try this out.
- For a resource group, give a user the Logic App Contributor Role



Next go ahead and log in as the user , in this case

userB@techsup1000gmail.onmicrosoft.com

Go ahead and click on Create a resource



In the marketplace search for Logic App

[Home](#) > [New](#)

New

Azure Marketplace [See all](#)

Popular

Next go ahead and Create the Logic App

Logic App

Microsoft



Logic App

Microsoft

[Save for later](#)

Create

Provide the details of the Logic App

Logic App

Project details

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

Subscription *

Resource group * [Create new](#)

Instance details

Logic App name *

Select the location ☒ Region ☐ Integration Service Environment

Location *

Log Analytics ⓘ

[Review + create](#)[Download a template for automation ⓘ](#)

Go ahead to Review + create and create the Logic App

Logic App Reader

- This only allows to read, enable, and disable logic apps, but not edit or update them

Security Read

- This role is meant for Azure Security Centre. Sometimes this gets confused with giving access to users, or allowing high privileged access for users. But this is not the case.
- This role allows to view recommendations and alerts, view security policies, view security states in Azure Security Centre.
- But here you cannot make changes in Azure Security Centre.

Security Admin

- This role allows one to view security policies, view security states, edit security policies, view alerts and recommendations, dismiss alerts and recommendations in Azure Security Centre

AZURE POLICY SERVICE

AZURE POLICY SERVICE

- It is used for governance aspect for your azure resources
- Company wants to restrict the size of the VM for the users bcoz of security aspect or
- company needs to have antimalware extension installed on the VM's.
- All resource group should have tags in place.

- We can create our own custom policies as well and can also use existing policies.

LAB

- Go to policy service-->we can see many definition
- Search for compute-->Microsoft malware policy select kro
- We will see JSON policy definition

ASSIGNING POLICY

- **BASIC TAB**
 - Click assign-->define scope(subscription/RG) -->select
 - We have exclusion also we can choose any RG to exclude, don't exclude.
- **PARAMETERS TAB**
 - Effect(auditnotexists/disabled) -->choose audit
- **REMEDIATION TAB**
 - Leave
- **NON-COMPLIANCE MESSAGES**
 - Leave
- **REVIEW AND CREATE TAB**
 - Create krdo
- Refresh the policy, we will see the compliance devices as we have no malware installed.
- Go to any VM-->Extensions section-->no antimalware (we can see) is installed.

AZURE POLICY REMEDIATION

- Go to Policy resource--. definitions sections-->Choose deploy default Microsoft malware (in this we are saying please deploy)

ASSIGNING POLICY

- **BASIC TAB**
 - Click assign-->define scope(subscription)-->choose a RG-->select
 - We have exclusion also we can choose any RG to exclude, don't exclude.
- **PARAMETERS TAB**
 - Effect (AuditnotExists/disabled) -->choose Audit
- **REMEDIATION TAB**
 - Create remediation task-->it will use the managed identity(allow)→permission is contributor-->add
- **NON-COMPLIANCE MESSAGES**
 - Leave
- **REVIEW AND CREATE TAB**
 - Create krdo
- Now in VM-->extensions -->we will see Microsoft malware
- We can delete policy from policy resource.

AZURE BLUEPRINTS

AZURE BLUEPRINTS

- It gives the ability to orchestrate the deployment of artifacts to azure.
- using Azure blueprints we can orchestrate like ARM templates, Azure Policies, RG, Role Based Access Control.
- Suppose a company they may have some standard or default deployment for certain resources, policies RG's or even users who need certain type of rules, they can go ahead and use this.
- When subscription there are some resources or policies that need to be assign to particular users and all we can create this.
- It is different from ARM templates as these are infrastructure as a code, we can use JSON to create resources, but Azure Blueprints is architecture level scenario where we need certain resources and polices and role-based access control as part of Subscription or set of subscriptions.

DIFFERENT STAGES FOR AZURE BLUEPRINTS

- First define the blueprint itself
- Associate with subscription or mgmt. grp. it can't be defined at RG level.
- Publish Blueprint, we can have many versions of the blueprints
- Assign the blueprint.

POINTS

- Helps to define a repeatable set of processes that can adhere to an org. Standard and patterns.
- You can declaratively define artifacts such as:
 - Role Assignments
 - Policy Assignments
 - ARM templates
 - Resource Groups.
- STAGES OF AZURE BLUEPRINTS
 - **Definition**-here you define the blueprint itself. The blueprint needs to be saved to either a mgmt. group or subscription.
 - When you save blueprint to a mgmt. grp, the blueprint can be assigned to any subscription which is part of management grp.
 - to save blueprint definition, you need to have contributor access to either the mgmt. grp or subscription.
 - **Publishing**-Once the Blueprint is defining, you can publish it, Here you can Assign a version number for blueprint.
 - **Assignment**-Here the blueprint is then assigned to a subscription

- BLUEPRINT RESOURCE LOCKS
 - Here you can protect resources deployed via a blueprint resource locks.
 - Here even if there is a user with the owner role, still user will not be able to remove the lock.
 - You can remove the lock by unassigning the blueprint.

LAB BLUEPRINT-DEFINITION

- Search for blueprint-->create
- Blueprint definitions-->create
- Blank Blueprint
- Definition location-->choose mgmt. grp or subscription (if we don't have mgmt. grp choose the tenant grp), we need to have contributor access to tenant grp otherwise we will get an error.
- Go to tenant grp-->access control-->add role assignment for the root account-->owner or contributor role assign krlo.
- Further in blueprint definition-->next
- Artifact-->add-->type (RG/policy/role/ARM templates) -->choose RG-->Display name for artifact-->location-->tag name(leave)-->add
- Add another artifact-->type (Policy assignment) -->choose policy (azure backup) -->add
- Add another-->ARM template-->Name-->specify JSON template (creating SA's)
- Add another-->Role assignment-->Role (Owner)-->add user-->add
- Save draft
- In context menu of our blueprint definition, we created and choose publish blueprint.
- In context menu -->assign Blueprint

ASSIGNING BLUEPRINT

- Assignment Name
- Location
- Blueprint definition version
- Lock assignment-leave
- Managed Identity (system assigned)
- We can specify the parameter value for the artifacts we created like RG and all
- Click assign
- Go to all resources tab in our subscription we will see 2 SA's
- We will also see RG
- In RG we will see the role assignment (owner role).
- Policies-->we will see backup policy in place.
- We can go to blueprints -->Assigned Blueprints section-->we can unassign blueprint

- Now resources will be remained that we created as part of blueprint, we manually need to delete them.

AZURE WEB APP SERVICE

AZURE WEB APP SERVICE

INTRODUCTION

- We have a Web app we can create a VM in Azure and then we can host the web app using this VM(IaaS) or we can develop that web app using azure which is a platform as a service which is managed but azure (PaaS), we don't need to manage anything.
- Advantages of Azure Web App Service:
- You don't have to maintain the underlying compute infra.
- we can host app of (.Net, .NetCore,Java, Ruby,Node.JS,Python)
- It has feature such as Autoscaling and security.
- It has DevOps capabilities which include continuous deployment.

AZURE APP SERVICE PLAN

- When we create App, we need to create a app service plan, it is on which our app is built on, acc. To our usage we our billed.
- We can have 1 app service plan and can host multiple apps on that. but it depends like if we have windows VM service plan, but we want to host Linux based app than we can't host, we need another service plan.
- Check out this link
<https://azure.microsoft.com/en-us/pricing/details/app-service/windows/>

LAB

- **BASICS TAB**
 - Create a web app-->name-->subscription-->RG-->Publish9Code/docker container) -->Runtime stack (.Net Core V4.8)-->Windows(OS of the VM's that will host our app)-->Region-->App Service plan(new create Karlo)we can change size acc.(we can upgrade and downgrade our plan after creating a app also)
- **MONITORING TAB**
 - Leave
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
 - Create krdo

LAB-PUBLISHING APP USING VISUAL STUDIO

- Open visual studio-->ASP.Net-->name of project-->location-->Create

- Choose web app template-->Create
- Right click on project and publish it using Azure-->sign into azure-->select our web app-->Publish.
- It will launch browser window and we will see our web app running on azure.

EXPLORING AZURE WEB APP

- We can restart or stop web app at any time in azure.
- We can change the stack settings anytime
- Due to inactivity app goes to sleep we can always on setting to remain on all time
- We can define authentication and authorisation at any time.
- We can also take backup for our apps
- We can also define custom domains
- We can also scale up or scale out our app service plan
- We can do VNet integration, we can turn on hybrid connections, we can also use Azure CDN.

AZURE WEB APP--GITHUB

- Install extension of GitHub for visual studio
- In visual studio create a ASP>Net app and give name and all web app template-->enter something in <div> tag
- Now in browser open GitHub sign in also
- Now in visual studio-->buttons-->team explorer-->GitHub-->login into GitHub-->connected hojayega
- Go to solution explore-->add solution to solution control-->right click commit-->commit message-->sync-->publish to GitHub-->mention name of repository
- Open GitHub in browser we will see the repository.
- In portal -->web app-->Deployment centre section-->GitHub Actions-->connect krdo GitHub se-->continue
- Choose app service build service (now what is going to happen is our web app will be pick up from GitHub and tha build as an app service--continue
- Choose repository-->continue-->Finish
- Now our code is sync
- Now if we will go to our website, we will the text, so our app is built on azure.
- If we will make changes in app using visual studio, in portal we will see it automatically take the changes after committing changes on GitHub using visual studio.

AZURE WEBJOBS

This is a feature of Azure Web apps

- This allows you to run background tasks

- We can also run PowerShell scripts.
- WebJob Types
 - **Continuous**
 - Starts immediately, when the webjob is created. To keep the job from ending the program or script typically does its work inside an endless loop. If the job does end, you can restart it.
 - Runs on all instances that the web app runs on. You can optionally restrict the WebJob to a single instance.
 - Supports remote debugging.
 - **Triggered**
 - Starts only when triggered manually or on a schedule.
 - Runs on a single instance that Azure selects for load balancing.
 - Doesn't support debugging.

LAB

- Now we have a batch file that write or prints "Hello World".
- Now in Web App-->webJobs Section-->add-->name-->select file (should be zip)-->triggered(manual)-->ok
- Job will be created
- choose webjob and run the webjob
- Click on Logs tab-->it will open new page in browser-->Success-->go to job-->we will see 'Hello world'

AZURE WEB APP-APP SERVICE LOGS

- You can enable diagnostics for you web based apps in Azure web app service.
- **Application Logging**-->These are log messages that are generated by application code.
- **Web server logging**-- This is raw HTTP request data.
- **Detailed Error messages**--These are copies of the .html error pages that would have been sent to the client browser.
- **Failed request tracing**--This provides detailed tracing information on failed requests.
- **Deployment Logging**--These are logs generated when content is published to an application
- Check out this link
<https://winscp.net/eng/download.php>

LAB

- Watch video.

AZURE WEB APP-AUTOSCALING

- It is part of App service plan
- if we opt for standard or higher, we will get this feature
- Now lest say number of users are increasing on app and load increasing on the VM also or infra also
- we can alleviate this issue we can have 1 more instance of VM.
- In basic plan we can scale up to 3 instances, but it is manual process here.
- If we choose standard or higher, in this it happens automatically based on conditions we specify like CPU percentage, acc. to that we can scale in and scale out.

LAB-AUTOSCALING WEB APP

- We have web app based on Basic app service plan-->if we will go to scaling section, we need to do it manually.
- Go to scale up (app service plan) section-->Production tab-->Standard choose Karlo-->apply.
- In webapp-->scale out section-->custom autoscaling-->we can have multiple conditions-->we can based on metric also-->we can Add rule--> acc. to which it will apply
- We will add CPU percentage-->duration
- **Cooldown period**-->If the rule is triggered and it add another VM, not it will take time to distribute load across these VM's, in this cool period no scale out condition will work as it is waiting to distribute load.
- Mention number of minimum and max instances also-->.add
- We will see the instance count increases bcoz we set condition such that rule could be triggered.

AZURE WEB APP-DEPLOYMENT SLOTS

- We can make use of staging environment for web apps
- Now we want to deploy the new version of our app, if we publish it will replace but we want to check first and then publish.
- We can create deployment slot (staging slot and production slot)
- It can be created for standard, premium, and isolated app service plan.
- Application in deployment slots have their own names
- Advantages of deployment slots
- You have the chance to validate all app changes in the staging deployment slot.
- You can then swap the staging slot with the production slot.
- This helps eliminate the downtime for your app when new changes are deployed.
- You can also easily roll back the changes

LAB-DEPLOYMENT SLOTS

- Go to the web app-->deployment slot section--Add Slot
- Name of slot-->not clone anything.
- In visual studio-->index file-->make changes in text in app
- In portal click the slot(staging)
- In visual studio -->publish app--<new profile-->azure web app-->deployment slot(staging)-->ok
- URL of the app type on browser-->we will get msg.
- And now our testing is complete, and we can swap slots in portal and not if we will refresh page we will see previous version of the app.

AZURE WEB APP-APPLICATION INSIGHTS

- Application Performance management service for web developers.
- You can use this tool to monitor your apps.
- It can help developers detect anomalies in the app.
- It can help diagnose issues.
- It can also help understand how users use the app.
- It can also help you improve performance and usability of your app.

LAB-APPLICATION INSIGHTS

- In portal in web app-->Application Insights section-->Turn on krdo
- It will create a new application insight resource--name-->leave everything as it is.
- Open the URL of our web app, exercise the various part of app again and again so, that we can record our insights.
- In web app-->Live metric section-->we will see the graph and live data from the app.
- In performance section we can see performance of our web app.

AZURE WEB APP-VENT INTEGRATION

- Now web app is public service, now if we want app to interact the app with the DB server which is hosted on VM and is in VNet, it is isolated network, and we can only access using the private IP.
- Now we want to interact web app to interact with DB server.
- One way is we also have a web server hosted on VM in same VNet and it is exposed on port 80 and through this we will interact to DB server and then we can connect that web server to the web app, but it is a security risk. So, we can connect web app to the VNet.

LAB

- Create a new web app

- Create a VNet-->default subnet only.
- Go to Web app-->Networking section-->VNet integration -->Configure
- Add VNet-->choose our VNet--chose default subnet-->done
- Now create a VM-->image (SQL server 2019) it will have VM with SQL server installed on it.
- Create VM in same VNet and add another subnet for it as it will not take otherwise.
- SQL connectivity settings ayengi-->Public—Port (1433) -->create krdo VM.
- Copy Public IP of VM
- Open the SQL mgmt. studio on your lapy
- Enter the IP of the VM-->connect-->login credentials of VM.
- Right click on the Db in mgmt. studio-->create a table -->insert some rows-->select query to check the table.
- Open visual studio-->code hoga (no need of that) -->connect kr rhe app ko DB se
- Go to the VM in portal disassociate IP as we do not need now.
- Now copy private IP of VM and paste it on the visual studio code
- Publish app-->choose the app-->Ok-->publish krdo
- It will open the page in browser and will show the table we created.

AZURE FUNCTIONS

AZURE FUNCTIONS

- This service allows you to run small piece of code as functions
- Here you just develop and upload the code to an azure function.
- You only get billed for the amount of time the code is run.
- you can use variety of programming languages in Azure Functions
- C#, Java, JavaScript, PowerShell and python.
- You can use libraries by using NuGet and NPM packages.
- **Invoking you function**-There Are different ways you can invoke your function
 - HTTP Trigger
 - Timer trigger
 - Blob Storage
 - Queue storage
 - Event hubs
- **Pricing Plans Available:**
 - **Consumption Plan**--Here you only pay for the time the code tuns.
 - **App service Plan**--It you already have an App service plan that runs a web app, you can resume the same plan to run Azure Functions. This will save on cost if you already have an App service plan in place.

- **Premium Plan**--Here you get a number of pre-warmed instances that are always online and ready to run your functions. The plan also automatically adds more compute when required.

LAB- AZURE FUNCTIONS

- Search function in portal.
- Create function
- **BASICS TAB**
 - Subscription-->RG-->function Name-->Publish9code/docker container)
 - Runtime stack (choose language) .Net
 - Version (of the language)3.1
 - Region
- **HOSTING TAB**
 - We need to have a SA acc.
 - Underlying OS (windows or Linux)
 - Plan type (Consumption/ASP/Premium) choose consumption
- **MONITORING TAB**
 - Enable application insights
- **TAGS TAB**
- **REVIEW AND CREATE TBA**
 - Create

ADDING A FUNCTION

- Open the function resource-->Functions section
- Add function-->choose development Env (develop in portal) -->select the template (HTTP Trigger) -->add
- Go to the function-->Code + Test section-->there would be default code-->test and run
- Input the HTTP method and query parameter
- and run
- It will give us the result
- We can also see the logs

AZURE FUNCTION-APP SERVICE PLAN

- We have 2 app plan based on basic tier in different location
- Create a new function and choose app service plan.
- In function we will now see the backup option or section which will not be there for consumption plan.

USE OF AZURE FUNCTIONS

- It is serverless service, no need to manage infra.

➤ **EXAMPLE**

We have an app, in which we have different modules like for adding product process orders or display orders and this can be increased day by day. So, instead we can use azure function for different modules as rather than growing our code only and adding this there only.

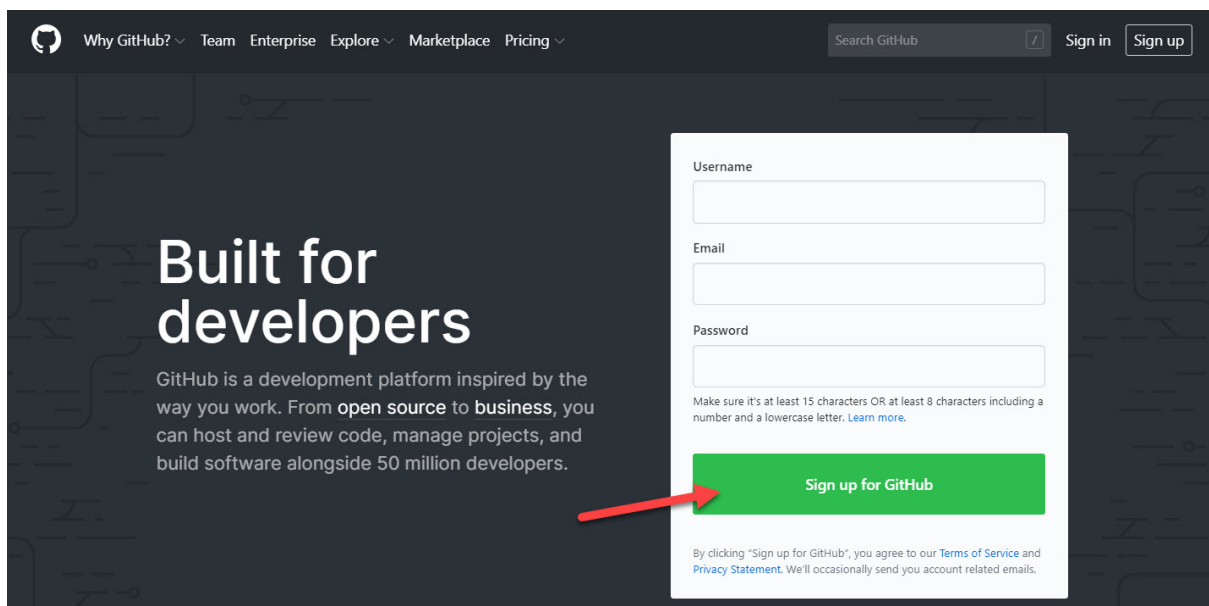
- We can have different underlying programming languages as we can have in >net and we can have Azure functions in different languages like Java.
- There are triggers and bindings from other azure service like message added to queue and we can then invoke function and then link to app.

QUICK NOTE FOR THE NEXT VIDEO

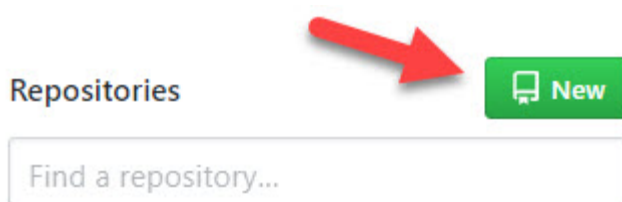
DEPLOYING AN APPLICATION FROM GITHUB

- In the next video we are going to see how to deploy an application from GitHub onto an Azure Web App
- If you don't have an account with GitHub, it is very easy to go ahead and sign-up for a GitHub account.

1) You can go to the following URL - <https://github.com/>
And then sign-up for a GitHub account



2) Once you sign-up and sign into GitHub, on the left hand side of the screen , you should be able to see an option for creating a new repository




3) You can then give a name for the repository and hit on Create repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

Repository name *


 alashro ▾

/ newrepository ✓


1

Great repository names are short and memorable. Need inspiration? How about [urban-parakeet?](#)

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.


Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾

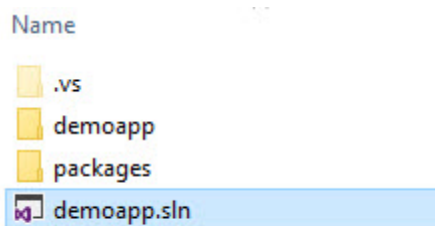


Create repository

2

4) Extract the code which is present in the zip file attached to this chapter

You should get the following files after extraction



5) Next when you create your repository, you will see steps on how to add your application code to the repository

alashro / newrepository

Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/alashro/newrepository.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

1

...or create a new repository on the command line

```
echo "# newrepository" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/alashro/newrepository.git
git push -u origin master
```

2

...or push an existing repository from the command line

```
git remote add origin https://github.com/alashro/newrepository.git
git push -u origin master
```

I used the second method and i have the following code in my repository

Branch: master New pull request Create new file Upload files Find file Clone or download

techsup1000 secondary change Latest commit d9cdda4 27 days ago

demoapp	secondary change	27 days ago
.gitattributes	Add .gitignore and .gitattributes.	27 days ago
.gitignore	Add .gitignore and .gitattributes.	27 days ago
demoapp.sln	Initial commit	27 days ago

AZURE LOGIC APPS

AZURE LOGIC APPS

- Cloud Service that helps you schedule, automate, and orchestrate tasks business processes and workflows.

HOW IT WORKS

- You first design a workflow in Azure Logic Apps.
- Each workflow starts with a trigger.
- The trigger is fired via a specific event.
- When the trigger is fired, the logic app engine created a logic app instance that runs the workflow.

CONNECTORS FOR AZURE LOGIC APPS

- These connectors provide easy access to event, data and actions that are sent from external apps, services, systems, or platforms.
- You have built-in connectors that can connect to Azure services such as Azure functions, Azure API Apps etc.

- You have managed connectors that can connect to platforms such as Office 365, Microsoft Dynamics.

LAB

SCENARIO

- We will create an instance of Azure logic App
- we will build a workflow
- Workflow is if any event occurs in RG, then details of that event must be added. as a blob on the blob storage
- We want to ensure that when the event is triggered it is added in blob.

LAB- AZURE LOGIC APPS

- Creating a logic App
- **BASICS TAB**
 - Subscription
 - RG
 - APP NAME
 - Location
 - Log analytics turn off
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
 - Create
- Go to resource we will be presented with logic app designer.
- we have ability to use inbuilt template also
- but we will create a new
- Now we want when event occurs in RG, we want to trigger the logic app. Now in azure events from various services goes to central service known as event grid.

EVENT GRID-- It is a service that has capability of listening on to events that are emitted by other as your services.

- Search for event grid in designer-->we will see (when a resource event occurs) trigger choose that-->choose default directory-->choose azure account-->sign in
- **BUILDING WORKFLOW**
 - Now for trigger enter details
 - choose subscription-->R Type-->resource name (RG)-->Event type (delete success write success choose. -->add
 - Choose an action now when the trigger is done-->create a new container in different tab-->choose blob container-->and action select blob-->create blob/copy etc.-->connection name-->choose our SA acc.
 - after action complete choose our container

- Now it will ask for blob name and what add content we need to add (dynamic content choose kr skte) (Blob ID, event type, subject)
- done with workflow. and save it
- now perform action in that RG like in SA go to sections and all.
- after some time, we will have a file in our container, and we can see the content it will contain details of the trigger we mentioned.

AZURE SERVICE BUS

AZURE SERVICE BUS

- It is messaging system
- There are 2 services which we can use in service bus: -
 - **QUEUE**
 - We have seen in SA, in service bus, it provides first in, first out message delivery (FIFO).
 - The receiver will receive the messages in order in which that are received.
 - **Example-->** So at one end of our app, we will have a publisher, publisher will go out and send messages onto the queue and then you will have a consumer. Consumer will go ahead and get the messages from the queue.
 - **TOPICS**
 - It is also a messaging service.
 - But on RHS we can have multiple consumers like we have subscribers,
 - Subscribers can go ahead and subscribe to the topic so that when a message is sent from the publisher onto the topic, that message will be available to all the apps.
 - For example: lest say a service provider is sending a notification or some sort of important info, so now if we want to go ahead and subscribe to that service and get the message every time, this works like this.

LAB-SERVICE BUS

- Create a new resource service bus
- **BASICS TAB**
 - Subscription
 - RG
 - Namespace Name
 - Location
 - Pricing tier (standard-topics not available in basic)
- **TAGS TAB**
- **REVIEW AND CREATE**
 - Create

- Now go to the resource--> QUEUE Section

CREATING QUEUE

- Name
- Max queue size
- Message time to live
- Lock duration
- Accept default settings
- Go to the queue-->service bus explorer-->we can write message and then send the message
- We can receive message
- We can also Peek the message (we will get the message properties).

CREATING TOPIC

- Go to service bus resource-->Topic section
- Name of topic
- Max topic size
- Message time to live
- Create
- Go to that topic
- Add subscription-->name-->max delivery count (3)-->auto delete-->never delete checkbox-->forward messages checkbox
- Message sessions-enable
- Time to live
- Create
- Create another subscription
- Go to service bus explorer section--. send krke dekho-->we will see that each subscription will get the message as both our subscribers.
- We can receive and peek message we need to choose the subscription when doing actions also.

CONTAINERS

PRIMER ON CONTAINERS

EVOLUTION OF CONTAINERS

- Now if company have 3 apps and they deploy these app on 1 physical server. Now, 2 apps have same set of dependencies like libraries and all. and 3rd app have different dependency. Now all apps running on physical server, now if we make changes acc. to 3rd app, and as part of change, we are making changes in the one its dependencies.
- Now it will cause issues for the other apps. It was always an issue, Then we can to evolution of the VM's

- In which on physical server on which we can install tool like VMWare, and we can host VM's on it and we will deploy apps separately on each VM's and now we can install dependency on each VM.
- Now we have next set of evolution, in this we will have physical server on it a hypervisor than we can host VM's on hypervisor. Now on VM's you can deploy your application within or as a container. So, app will be embedded inside a container

BENEFITS

- Container is package of OS or system libraries or app itself.
- Now we can run many containers on the VM's,
- Instead of having separate VM's for your App, we can have application as a packaged as containers at all of these containers could run on VM.
- Instead of running app on single VM, you can just use containers on single VM.
- The size of container, if the OS is of size 1GB on the VM, the container will contain OS of size 150 MB, so its lightweight OS.
- we can also move these containers on the VM's as well. To manage or move these containers we use the docker engine.

WHAT IS DOCKER?

- It is basically a toolset that can be used to create, deploy, and run your app as containers

WHAT IS CONTAINER?

- Standard unit of software.
- Contains your software code and whatever is required to run your software code.
- The container is lightweight and standalone.
- you could easily port this and run it on any computing environment.

WHY DOCKER?

- It helps to containerize your app. makes it easier to maintain and deploy.
- Since it is smaller in size, it becomes faster to deploy.
- Also reduces the size and cost for your infrastructure.
- Build common docker images.
- Since you can deploy the same app on different computing environments, it reduces the blame game when you have the scenario, "But it works on my computer"

LAB-LOOK AT DOCKER

- In portal-->add a new VM
- Fill details and create
- Connect to Linux VM-->run commands to install docker (available at docker documentation).
- We will pull the nginx image from the docker.
- we will have nginx image
- If we want the container out of that image run that command.
- In portal on Linux VM, add inbound port rule for port 80.
- Take public IP and paste it on browser we will see the nginx server image which is running on container.

WHAT WE HAVE DONE IN LAB:

- We spin up the VM.
- On that VM we install a docker engine.
- We then ran a nginx container on this VM using Docker and then using Private or public IP we will be able to see home page of nginx server.
- docker run command means we want to run container.

Command- **docker run -name mynginx1 -p 80:80 -d nginx**

HOW CONTAINER PULLS IMAGE FROM DOCKER?

- There is a place called docker Hub, so this place on the internet has a public repository where all images are available to run on docker.
- So, when we mentioned the image name as nginx, it goes on to docker Hub, it finds for the nginx image, it goes ahead and pulls image from docker Hub and places it on this VM. Docker Hub have a lot of images at public repository.
- Once image is in place we are using docker run command, to now create and run the container out of this image.
- It will do that then it will name to container we have provide in command
- Now nginx is running on its own container on this VM, and within this VM the nginx runs on port 80 and we need to map port the nginx server to that port.

LAB-DEPLOYING .NET CORE APP ON THE VM

SCENARIO

- We have an existing Linux VM, where we install the ASP.NET Core SDK v3
- Publish our app using visual code.
- copy publish folder to server.
- run the app using dotnet.

- When we run our app, it will run on the kestrel web server(<http://localhost:5000>), as it does not have all features of web server, we can use nginx server or apache server.
- We will see how we can reach our dotnet web app that is running on the Kestrel Web Server via a Nginx Web server.
- So, on same VM we will install nginx server, and this is going to behave as a reverse proxy.
- So, when user makes request to the homepage for nginx they will be directed on to dot net core app.
- For this we need to
- We must add code to forward header information.
- Modify the nginx configuration file.

LAB

- On local system we have a dotnet core app-->go to startup file and add the right middleware to this.
- Open CMD-->go to your project folder-->publish app using command dotnet publish.
- On Linux VM we have a nginx server and install nginx
- Allow port 80 onto this network, with public IP of this VM we are able to open the home page for nginx app, but we want dotnet.
- Install the dotnet SDK on VM using commands.
- Connect to new server using the win SCP and public IP of VM daalo.
- In dotnet core folder we can copy publish folder
- And run command to reach that file and publish krdo dotnet project which we created.
- Watch video

LAB-CONTAINERIZING A .NET APP

- Watch video

LAB-AZURE CONTAINER REGISTRY

- Now we have an image of Linux VM which we installed from docker hub, and we want to use this image on other VM's as well, we can store this image on either docker hub or in Azure container registry.
- On Linux we can run commands to push that image to container registry.
- Create a container registry from portal simple details.
- Go to Linux VM, install the cli using commands
- After that login to azure using command

- Now login to container registry using cmd.
- Tag the custom image
- Push the image to container registry
- In portal-->CR-->repositories section-->we will see the image.

AZURE CONTAINER INSTANCES

- This is a service you can use to deploy containers in isolation
- Here you don't need to manage the underlying infra.
- You can persist data using file shares.
- The container also gets an IP address and fully qualified domain name.

LAB

- Add new resource-container instance
- Fill details
- Subscription-->RG-->location-->name-->image source (ACR)->registry choose kro apni-->image(dotnetapp)-->image tag-->OS type(Linux)-->size
- Networking expose port 5000 at our container listens here
- Create krdo.
- Public IP copy kro browser pe we will see our dotnet app.

AZURE WEB APP-DOCKER CONTAINER

- We can create web app based on containers.
- Azure web app can take image from ACR also.
- Create a new azure web app, fill details.
- In publish option choose(docker container)
- Create krdo,bki details bhrke and choose to take image from ACR.
- Go to web app-->IP-->we will have our app.

KUBERNETES

WHAT IS KUBERNETES AND AZURE KUBERNETES?

- In company we will have 100 of containers, and to manage the containers, scale the containers, keep track of containers on these VM's, if container stop working take action, all of this is a tedious task and manual process, and it gets difficult.
- In such case we need a orchestration software to manage containers which is Kubernetes.

- We will have VM on which we will install Kubernetes software and mark it as master node of the Kubernetes cluster, on this cluster we will have nodes, on which we will run docker containers.
- These nodes would be managed by your master node. So, each of this node we will install Kubernetes software, and connect them to the master in our cluster.
- Then we will deploy your containers onto the nodes using kubectl tool.
- **KUBECTL**--it is command line interface tool that can be used to work with this cluster

POINTS

- It is an open-source platform that is used to managing containerized workloads
- Kubernetes is able to provide a DNS name to your container.
- if there is a high load on your containers, Kubernetes can load balance and distribute network traffic.
- Kubernetes can also restart containers that fail.
- It can be used to replace or kill containers.
- It can also help to store and manage sensitive information such as passwords OAuth tokens and SSH keys.

AZURE KUBERNETES

- Fully managed Kubernetes service on Azure.
- Makes it easy to deploy and manage containerized apps
- It helps to remove the burden of managing the underlying infrastructure for the Kubernetes deployment.

LAB-CREATING KUBERNETES CLUSTER

- Add new resource--Kubernetes service
- Subscription-->RG-->region-->name -->version-->node size-->node count
- We can add node pools
- Networking
- we are creating using Kubernetes cluster using azure CLI.
- we will have our cluster.
- Watch video.

UNDERSTANDING OF APP DEPLOYMENT TO A KUBERNETES CLUSTER

- When we are running a container on a node in a cluster, it runs on something known as pod.

- **POD**--It is smallest and basic deployment on the Kubernetes cluster. it can have its own internal IP, normally 1 pod is made to run 1 container but we can run multiple containers as part of the pod.
- We can have multiple pods running as part of deployment.
- **Replicas**-Number of pod to run (to load balance traffic to have high availability for containers to have multiple pods running we can have replicas).
- **Rolling deployments**-->change the container image
- **Healing**-->if a pod fails, the deployment will create a new pod.

SERVICE PRINCIPAL

- Now we want azure Kubernetes service to pull out the image from the ACR.
- Now ACR is separate service and Kubernetes cluster is a separate service, so when we want one service to talk with another service in azure, the service needs to be authorized to basically use either service.
- we will use service principal. it is an identity which will have ID and we could assign the role to the service principal and service principal could be assigned to the service which would be Kubernetes, which could have rights to access ACR.

LAB-DEPLOYING NGINX SERVER ON TO THE KUBERNETES CLUSTER

- Watch video.

DATA FUNDAMENTALS

AZURE SQL DATABASE SERVICE

- First is IaaS, in which we can have a VM on the platform and we can install SQL DB services on it, and then setting up our databases.
- Second, PaaS, In here, we don't need to worry about the underlying platform, backup and all, we can use Azure SQL DB

DIFFERENT OPTIONS

- **SINGLE DB**-->When we want to create 1 DB on the azure.
- **MANAGED DB**-->In this if company has their DB, they can simply shift and lift their DB to the Azure SQL DB service.
- **ELASTIC POOL**-->We can combine all the SQL pool into an elastic pool, where now all the databases will share the underlying resources.

UNDERSTANDING

IAAS ADVANTAGES

- You have full control over the SQL Engine.

- It's easy to migrate from an on-premises instance, because you can install your own DB version.
- You have private IP addressing via a Virtual Network.

BUT

- You must manage the underlying VM.
- You must manage the SQL instance.
- You have implemented your own high availability solution.
- You need to manage the backups.

PAAS ADVANTAGES

- Here 99.99% availability is guaranteed.
- It has feature such as built-in backups, patching and recovery.
- Has built in advanced intelligence and security.

BUT

- Migration from an existing on-premises instance might be more difficult.
- Some SQL features may not be available.
- You don't have the concept of private IP addressing. You need to manage access via firewalls.

PURCHASING MODELS-PaaS

- **DTU (Database transaction Unit)**
 - Blended measure of CPU, Memory, and read-write rates.
 - There are different service tiers.
 - Each service tier had different features and different pricing.
- **vCore-Based**
 - Here you can independently scale the compute and storage.
 - You can choose different service tiers.
 - You also have replicas for high availability.
 - You can also make use of Azure Hybrid Benefit.
 - Here you can save on costs if you have existing SQL server licenses.

MANAGED INSTANCE OFFER

- New deployment model that allows for easy of existing on-premises SQL server databases
- Provides 100% compatibility with the latest SQL Server on-premises (Enterprise Edition) database Engine.
- Also provides a native virtual network implementation.
- Here there is no management of infrastructure.

LAB-AZURE SQL DATABASE SERVICE

- Add new resource-->SQL Database
- **BASICS TAB**
 - Subscription
 - RG
 - DB name
 - Server (create a new which our DB will be hosted) -->name-->credentials (username and pswd) -->location-->ok
 - Elastic pool (yes or no)
 - Compute +Storage(size)-->configure DB-->here we can see cost summary and give us pricing model. (DTUs, vCore based)
- **NETWORKING TAB**
 - Connectivity method (no access/Public endpoint/private endpoint) in public add you IP of our workstation.
- **ADVANCED SETTINGS**
 - Data source (use existing data-->adventure works DB choose)
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
 - Create krdo.
- Now go to the resource
- To connect we need to download SQL server mgmt. studio
- Take server name from SQL DB from portal
- In SSMS tool-->enter details to connect to DB server.
- We will our databases and tables
- To connect to another workstation in SQL DB service we can add IP through firewall settings.

AZURE SQL DB-USING QUERY EDITOR

- In SQL DB service on portal-->Query editor Section
- Login to you DB
- Now we will have query editor
- We can write the queries for our table and create table and all and can see results.
- For full functionality we can use SSML but here also we can run some queries as well.

SQL ELASTIC POOLS

SCENARIO

- Suppose we have spun up 2 SQL DB and it is not part of elastic pool, and it is based on basic DTU model-5 DTUs, for underlying SQL DB.
- At some point in time, bcoz of load on DB, the DTUs are not enough, then we can scale up the amount of the data used for your underlying OS.

- But when you scale out the data use, you might get a slight interruption onto DB service. It can be issue, this interruption is bcoz when we scale the amount of DTUs, there are new compute resources that get created
- So, then DB connections are being created, so all the current DB connections are actually dropped, and all new connection are then created.
- We can keep scaling up and down.
- We can make use of elastic pool, we can make DB's part of the elastic pool.
- So, instead you are specifying the amount of details on a per DB model which we can do also, as a part of DTU model.
- We can go out an specify DTUs as part of elastic pool level.
- During peak times more DTUs will be assigned by the elastic pool onto the underlying DBs and again there will be no interruption and compute instances will be there as there will be minimum 50 instances.
- It can be used when we don't understand how many DTUs will be needed.

POINTS

- This is a cost-effective solution that can be used to managing multiple databases that have varying and unpredictable usage demands
- The databases that are part of the elastic pool need to be on a single server.
- These databases that are part of the elastic pool need to be on a single server.
- These databases would share the resources.
- The resources are already allocated at a set price.
- Because the resources are already available for all the databases based on the load is done automatically.
- When you move database in and out of an elastic pool, there can be a brief outage in order of seconds. Here again at the end of operation the existing database connection are dropped.

LAB- ELASTIC POOLS

- Add a new SQL DB service, fill details, and make it part of elastic pool
- Create an elastic pool-->pool name
- Configure the elastic pool-->eDTUs will be minimum --50--Apply
- Create krdo.
- Go to SQL Db service -->from that go to DB server-->we will see the elastic pool
- Go to elastic pool-->configure-->databases Pool-->we can add or remove the databases from here.

LAB-WORKING WITH DATA IN A SQL DATABASE

- Here we have 3 tables orders, customer, and course.
- We have primary key and foreign key.
- Orders table have course ID and Customer ID

- Orders table have relation between both customer and course table.
- So, we first we need to create customer and course table.

LAB

- In portal-->SQL DB-->connect to your SQL DB using SSML tool on our workstation, create tables and insert data or rows also.

LAB - WORKING WITH DATA IN A SQL DATABASE - RESOURCES

You can use the following commands as a reference for the previous chapter

1. The following commands can be used to create the database tables

CREATE TABLE Customer (

CustomerID varchar(100) NOT NULL,

CustomerName varchar(1000),

PRIMARY KEY (CustomerID)

);

CREATE TABLE Course (

CourseID varchar(100) NOT NULL,

CourseName varchar(1000),

Price real,

PRIMARY KEY (CourseID)

);

CREATE TABLE Orders (

OrderID varchar(100) NOT NULL,

CourseID varchar(100),

CustomerID varchar(100),

Discountpercent int,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (CourseID) REFERENCES Course(CourseID)

);

2. The following commands can be used to insert data into the table

```

INSERT INTO Customer(CustomerID,CustomerName) values ('C1','UserA');
INSERT INTO Customer(CustomerID,CustomerName) values ('C2','UserB');
INSERT INTO Customer(CustomerID,CustomerName) values ('C3','UserC');
INSERT INTO COURSE(CourseID,CourseName,Price) values ('D1','AZ-900',99.99);
INSERT INTO COURSE(CourseID,CourseName,Price) values ('D2','DP-900',100.99);
INSERT INTO COURSE(CourseID,CourseName,Price) values ('D3','AZ-104',89.99);
INSERT INTO Orders(OrderID,CourseID,CustomerID,Discountpercent) values
('O1','D2','C1',90);
INSERT INTO Orders(OrderID,CourseID,CustomerID,Discountpercent) values
('O2','D1','C2',50);
INSERT INTO Orders(OrderID,CourseID,CustomerID,Discountpercent) values
('O3','D3','C3',60);

```

Inserting the following row of data will generate an error

```

INSERT INTO Orders(OrderID,CourseID,CustomerID,Discountpercent) values
('O4','D3','C4',60);

```

AZURE SQL MANAGED INSTANCE

DIFFERENCES

- We might not get all the features of the Microsoft SQL server by using Azure SQL server
- SQL server agent which can be used to execute jobs but in Azure SQL server we can't use that.
- If a company wants to migrate to azure and need less maintenance overhead and want to use all the features of SQL server, than we need to use Azure Managed Instance.

AZURE MANAGED INSTANCE ADVANTAGES:

- 100% compatibility with the latest SQL server Enterprise Edition
- Provides native virtual network implementation.
- High availability backups.
- It deploys on VNet, now we have a isolation, for security it is very useful.

LAB-CREATING AN AZURE SQL MANAGED INSTANCE RESOURCE

- Add SQL managed instance--Create.
- **BASICS TAB**
 - Subscription

- RG
- Name
- Region
- Compute +Storage-->configure-->no DTU model we need to choose service tier and compute hardware, backup storage redundancy, we will see our estimating cost.
- Credentials (username and pswd)
- **NETWORKING TAB**
 - Create a new VNet
 - Additional setting tab
- **TAGS**
- **REVIEW AND CREATE**
- Create krdo.
- Go to the resource
- From portal, Create a new VM in same VNet but in another subnet as managed instance.
- Go to VM--<connect to VM using the RDP file
- In server mgr.-->local server-->IE enhanced→on
- From internet explorer, download SSML studio.
- Open and launch SSML tool
- In portal go to managed instance copy the Host name
- Put that host name to the SSML tool and add credentials to connect to DB.
- We will have all features of the SQL server.

SERVER SIDE TRANSACTIONS

- We can create DB transaction; transaction can be set of DB commands that are executed to go ahead and perform a process.
- let's say we have a site where customers can go on and place orders, we might have many steps in a transaction like (create order-->charge the customer-->Generate receipt)
- End to end it should be encrypted.
- If one step goes wrong whole process goes again, roll back
- transactions can itself, it might be hitting different tables to go ahead and interact with the data.
- like for order it has SQL DB and for each like this.
- These tables can be part of different DB's
- if we are working with dotnet program, in which there are libraries in place, to go ahead and perform server-side transactions.

POINTS

- The server-side transactions are supported when:
- The databases can be multiple Azure SQL databases on same server.
- The databases can be multiple Azure SQL databases on different servers.
- The databases can also be hosted on Azure SQL managed Instance.

AZURE SQL DB--ACTIVE GEO REPLICATION

- Feature to make data more highly available.
- This feature allows you to create a readable secondary database.
- This database would be based on primary database.
- The secondary Database can on the same server or different server.
- This is not supported for Azure SQL managed Instance.

USE CASE SCENARIO

- We have an app working with data in primary database, and we have a lot of reports that have been made and basically working with data, now instead of reports working and pulling data from primary database and putting pressure on the primary database itself, which could in turn effect the app, the reports could actually be based on working with data in secondary database.
Primary database can be working with applications and secondary database can be used for serving the reports.
- Another scenario is based on failover, in this we are using active geo replication to replicate the primary database to a secondary database that is in a different region altogether, but our primary region goes down, than we can failover onto secondary database, failover must be initiated by user itself.

LAB-ACTIVE GEO REPLICATION

- We have an Azure SQL database here and connected to the DB using SSML.
- Create some tables in SSML tool.
- In portal SQL DB-->Geo Replication section-->choose the secondary region
- Choose krlo region-->we need to have target server in the region we are hosting secondary region, if dot we can create -->server name-->credentials-->location-->select
- Pricing tiers select krlo
- Ok
- Now the secondary region will be created.
- Go to secondary server -->set server firewall settings-->add client IP of our workstation-->take server name-->connect to this server using the SSML tool-->we will see our tables.
- We can stop the replication from primary database-->geo replication section

- We need to delete data manually and database also.

AZURE SQL DB--AUTO FAILOVER GROUPS

- This is a feature that is built on top of Active-Geo replication.
- This feature is available for azure QL Managed Instance.
- Here you can replicate and failover a group of databases on a server.
- The failover can be done manually pr automatically via a policy.

LAB-FAILOVER GROUP

- We have SQL Database-->failover group section-->Group name-->we need to create a server in different region
- Failover policy
- We can select databases to add
- Add krdo.

NO SQL DATABASES

NO SQL DATABASES

PRIMER

- We need no SQL DB bcoz, it gives more flexibility as in relational DB there are restrictions.
- We have different types of Np SQL databases for
 - Key/value pairs.
 - Documents
 - Graph

MAIN APPLICATION OF USING NO SQL DB

- Some application didn't need the overhead of maintaining relationships.
- The querying of data would sometimes lead to performance of overheads.
- You don't need to perform to any define schema.

INTRODUCTION TO AZURE COSMOS DB

- This is a fully managed BOSQL database.
- It provides fast response time.
- It provides high availability of your data.
- It is highly scalable.
- It carries our automatic updates and patching to the underlying database engine.

- One of the best parts about Cosmos DB is that we can create a resource, based on Cosmos DB, we can choose something known as API.
- Based on API we choose we can store data different types of data using Cosmos DB
- **TYPES ARE:**
 - **SQL API**--we can store SQL commands
 - **Mongo DB API**-- Use to store mongo Db data, data is stored in form of JSON documents.
 - **Gremlin API**--If we want to store data in form of graph, we can use this.
 - **Cassandra API**--Use to store Cassandra data we can use this.
 - **Table API** -- If we want to store data in form of Key/value pairs we can use this.

LAB-AZURE COSMOS DB

- In portal create a azure cosmos DB.
- **BASIC TAB**
 - Enter subscription
 - RG
 - Account Name
 - API
 - Free tier discount
 - Location
 - Account type (Production. Non production)
 - Geo Redundancy (enable or disable)
 - Multi-Region read writes (enable or disable)
- **NETWORKING TAB**
 - Connectivity methods
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
 - Create krdo
- Data Explorer (where we can create containers)
- Create new container
- Database ID
- Provisioned Throughput

PROVISIONED THROUGHPUT

- In Azure Cosmos Db you are charged for the throughput you provision, and the storage consumed on an hourly basis.
- Depending on the amount of throughput you provision, the right amount of CPU.
- You can provision throughput at Customer Level or the DB level.
- The cost to read a 1KB item is 1 request Unit.

- The other database operations will take their own amount of Request Units depending upon the operation

LAB CONTINUING

- We can see throughput estimated cost also there.
- Container id or name
- Indexing (auto off)
- Partition key (data will be spread ac. to that partition key), we are taking (/city)-->Ok
- In the container than we can add new item.
- We will have a new wizard for it.
- Add the JSON Document, to add data to table.
- We can see details of the item.

LAB-AZURE COSMOS DB -- WORKING WITH ITEMS

- Go to our items
- We will have extra properties rather than we defined, id [property is used to uniquely identify the item within partition and the partition key with the other id, is used to uniquely identify this item in cosmos DB.
- Add other items as well.
- We are creating another object also(orders)
- We are creating different items.
- Now we can click new SQL query and we will get our info.

LAB - AZURE COSMOS DB - WORKING WITH ITEMS

The following can be used as a reference for the last chapter

1. The following items can be added to the container

```
{
  "customerid":"C1",
  "customername":"UserA"
}

{
  "customerid":"C2",
  "customername":"UserB",
  "Orders":[
    {"Orderid":"O1","Course":"AZ-104","Price":100}
```

```

]
}
{
  "customerid":"C3",
  "customername":"UserC",
  "Orders":[
    {"orderid":"O2","Course":"AZ-104","Price":50},
    {"orderid":"O3","Course":"DP-900","Price":80}
  ]
}

```

2. You can run the below queries on the items

```
SELECT * FROM c.Orders
```

```
SELECT o.Orderid, o.Course FROM o IN Course.Orders where o.Course='DP-900'
```

COSMOS DB-PARTITION KEY

- Partitioning concept is used in cosmos Db for grouping the data.
- The data that is store in a container using partitioning, the data is segregated into multiple logical partitions.
- It becomes easier to query data based on partitioning.
- The underlying physical infrastructure the logical partitions are store on different physical infra.
- When we have millions of items that the partitioning improves the performance.
- Each partition can store up to 20Gb of data.
- Each partition can have unlimited number of partitions.
- Once we set partition we cannot change it, we need to create another container otherwise.

AZURE COSMOS DB-REPLICATING DATA

- One of the advantages of the cosmos DB is, that we can replicate our data.
- Suppose we have users in different location, and we want to our app to go ahead and fetch the data from closest location, we can increase read region and we can also create regions.
- In portal-->cosmos Db resource-->replicate globally section
- Choose the locations and we can add an additional read region.

- If our 1 region fails we can use secondary region

COSMOS DB-CONSISTENCY LEVELS

CONSISTENCY

- **STRONG**-You get consistency but losses the performance.
- **EVENTUAL**-You win on performance but loose on consistency.
- **BOUNDED STALENESS**-Here reads can lag behind the writes by at most "K" versions of an item but "T" time interval.
- **SESSION**- here reads can be guaranteed for the same session.
- **CONSISTENT PREFIX**--Here there is delay in the results of the most recent data, but you will never see out of order writes.

LAB-COSMOS DB--CONSISTENCY LEVELS--SETTING THE LEVEL

- In Cosmos DB, resource-->In default consistency level.
- We cannot choose strong consistency if we have location far away from each other.

LAB-AZURE COSMOS DB-Table API

- Create a new Cosmos DB account.
- **BASICS TAB**
 - Subscription
 - RG
 - Account Name
 - API
 - Location
 - Capacity mode(provisioned throughput/serverless)
 - Account type
 - Free tier acc.
- Create krdo.
- Go to resource
- Data explorer section
- New table
- Table id and bki sab same h in sense of throughput.
- Add.
- Go to the table-->add entity (same concept of partition key and row key(course ID and course Name)
- Add krdo details.

LAB-AZURE COSMOS DB-GRAPH API

- Create a new cosmos Db account and choose API as--Gremlin.

- Go to the resource-->data explorer section-->new graph-->DB Name-->Graph id-->partition key(/city)
- Go to that database-->graph-->query run krdo add nodes and vertices and edges.
- Execute krdo, here we can see the hierarchy.
- Load krenge toh we will see graph with nodes and vertices.

LAB - AZURE COSMOS DB - GRAPH API - RESOURCES

The following can be used as a reference for the previous chapter

1. Adding the vertices

```
g.addV('person').property('id','emp01').property('name','UserA').property('city','Chicago')
g.addV('person').property('id','emp02').property('name','UserB').property('city','Chicago')
g.addV('person').property('id','emp03').property('name','UserC').property('city','New York')
g.addV('department').property('id','DepartmentA').property('city','Chicago')
g.addV('department').property('id','DepartmentB').property('city','Chicago')
```

2. Adding the edges

```
g.V('emp01').addE('Manages').to(g.V('emp02'))
g.V('emp01').addE('Manages').to(g.V('emp03'))
g.V('emp02').addE('In').to(g.V('DepartmentA'))
g.V('emp03').addE('In').to(g.V('DepartmentB'))
```

COSMOS DB-EAMPLE REFERENCE ARCHITECTURE.

- Suppose we have an app which have infra in different regions we have traffic manager to route the traffic to different regions, than we have 2 LB's which can distribute the traffic onto the VM's in their locations, VM could have our web applications and that web app are using the Cosmos DB account to store the data.
- Now we had infra on another region, for global distribution, so users are closest to that region, so users would be reading the data from that infra., to ensure that app does not have to go that region to fetch data.
- It helps to reduce latency for the entire architecture.

LAB

- Cosmos DB Resource→ resource Globally sections and select the location and click save, we have added replication, now data will be replicated.

AZURE KEY VAULT SERVICE

AZURE KEY VAULT SERVICE

- It is cloud based service to hold secrets
- Secrets can be encryption keys, certificates and secrets.

LAB

- In portal search for key vault click create.
- **BASICS TAB**
 - Subscription
 - RG
 - Key vault name
 - Region Name
 - Pricing tier
 - Recovery options
- **ACCESS POLICY TAB**
 - Enable access to disk encryption
 - We can add user to which we want to give access to key, secret and certificate.
- **NETWORKING TAB**
 - Leave
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
 - Go to the resource-->in setting section we will see keys, secrets, and certificates.

AZURE KEY VAULT -- SECRETS

- Go to secrets section in key vault
- Click generate and import
- Upload options (Manual)
- Name
- Value(password)
- Create
- Using the visual studio, we can use this secret using dotnet program.
- To access this secret, we need to make use of Azure AD, where we would create an application object that will be represented by an dotnet program and we will give this application object access to azure key vault and then our dotnet program will now assume that application and authorize the key vault.
- In portal-->go to Azure AD-->App registration section
- Register app-->name-->leave all other as it is

- We can embed app in various ways, we will define env. variables on my machine than embed all details of the local machine.
- In that app we created copy the App Directory ID.
- Go to certificates and secrets section of azure AD-->new client secret-->name-->add-->copy value of secret.
- In our local machine→ control panel-->system-->advanced settings-->add env. variable (name and value) which is secret, client id and tenant id-->Ok
- Open program of visual studio again.
- In key vault--.add access policy (to give access to app) -->add the app object we created we will get the option to choose.-->add krdo-->save
- In visual studio run that program and we will see that value.

LAB - AZURE KEY VAULT - SECRETS - RESOURCES

You can also use the resources as a reference for the last chapter

You can set the following environment variables on your local machine. Ensure to change the values accordingly.

AZURE_CLIENT_ID="0bb234e8-ba4a-46b2-b3a7-939e9add8d6e"

AZURE_CLIENT_SECRET="mZ6h~DIyE58MnqW7M0V.UjkxdO67r.8KR~"

AZURE_TENANT_ID="5f5f1c90-abac-4ebe-88d7-0f3d121f967e"

MANAGING AZURE KEY VAULT SECRETS.

- We will make use of command in Azure CLI.
- With this command we can see list of the secrets in our key vault.
- Watch video.

Managing Azure Key Vault secrets

You can use the following commands for practice

// List the secrets in the key vault

```
az keyvault secret list --vault-name demovault9090
```

// List the value of a particular secret

```
az keyvault secret show --vault-name demovault9090 --id
"https://demovault9090.vault.azure.net/secrets/dbpassword"
```

// Mark a disabled secret and enabled

```
az keyvault secret set-attributes --name apppassword --vault-name demovault9090 --
version "03daa86af5894027890852d0d8e76d04" --enabled true
```

LAB-DISK SERVER-SIDE ENCRYPTION

- We will make use of encryption keys for the disk encryption.
- By server-side encryption, our data is encrypted by azure, if we want to use our own key for encryption we can use create and store using key vault.
- Create a new VM.
- In disks section of VM, add a data disk, choose encryption type, default is server-side encryption, accept krlo.
- Stop the VM
- In key vault-->keys-->generate and import ley--.name-->all others as it is.
- Add new resource-->disk encryption sets-->add-->create-->Subscription-->RG-->name-->region-->type (customer managed key) -->select the key vault the key we created (key vault, key and version) -->create krdo
- In key vault-->access policy section-->we will see the disk encryption set has permissions on key vault.
- In VM-->encryption section-->add the encryption-->choose the encryption set we created-->save
- Then we can start our VM.

LAB-AZURE DISK ENCRYPTION

- Create a new VM-->having 2 disks
- Disks section-->additional setting tab-->choose both disks to encrypt.
- In key vault -->access policies-->select disk encryption option checkbox
- On VM-->Disks section-->additional setting tab-->choose both disks to encrypt-->select key, key vault, version-->Ok
- We will have disk encryption in place.

ADDITIONAL POINTS

- We can enable azure disk encryption instead it our based on standard SSD or standard HDD.
- The availability is managed by azure.
- We can also create unmanaged disks, not managed by azure, these are stored in SA's, in this disk encryption is also supported.

AZURE KEY VAULT-FIREWALL SETTINGS

- In above we were using app object
- Key vault is a public service.
- For security if, we want only the network holding VM could access key, we can use this,
- We have a dotnet program and we were able to fetch value.

- In key vault-->networking section-->Firewall and VNet settings-->add existing VNet-->subscription-->select VNet in which we have our VM-->subnet(default)-->we need to have service endpoint
- So go to VM-->VNet-->service endpoints section--.add-->service (key vault) -->choose subnet(default)-->add
- Not add that service endpoint to the key vault-->Ok
- Not go to the visual studio and run program again we will get an error as our local machine will not be able to access now, only VM can access
- connect to VM--run that key vault MSI file we will get our pswd.

MANAGED SERVICE IDENTITY-SETUP

- We have given access to dotnet program to use key vault by adding env. variables on our local machine, but to give access to the VM using dotnet program we use other way
- Which is managed identity, in case of VM can be generated for VM itself-->that identity is stored in Azure AD
- That we can use similar App object and give access to managed identity onto the azure key vault.
- So, VM will use authentication token instead of client ID.

LAB

- We have a program in pace as zip file
- Create a windows VM
- connect to that VM using RDP file
- Right click on RDP file-->edit-->local resources-->more-->drives-->connect to c drive to copy the program onto the VM.
- in VM-->file explorer-->C drive-->folder jhn file h-->copy entire bin folder-->copy to D: drive of the VM.
- On the VM in portal-->identity section-->system assigned-->on krdo
- In key vault-->access policy-->add policy-->secret permissions (get and list) -->principle(search demovm)-->add-->save.
- Not inside VM-->open that bin folder-->run that program-->we will see the pswd

AZURE MONITOR SERVICE

AZURE MONITOR SERVICE

- It is a monitoring service.
- We can view the metrics for the azure resources like CPU Usage, Disk metrics and Network stats. We can also create alerts for them.
- We can also view activity logs, it is logs for all control plane activities. like when a VM is stopped or when a VM is created. we can also create alerts for this.

- All the logs are stored in log analytics workspace (It is a central solution of your logs).
- And then we have Application Insights, which is used for performance management for you live apps. It can be used by developers which can use these insights to look into performance of their web apps like they can see metrics on them.

WORKING WITH AZURE MONITOR

- Go to the monitor service in portal
- Metric sections
- we can choose subscription-->resource group->choose resources-->choose the metric and we will get graph.
- we can change representation of the graph and all.
- Activity log section-->we can see all operations and we can also download the logs file and see what actions has been performed.
- Alerts section, we can create alert for the CPU usage or acc. to any activity.
- Create an alert rule->subscription-->resource grp->select resources or VM's-->add condition (based on metrics like CPU percentage goes beyond the threshold)(or acc. to activity log lie to notify for all administrative activities)-->action grp create-->subscription-->RG-->action name-->email(mention email)-->add
- Rule name dedo and enable krdo and we will have a alert rule in place.
- Now stop any VM, now we will get an email for that.

AZURE MONITOR--DYNAMIC ALERTS

- Dynamic thresholds in azure monitor makes used of advanced ML.
- This technique makes use of historical behavior to identify any patterns or anomalies that might indicate possible service issues.
- When you choose to use dynamic thresholds, you must mention a setting known as sensitivity. There are three settings:
 - **HIGH**-Here an alert rule will be triggered on the smallest deviation that could result in more alerts.
 - **MEDIUM**-Here you have less tight and more balanced thresholds and fewer alerts will be generated.
 - **LOW**-Here alerts will only be triggered is there are large deviations.

LAB

- Go to monitor service-->Alerts section-->add alert-->choose resource-->in condition-->CPU-->In threshold option we have static and dynamic
- In dynamic it will work with ML and then highly decide threshold value and then if higher it will generate the alert.

LOG ANALYTICS WORKSPACE

- It is a central solution for all of your logs.

- Like we have VM and on-premises servers we can send all the data to the log analytics workspace.
- We have an Azure SQL DB and we want to send the audit information that we can also store in log analytics workspace.
- Once we have data available in workspace, we can use the Kusto query language to work on the data and also use various other solutions available in workspace.

LAB-CREATING A WORKSPACE

- Add a new resource-->log analytics workspace.
- **BASICS TAB**
 - Subscription
 - RG
 - Name
 - Region (same as resource to save on cost)
- **PRICING TIER TAB**
 - Leave
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
- Create
- Go to the resource
- Go to the VM section
- We will see our VM's
- when we will click a VM in workspace-->connect, it will install an agent (Microsoft agent onto that VM)
- In the VM-->extensions section-->we will see that agent (Microsoft monitoring agent).
- Go to workspace-->agent mgmt. section (we will get steps to install the agent on the VM manually also)
- connect to the VM on which we haven't install the agent using workspace
- Map the local drive of our workstation (contain the agent installer file) to that VM-->and connect using RDP file
- Open the file explorer on VM-->go to mapped drive copy that file onto VM
- Run that exe file-->mention workspace id (from portal) and primary key (from portal)-->install krdo.
- Now VM will send data to workspace.

LAB-CHOOSING DATA TO BE SENT TO THE WORKSPACE

- Go to the workspace-->Agent configuration section
- We can add windows event log tab--> add windows event log-->choose application (we will see all error, info and warning logs, not security logs)

- We can add windows performance counter-->add counter-->choose memory-->add.
- Then in workspace -->Logs section-->under log management we will have tables as it is streaming the data
- We will have table (perf)-->in query editor we can see the table->for all VM's.
- We can see all performance related info in workspace.

LOG ANALYTICS-QUERIES

- We will also have event table-->here we will see all our app logs for the VM's
- if we want to search for the event related to demovm (Event | search "demovm")
- Check the doc.

LAB - LOG ANALYTICS - QUERIES - RESOURCES

The following can be used as reference for the previous chapter

Event | search "demovm"

Event | take 5 (This is taken in no specific order)

Event | top 10 by TimeGenerated

Event | where EventLevel == 4

Event | where TimeGenerated > ago(5m)

Event | where TimeGenerated > ago(5m) | project EventLog, Computer

Perf | where TimeGenerated > ago(5m) | summarize count() by ObjectName, CounterName

Perf | where TimeGenerated > ago(5m) | summarize avg(CounterValue) by Computer

LOG ANALYTICS ALERTS

- We can create alerts based on the queries.
- When we write query in query editor in workspace we will see alert option-->enable that we need to see pricing model.

AZURE LOG ANALYTICS--SENDING CUSTOM LOGS

- here connecting to Linux VM using putty tool
- Install nginx web server on Linux VM
- In Linux VM -->go to folder-->/var--.log-->nginx-->ls-->we will see the access and error log.
- Using WINSXP tool type the public Ip and connect to VM using this tool
- Ab isme saari files khul jati vo vale log folder pe chle jao and copy krdo access log file krdo from LHS to RHS (hamare machine pe jaati)
- In workspace-->Customs logs section--.add-->browse the file from our machine→ok
- We will have ability to read data

- Next page-->choose the collection path-->Linux VM-->path dalo
Linux(/var/log/nginx/access.log)
- In details tab-->name-->next-->create krdo.
- In workspace-->advanced setting section-->Data tab-->custom log-->choose our custom log->save.
- Now it will start sending data to our workspace
- Try to open the home page of nginx server gain ana again to collect or see logs.
- In workspace-->logs section-->we will see our custom logs-->query run krdo and we will see table.

LAB-WORKING WITH DASHBOARDS

- We get an overview of how resources are behaving as part of our azure account.
- We have a widget for all resources, widget for quick start tutorial and many more.
- We can create our own dashboard as well.
- Go to monitor service-->metric section-->choose the metric for VM for CPU percentage-->we will see chart we can give name to chart and pin to a dashboard
- Add new dashboard-->we can add to private or shared
- Dashboard name-->Subscription-->publish-->create and pin
- In our dashboard from above select the dashboard name we will see that chart.
- We can change item and all from our dashboard for the required time we needed.
- In azure monitor create a new chart for SA's-->used capacity metric-->pin krdo-->choose dashboard
- We will that metric on the dashboard
- Further we can play with the size of the metric and move the positions and all and arrange them.
- We can also filter through dashboard, and we can also our auto refresh timing.
- Clicking On edit button we can add other widgets also, like user sign in summary, clock and all.

AZURE AUTOMATION

- **Process Automation**
 - Orchestrate processes using graphical
 - PowerShell
 - Python runbooks.
- **Configuration Management**
 - Collect inventory
 - Track changes
 - Configure desired state.
- **Update management**
 - Assess compliance

- Schedule update installation.
- **Shared capabilities**
 - Role based access control
 - Secure, global store for variables, credentials, certificates, connections
 - Flexible scheduling
 - Shared modules
 - Source control support
 - Auditing
 - Tags
- **Heterogeneous**
 - Windows & Linux
 - Azure and on-premises.

LAB-AZURE AUTOMATION

- We have an VM-->we will create a PowerShell runbook and it will run when an alert is triggered.
- We want the automated response or want something automatically done.
- In portal we will create automation account-->create-->name-->subscription-->RG-->Location-->run as azure acc (Yes)
- We have a script that will run (no need to know the script details).
- Webhookdata will extract details of the VM.
- we will attach this runbook to azure monitor service.
- So, that when the alert is generated, that alert would send the required data about the VM onto this script and then it will extract the details than it will stop the VM.
- Go to the automation account-->Modules section-->browse gallery-->search az. account-->import-->ok
- Go back to browse gallery -->Az. compute-->import-->Ok
- Go to automation account-->Runbook section-->Create runbook-->Name-->Runbook Type (PowerShell choose) -->Create
- Then the runbook will open and paste our script-->save-->publish
- Open azure monitor service→alerts section->new add alert-->search for our VM-->Done-->add condition-->network (we can choose nay to trigger the alert) -->add krdo-->

ADDING ACTION GROUP

- **BASICS TAB**
 - Action group add-->subscription-->RG-->name
- **NOTIFICATIONS TAB**

- Action Type (Automation Runbook) -->Runbook source (User)-->Subscription-->Automation account choose-->Runbook(VM)-->Ok

- **ACTIONS TAB**

- Action type
- name for an action

- **TAGS TAB**

- **REVIEW AND CREATE TAB**

- Create
- Ab alert rule m name dedo alert ka and add krdo
- In monitor service we will see the alert is generated.
- In automation account-->Jobs section-->we can see output-->succeeded
- and we will see our VM is in Stopped state.

LAB - AZURE AUTOMATION - RESOURCES

The following can be used as a reference for the last chapter

1. First add the following modules

Import Az.Accounts

Import Az.Compute

2. Then use the following in the PowerShell runbook

```
param (
    [object]$WebhookData
)
if ($WebhookData -ne $null) {
    $WebhookBody = (ConvertFrom-Json -InputObject $WebhookData.RequestBody)
    $Context = [object]$Body.context
    $AlertContext = [object] ($WebhookBody.data).context
    $SubId = $AlertContext.subscriptionId
    $ResourceGroupName = $AlertContext.resourceGroupName
    $ResourceType = $AlertContext.resourceType
    $ResourceName = $AlertContext.resourceName
    $ConnectionName = "AzureRunAsConnection"
    $Connection = Get-AutomationConnection -Name $ConnectionName
```

```

Add-AzAccount -ServicePrincipal -Tenant $Connection.TenantID -ApplicationId
$Connection.ApplicationID -CertificateThumbprint $Connection.CertificateThumbprint |
Write-Verbose

Set-AzContext -SubscriptionId $SubId -ErrorAction Stop | Write-Verbose

Stop-AzVM -Name $ResourceName -ResourceGroupName $ResourceGroupName -Force
}

```

UPDATE MANAGEMENT

- It is feature available as part of Azure automation.
- If we want to automate update on the windows machine or Linux machine, we can make use of update mgmt. in azure automation.
- Like OS have latest security updates we can use this service.
- We will have log analytic workspace, and these VM's will connected to these VM's→then an agent will be installed on this VM's
- And those machines will be sending the updates that it has on its machine to the workspace, then azure automation will query the workspace and look fr the updates required missing on the machines and then will do required implementation of the updates on those VM's.

LAB-UPDATE MANAGEMENT

- We have a VM in place in UK South location based on windows 2019 server.
- Add a new VM based on Linux-->region (North Europe) -->create krdo
- Create a automation account-->location (North Europe)
- Create a log analytics workspace also-->north Europe-->create
- Go to Automation resource-->update mgmt. section-->Link krdo automation acc. and log analytics workspace ko.
- Connect to the windows VM using RDP File.
- Search for updates on VM-->update settings-->we have updates available and install now and install these updates, but no.
- In automation account-->update mgmt. section-->add azure VM's tab-->choose the location (choose the Linux VM and enable krdo) it should be running state.
- Choose another location and add another VM also
- We will see extension(agent) the in VM on portal in extensions section.
- In workspace Both VM's are connected.
- After some time in update mgmt., we will the info, for both VM's we will see which updates missing.
- We can see the missing updates tab to check which updates missing.
- In log analytics workspace-->logs-->we will see update logs also.

- In automation acc-->update mgmt.-->we can schedule update deployment-->name-->OS (windows)-->machine to update(select)-->Ok-->update classification we can select the updates we want to install-->schedule-->start date and time-->reboot options-->create krdo.
- Now update deployment will install all update, after 1 hr. the machine will be in compliant now.

AZURE MONITOR PRICING

- We have different prices for alert, log analytics and metrics
- For log analytics we are paid acc. to storage of data like 5GB is free and bki check doc.
- For application insights also different
- Check the doc once

<https://azure.microsoft.com/en-in/pricing/details/monitor/>

DIAGNOSTICS FOR YOUR RESOURCES

- Here you can send the platform logs for azure resources to get detailed diagnostic and auditing information.
- We can stream logs to Log analytics workspace, SA"s and event hubs also.

LAB-AZURE ACTIVITY LOGS-DIAGNOSTICS

- Go to Monitor service->Activity logs section-->diagnostic settings tab

POINTS:

- You can send activity logs via a diagnostic setting onto an azure storage account, onto azure event Hub or onto a Log Analytics Workspace.
- The different categories of info that gets sent
 - **Administrative**-This contains the record of all create, update, delete and action operations that are performed via a Resource Manager.
 - **Service Health**-This contains the records of any service health incidents.
 - **Resource Health**--This contains records of any resource health events that occur on the Azure resources.
 - **Alert**-This gives the record of activations for azure alerts.
 - **Auto scale**-This contains events of any auto scale events that occurs as part of your subscription.
 - **Recommendation**--This gives recommendation events from Azure advisor.
 - **Security**-This contains the record of alerts generated from azure security center.
 - **Policy**-This contains the records of all effect operations that are performed by azure. policy.

LAB CONTINUED

- Create a resource Event Hub
- **EVENT HUB** -- It is a data ingestion service in azure, in this we can ingest data from various sources on to the event hub.
- **BASICS TAB**
 - Subscription
 - RG
 - namespace name
 - Location
 - Pricing tier(basic)
 - Throughput units(leave)
- **FEATURE TAB**
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
- Create
- Create a new resource stream analytics job
- Subscription-->RG-->Location-->leave everything as it is.
- Create krdo.
- Go to event hub-->create new-->name-->bki sab same-->create
- Go back to diagnostic setting -->name-->log(administrative)-->send to log analytics workspace-->Name and select our workspace-->select to stream to an event hub-->name-->save.
- Now logs will stream n both event hub and workspace.
- In stream analytics job-->add stream-->name-->event hub name-->JSON-->ok
- In log analytics workspace-->go to table of azure activity-->run-->we will see all our activity logs.
- In stream job-->Query section-->input select our-->we will see our table for log(to see structure of the data)

LAB-DIAGNOSTICS--VM's

- To get more info about our underlying VM.
- We don't have a metric for memory in monitor service to see memory usage.
- So, we have a VM-->diagnostic settings section-->we need a SA, can use existing SA which is used for booth diagnostics also, select a storage acc-->enable-->choose performance counter (CPU<MEMORY.....)-->Choose the event logs(critical,Error,Warning), we can also choose IIS logs and many more-->Save.
- diagnostic settings save krdo.
- Go to Storage Explorer-->go to your SA-->we will see our metrics-->we will get counter values, and this is stored in our SA.

LAB-DIAGNOSTICS-SA's

- You can capture metrics for you SA and also gets logs about how data is being accessed in your SA with help of diagnostics.
- With the logging facility, the following types of requests are logged:
- Successful requests
- Failed requests, including timeout, authorization, and other users.
- Requests that use a Shared Access Signature, OAuth.
- Requests that are made by the storage service itself.
- Event anonymous requests are logged.
- The logs are stored in \$logs container.
- There are 2 logs entry formats-version 1.0 and version 2.0
- Version 2.0 adds fields for logging information about the requests made to the Blob and Queue services that are authorized with the use of OAuth 2.0 token.

LAB

- We have a storage account in place-->diagnostics settings section
- We have tab of properties for each type of SA (Blob, File, Queue, table).
- In blob properties we have enable-->Metrics, minute metrics and also logging version(read/write/delete) -->save krdo
- All of these log data is stored in the SA account only.
- We will use SE, to view metric data-->these will be stored as separate tables.
- We will have tables for both minute and hour.
- We have logs data separately in (\$log folder).

AZURE FUNCTIONS-METRICS

- You have a lot of metrics available for azure functions.
 - **Response Time**-This is the time taken for the application to server requests.
 - **Average memory working set**-This is the average amount of memory used by the application in MB.
 - **Data In**-This is amount of incoming bandwidth that is consumed by the application.
 - **Data Out**-This is amount of outgoing bandwidth that is consumed by the application.
 - **Function Execution Count**-This is the number of times the function was executed.
 - **Function Execution Units**-This is the amount of MB used per milliseconds.

LAB

- In portal search for function App

- Create a function app
- **BASICS TAB**
 - Subscription
 - RG
 - App name
 - Publish (APP/Docker)
 - Runtime stack (>NET Core)
 - Version
 - Region
- **HOSTING TAB**
- **MONITORING TAB**
- **TAGS TAB**
- **REVIEW AND CREATE TAB**
- Create a krdo
- Create a function
- Default HTTP trigger--Create
- In code+test section-->input value and parameter and run krke dekhlo.
- in monitor service-->metrics-->apps service (resource type m) -->select the functions-->metric select krlo (Http ki)-->ok

AZURE FUNCTIONS-DIAGNOSTICS SETTINGS

- In Function app-->diagnostics settings we can sent data to SA, event hub, log analytics workspace.
- we can also archive on to the SA.
- Destination select krne ke bd save krdo
- After some time, go to workspace->we will see logs table for function app.
- We can run query to get data also.

AZURE APP SERVICE PLAN --METRICS

- If you are using an App Service Plan that is based on the Basic, Standard or premium tier, then below are some of the metrics you can get from Azure monitor.
 - **CPU Percentage**--This is the average CPU used across all instances of the App service plan.
 - **Memory percentage**--This is an average Memory used across all instances of the App service plan.
 - **Data In**--This is the average incoming bandwidth that is used across all instances of the App Service Plan.
 - **Data Out**-- This is the average outgoing bandwidth that is used across all instances of App Service Plan.

AZURE WEB APPS --METRICS

- You have a lot of metrics available for the web apps as well.
 - **Response Time**--This is the time take for the app to server requests.
 - **Average memory working set**--This is the average amount of memory used by the app in MB.
 - **Data In**--This is the amount of incoming bandwidth that is consumed by the app.
 - **Data Out**-- This is the amount of outgoing bandwidth that is consumed by the app.

LAB

- Create a new web app.
- Using visual studio-->publish our app
- Exercise different parts of app
- Go to the monitor service-->metrics section-->select the app service-->select metrics-->we can see different metrics
- Go to the monitor service-->metrics section-->select the app service plan-->we can choose metrics acc. to it, it will have different metrics.

AZURE WEB APPS--DIAGNOSTICS

- In app web app-->diagnostics settings section-->we can add to SA, Event HUB, Log analytics workspace.
- We can add multiple diagnostics sections.
- Choose the logs and destination (can choose 1 or more).
- Exercise different parts of your app
- Go to you log analytics workspace-->logs sections-->we will see the app service logs

INTRODUCTION TO RESOURCE TAGS

INTRODUCTION TO RESOURCE TAGS

- We have tags option for all the resources.
- In tag option we need to specify name and value pair.
- Suppose company has multiple departments (HR, Logistics, Procurement, IT)
- And the VM is used by HR and logistics use AZURE SQL and VM's for other depts.
- It helps to logically grouping the resources.
- Now a particular resource can be staggered in multiple depts, we can put tag to that resource.
- We can have multiple tags to the resource.
- We want billing acc. to different depts.
- It helps to organize the resources.
- When we search for tags, we can see all tags and can search resources acc. to dept. wise.
- We can see cost analysis, acc. to the tags name.

- After creating we can also add tag.

LAB--COSTING IN AZURE

- Azure has many ways to tackle costs.
- Cost Analysis as part of your subscription
- **Overview**
 - Here you can see the current spending.
 - See spending per resource.
 - See your forecasts.
- **Cost Analysis**
 - See your spending history
 - See the spending based on tags, resource types etc.
- Azure Advisor.

LAB

- In portal-->cost management and billing section-->billing section
- Cost analysis section also--> we can download report from here
- In azure advisor we can see various recommendations-->what we can do to save on costs.

AZURE ADVISOR

- It is recommendation tool in azure.
- It give recommendation based on
 - Costs
 - Security
 - Reliability
 - Operational excellence
 - Performance
- we can download the pdf or csv file also.