AAD

Tiers:

- **Azure Active Directory Free.** Provides user and group management, on-premises directory synchronization, basic reports, self-service password change for cloud users, and single sign-on across Azure, Office 365, and many popular SaaS apps.
- **Azure Active Directory Premium P1.** In addition to the Free features, P1 also lets your hybrid users access both on-premises and cloud resources. It also supports advanced administration, such as dynamic groups, self-service group management, Microsoft Identity Manager (an on-premises identity and access management suite) and cloud write-back capabilities, which allow self-service password reset for your on-premises users.
- **Azure Active Directory Premium P2.** In addition to the Free and P1 features, P2 also offers Azure Active Directory Identity Protection to help provide risk-based Conditional Access to your apps and critical company data and Privileged Identity Management to help discover, restrict, and monitor administrators and their access to resources and to provide just-in-time access when needed.
- **"Pay as you go" feature licenses.** You can also get additional feature licenses, such as Azure Active Directory Business-to-Customer (B2C). B2C can help you provide identity and access management solutions for your customer-facing apps. For more information, see Azure Active Directory B2C documentation.

Multiple subscriptions can trust the same Azure AD directory. Each subscription can only trust a single directory.

From <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-how-subscriptions-associated-directory>

To set MFA (free) : we go to all users, in the top we click on the ellipsis , choose MFA, in this screen we can choose users to enable MFA and also choose the method to confirm MFA (email, sms ..)
If we need to have multiple users, or admins or set of users to have MFA (needs premium p2) , we can use policies:
conditional access policy : define a group of users to use MFA for a specific registered app or all.
Steps : left menu in AAD "Conditional Access", top menu we click add policy: we set use/groups,

cloud apps, then the condition (location,) , then we set the Access controls : Grant/block access, force MFA ..

AD : 3 ways of sync

1-password Hash: AAD stores the hash of the hash, verification is done in AAD , no SSO , other yes , seamless SSO yes for PTA also but not ADFS

2-Federation (using AD FS, because there is another 3rd party tool) : more complex, ADFS & WAP servers are required , verification is done in local AD (on prem) , seamless SSO is off until we enable it in other ways, items we need : server of AD FS and AD DC , server for web application proxy server (WAP)

3-Pass through authen PTA : need to install an agent in local server (on prem AD) , which communicate with AAD via service bus , verification as above.

AD connect simplifies hybrid cloud and federation, to be installed in local server (on-prem) to perform the hybrid identity (hash sync, PTA, Federation)

ADD custom domain : in the left blade, custom domain , we can add new one like contoso.com and we need to verify that we own it (by adding a given CNAME record to our DNS)

-Domain Controller : manage and centralize the management and authentication of set of computers, it's part of the AD domaine service . this server has to have static private IP + DNS

-->process : there is main server = the domain controller , we set up there the Ad DS (AD , DNS , Static ip, name of the netBIOS (company\)) , then the computer 1 (in same network) needs to join that domain,

by changing the domain name in the computer settings, we set the DNS IP address (exists in the main server) , we need to authenticate for this operation as domain admin : result-> the computer 1 is part of the domain , which means users can authenticate using the same AD as main server , computer 1 can communicate with other resources of domain (printer, other computers ..)

We have 3 options to join a VM to a domain :

1. join to on-prem AD DS
2. Join AAD DS (below) https://docs.microsoft.com/en-us/azure/active-directory-domain-services/join-windows-vm
3. Join AAD directly using ad-on(app registered in the backend) , this is in preview (https://docs.microsoft.com/en-us/azure/active-directory/devices/howto-vm-sign-in-azure-ad-windows)

Join a VM to a AAD DS : so that the users can use their corporate credentials to log in , steps (wizard):

1. Create in azure AD DS (managed AD DS as PaaS without the need to have AD DC),
2. attach it with a subnet in VNET (subnet must be dedicated only for it)
3. Configure admin group : add users who can administrate the AD Domain controller
4. Configure the DNS , in the VNET , change DNS from default(azure) to custom and put the 2 ip address given by the AD DS (created in step 1)
5. enable password sync (from AAD to our managed AD in AD DS) (we should reset the password of the users if we want to sync them)

Azure Active Directory (Azure AD) **access reviews (P2 licence)** enable organizations to efficiently manage group memberships, access to enterprise applications, and role assignments. User's access can be reviewed on a regular basis to make sure only the right people have continued access.

From <https://docs.microsoft.com/en-us/azure/active-directory/governance/access-reviews-overview>

---->**Networking**

VNET peering is possible also cross region and cross subscription : no cost unlike VPN
To enable VNET peering : in 1st vnet go to peering and add the peering to the 2nd , it became initiated , do the same to the 2nd one. then it became Active.
resources in the same vnet (different subnets can connect by default , unless you restrict it in NSG)
IPs range should not overlap
VNET to VNET (site to site vpn) is possible also : there is cost
1 and only 1 dedicated VPN subnet must be reserved and created

Route table in azure: Azure automatically routes traffic between Azure subnets, virtual networks, and on-premises networks. If you want to change any of Azure's default routing, you do so by creating a route table. Steps: new resource Route table, we associate it with a subnet in VNET (can be associated to 0 or many subnets)

How Azure selects a route

When outbound traffic is sent from a subnet, Azure selects a route based on the destination IP address, using the longest prefix match algorithm. For example, a route table has two routes: One route specifies the 10.0.0.0/24 address prefix, while the other route specifies the 10.0.0.0/16 address prefix. Azure routes traffic destined for 10.0.0.5, to the next hop type specified in the route with the 10.0.0.0/24 address prefix, because 10.0.0.0/24 is a longer prefix than 10.0.0.0/16, even though 10.0.0.5 is within both address prefixes.

From <https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview#how-azure-selects-a-route>

0.0.0.0/0 address prefix

A route with the 0.0.0.0/0 address prefix instructs Azure how to route traffic destined for an IP address that is not within the address prefix of any other route in a subnet's route table.

From <https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview#how-azure-selects-a-route>

-NSG can be associated to a subnet or a VM network interface

-**point to site vpn** :
Create V. network gateway, type:VPN , we choose SKU, and route-based or policy based , SET the VNET in question (to where we want to connect to, it must have a gateway subnet) , and Public IP ,
After creating the vpn we go to the point-to-site configuration menu, we set the address pool (every client will get an address from this range) , we set the protocol (SSL VPN, IKEv2 VPN), authen type : Azure cert, then we create a certificate using for ex selfsign in, we set the public key in same config, then we save and download in the top the VPN client .
The vpn client is customized for us (using our config and SSL cert) , connecting will get us a private ip address from the pool we set earlier.

-**site to site** : we create local network gateway (refers to your on-premises location), we set public ip (of the on-prem router) , and address space(like address pool in point to site, here is the on-prem address space), then we need to link it to the virtual net gateway : in the VNG we go to connections , add , we set the Local NG, and a shared key.
In the on-prem, we need to set up a RRAS server, we install a vpn feature in the windows server, we specify the public ip of our azure network gateway, then it should be marked as connected in azure.

-**Azure Relay service :** small scale vpn connection, for hybrid connections, allow us for example to run an azure web app which is connected to an on-premises DB (an agent called Hybrid connection manager to be installed in on-prem machine which allows this SSL connection). without opening a port on your firewall.

Gateway transit is a peering property that enables one virtual network to utilize the VPN gateway in the peered virtual network for cross-premises or VNet-to-VNet connectivity

From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-peering-gateway-transit>

Protocols used by each VPN:
Site to Site : **IPSec** (IKE v1 and IKE v2)
Point to Site : **openVPN** (= SSL VPN, for Android, ios, Win, linux, MAc), **SSTP**(max 128 connection, Win 7) or **IKE v2** (from Mac)
VNEt to VNET like S2S (different than Vnet peering)

Service endpoint:
to allow access to PaaS services from a specific network or subnet instead of internet , 1-we go to storage account for example, we set the subnet in question(we allow in the firewall connection from the subnet from the menu : "Firewall and Virtual Networks"), 2-and in the vnet menu : the "service endpoints" left pane, we add storage account endpoint and the subnet to allow this service.

--> meaning : a resource within this Vnet can connect DIRECTLY (not over internet, using Azure backbone) to Storage account as soon it's in the subnet in question .
To do the same for Azure SQL, we add Microsoft.sql endpoint in the Service EndPoint of the VNET, and in the SQL DB firewall we allow the subnet to access to our server.

-service endpoint vs **Azure private link** : The main difference between the two is – Service endpoint uses the public IP address of the PaaS Service when accessing the service. Private Link introduces a private IP for a given instance of the PaaS Service and the service is accessed via the private IP
To create private link in portal : set resource type : Microsoft.Keyvault or Microsoft.sql .. , we specify the resource in question , we select the subnet in question, Enable private DNS zone (our resources will have a private ip) done. Now if we go to the blade of the azure private link created, menu private endpoints : we can see the resource attached (keyvault or sql db , with the private IP attached)
We can also attach resources to private link during creation, in the networking step.

DNS:
in azure we can't by a domain name, azure only host the owned domains for records managements.

Public **DNS**
-in azure we have Public DNS domain , where we add our domain name (owned) and sub domains : ex rachid.com and site.rachid.com
then in apps , or PaaS services, we can add custom domain name, like in function we have function1.azuresite.microsoft.com , we can have func.rachid.com , in this step we will need to add a record in our public DNS to verify that we own it.

Type of records:
A : is for IP addresses
MX : for mail eXchange
AAAA : ip v6
CNAME : alias (redirect two domain names : ex www.example.com to example.com , and we have example.com A 32.43.4.2)
Alias: not standard: For example, if your domain is example.com and you want it to point to a host name like myapp.herokuapp.com, you can't use a CNAME record
NS : NS record delegates a subdomain to a set of name servers , ex dnsimple.com. 172800 IN NS ns1.dnsimple.com.
dnsimple.com. 172800 IN NS ns2.dnsimple.com.
dnsimple.com. 172800 IN NS ns3.dnsimple.com.

private DNS zone :
not resolved in the internet , it can be set up to "Custom" or "default" in the virtual network : left pane , DNS servers . (default will use the azure provided one)
a dns name then is assigned to any VM in the virtual network
to manage records , we search "private dns zone" resource , then here we have different hostnames with mapped private IP address
Private dns is same in on-premises when we set the dns manually in the properties of TCP-IP .. We never do this in azure VMs, we rather do it as described above.
Private custom dns is needed when we configure AD DS, we need our machine to use the DNS of our AD DS to resolve the ip addresses when it joins the domain.

DNs end----------

IP Prefix:
an azure resource,
allocate set of (up to 16) IPs for set them in firewall for the future .. the whole range is billed (every reserved ip address is billed!)

-SSL certificate :
we upload a .pfx , then we can bind them to domain names (exmaple.com or www.example.com) , every domain and subdomain has one SLL cert registered with the domain in question (we specify the domain when we create the certificate)
or we could create the certificate with *.example.com to allow biding to many domains
type : SNI SSL , IP SSL. (1st one can be mapped to different domain names , 2nd is mapped to one specific ip address)

App gateway :
working on layer 7 of OSI , needs to have a separate subnet, components , the app gateway redirect the traffic to the backend pool depending on the rules set.
1-frontend (IP address either private or public or both) , 2-backend pool (one or set of resources : can be VM or IP or hostname , VM scale set , app service) :
. 3-listener : protocol and port name to listen to (ex : http:8080) . 4-rules : will connect a listener to a specific backend pool service (ex : whenever we get smtng from listener1 { http:8080 } sent traffic to the backed pool)
ALSO : need to set up http settings (cookies , connection draining , port ..) that will be linked to the rule.
ex : if we have in the backend pool BP with 2 VMs with http website port 80, the app gateway will sent the traffic to the BP , so either vm1 will receive the request or vm2 (high availability)
another example , if we want to redirect /images to VM1 and /files to VM2
WAF : web application firewall : prevent from some attacks: XSS , SQL injection .. , inspect request body : file upload size, max request body . Not included in SKU=Small

Load balancer :
distributes inbound requests to pools of systems , works in the layer 4 of OSI (transport) (it is more for VMs, VM ScaleSet, unlike ap gateway which is for http/https targets)
External LB : maps public IP address and port to internal targets (we set public ip during creation)
Internal LB : directs traffic only between internal resources, usig private ip addresses (we don't set public ip during creation)
frontend config : we set the public ip addressee (we can have more than 1)
backend pool : we specify ip v4 , then we associate it either with availability set, single VM, VM scale set.
Rules : the rules specify what frontend (IP) and traffic to redirect and to who : we can set our 1st IP as frontend IP address, protocol to TCP and port to 80 and backend to our backend . -> result : if we call in the browser the frontend ip selected, we will get a HTTP response from vm1 and sometimes from vm2.
health probes : check the health of the backend pool , HTTP:80 or TCP check , no send traffic if it's down. (note TCP unlike app gateway)

NAT port forwarding: we can connect to one of the VMs port , ex when we perform RDP (port 2334) to the frontend IP address of the LB, the LB will redirect this traffic to VM1 same/different port
use case : scale set of 2 instances where IIS is installed, load balancer provides high availably by redirecting traffic to either VMs

Azure Traffic manager : dns based , allows to control traffic distribution, routing method : Priority , Performances (by regions) , Weighted , Geographic
To Create: first create Traffic manager profile (as example routing type : Geographic) , then we set from the blade the Endpoints : as example : we add an endpoint (URL or app service) for a site for US , and other one for EU. Then we go to the overview of the traffic manager, we copy the URL , if we browse it from Us we get the US website, from Eu we get Eu website

Azure front door : improve high availability and scalability, it's http based load balancer , ex we have 2 app services, we want to route the user to the closer, or if 1 is down we route to the other one.
Steps : 1-durin creation we set front door : dns as x.azurefd.com , 2- add backend pool : at least 2 resource (ex 2 app services) , the health prob is checking the health of our backend endpoint frequently to know to which app should route the end user. During the prev step we can set priority and weight for routing. 3-add routing rule:maps the front end to backend
AFD vs application gateway : While both Front Door and Application Gateway are layer 7 (HTTP/HTTPS) load balancers, the primary difference is that Front Door is a global service whereas Application Gateway is a regional serv.

VM **disk encryption** :
Virtual disks on Windows VMs are encrypted at rest by using BitLocker. There's no charge for encrypting virtual disks in Azure. Cryptographic keys are stored in an Azure Key Vault by using software protection
By default managed disk are encrypted at rest but we can add another layer of security by encrypting our disks, steps : 1-create key vault in same location as the VM, 2-in the key vault menu "Access policies" Enable Azure disk encryption access , 3-call Set-AzVMDiskEncryptionExtension with params : VM, keyVault url , keyvault key url , OR from the portal : in the VM , meny Disks, in the top choose Encryption, choose OS &/or data to encrypt , select key vault and key , Save. Done the VM will reboot

----> **Storage account**

Storage account name must be lower case && UNIQUE in azure
Kind : General purpose V2 [supported Replication : like v1 + ZRS, GZRS,RA-GZRS] , V1 (there is no hot,cool,archive options unlike V2, [LRS, GRS, RA-GRS]) , blobStorage (no premium tier)
Premium tier is good for storing unmanaged disks.
Replication:

-LRS : 3 copies of data locally (same data center) ,protect from : if the servers rack goes down we still have 2 other copies

-GRS : data replicated to another data center in another region, so we whave 6 copies of data, protect from : data center failure, region outage

-RA-GRS : replicated as GRS + data can be read from the replica without MS initiating a failover, we got a second endpoint :youraccount-secondary.blob.core.windows.net, Access key are same.

-ZRS : 3 copies of data in the same region in different availability zones, available only for Standard and GPv2 , protect from data center failure

2 access keys : first key, secondary key , must be regenerated regularly , we should point the app to use secondary key then we generate first key , then we point the app to use first key and we generate the 2nd key. Access keys Enable access to the WHOLE storage account (files , blob ..)
Shared access signatures (SAS): allow access to a specify storage service : blob , files .. , we can also specify expire date , Read/write/add , IP range ... , it is singed with access key (key1 or key2)
Access policy : we can add another layer of access security on a specify service, ex we can create a table and add a policy that it's for read-only for a period of time or forever (by clicking on the table , then on the ellipses "Access policy")
Encryption at rest : by default storage account data is encrypted using managed key, we can use our own key from the key vault.

Files storage:
file share, supports SMB protocol, we can mount it to 1 or multiple VMs or on prem (for this one we need TCP outbound 445 open)
Azure file Sync service : sync azure file share (cloud endpoint) with on-prem or other vm (server endpoint, this on-prem server has to install Azure file sync agent)
Steps : we create sync group where we choose our storage account and the file share in question, we create our registered server (on-prem or other VM as destination)
The sync frequency is 24h
The file share and the file sync service must be in the same region
AAD storage account : enable this from "Configuration" menu , then we assign RBAC to our users.
Failover: if we choose GRS replication we can trigger a failover to the secondary storage acc endpoint, which means we will be using the secondary as primary (this might occur some files lost which are not yet synced to the secondary acc) , after failover our account will be LRS , we need to set it to GRS or RA-GRS if we want to (this will start the process of replication)

Diagnostics: we can view logs to see rejected requests (API call) or to diagnostic any issue : left menu "under Monitoring-Diagnostic" enable . A hidden folder with the name $logs will be created in the blob service, where our logs will be stored (we have chosen retention period when we enabled this logs)

Export import data to azure :
To import , the disk must enable bitlocker (we need to save the key), prepare the disk using the tool **WAimportExport**.exe. (v1 of this tool if we want to import data to blob, v2 to files)
To export : ship the empty disk to azure, configure blobs to export, MS ships back the drive to us, PS : **we can export from BLOBs onl**y.

Steps to import in portal : we prepare our disk to ship in the VM or the on-premises using the tool VWImportExport.exe , we create in the portal the resource "Import/export job", here we need to specify Type:import, and upload the journal file (create with the tool in prev step) , import destination (storage account in question) , we select delivery company and we put our address then next step is to ship that disk to MS address specified in the end.
Steps to export in the portal : type : export , we choose the storage acc to export (ONLY BLOBs) , we can choose whole storage acc or a specific container or blob, we set our address. Then we ship the empty drive to MS, MS will export data to them and ship them back to us , we go to the job resource we created in pre steps to get the Bitlocker keys in order to decrypt our files received from MS.

Access Policy in BLOB contains : we can set a policy which allow to read, list, write but not delete. There is also a policy for immutable blob : can write and read but never delete.

CDN : improve performances , rapid response and minimize latency , by caching the static content (storage account, app service..) in POP servers, the user accesses to this Pop server (distributed servers) instead of original endpoint. The file stays until TTL expires or 7 days if not specified
<endpoint name>.azureedge.net, if we put in path : /containerName and we have a file x.png there , then we can call it via edge like this : <endpoint name>.azureedge.net/x.png
For web apps : if we have static public site we will use the edge URL instead of the origin
Steps : we create CDN resource with Tier we want , then after we create inside an endpoint (name.azureedge.net), origin type : storage account, web app .., URL origin of resource.

-VM unmanaged disk: we need to have a storage account for them , the data will be stored there

CORS is an HTTP feature that enables a web application running under one domain to access resources in another domain. Web browsers implement a security restriction known as same-origin policy that prevents a web page from calling APIs in a different domain; CORS provides a secure way to allow one domain (the origin domain) to call APIs in another domain

From <https://docs.microsoft.com/en-gb/rest/api/storageservices/cross-origin-resource-sharing--cors--support-for-the-azure-storage-services>

---> backup
VM **backup** :

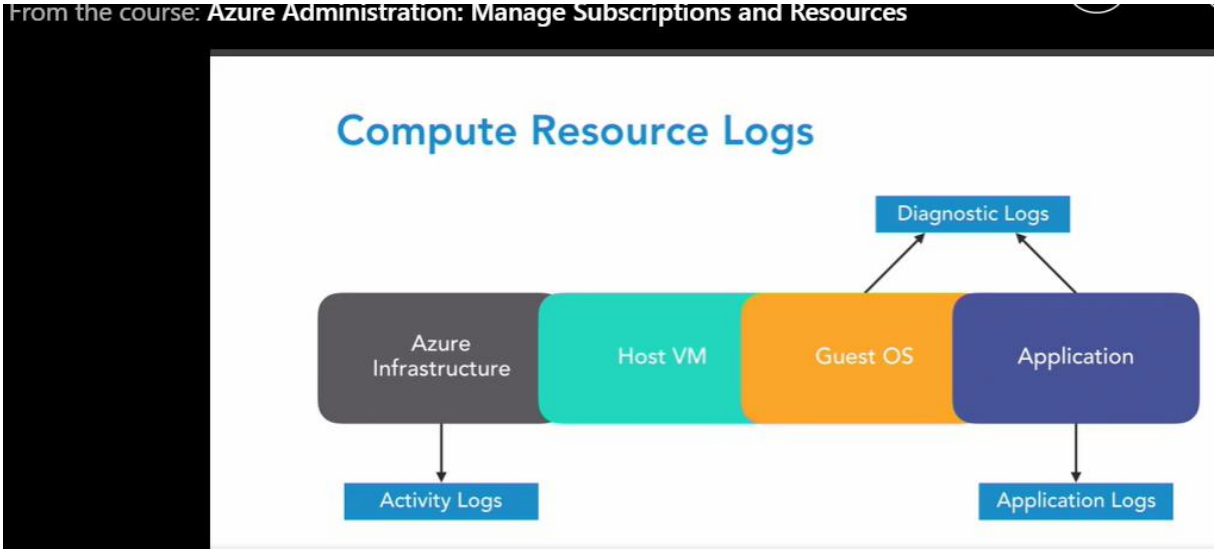
Backups are geo-replication by default , we can change this during creation to Local redundant. No agent is needed to be installed.
-We create **Recovery service vault (**we don't ned to create this resource, it's already exists , we create just a vault**)** , resources need to be in the same geo location as the vault, we create a policy for type=VM (then we specify frequency , retention ..) . now we go to the VM, menu backup, we specify our recovery service vault, we select our policy , then OK.
-For multiple VMs, we can do it from the vault, we click create from the top, we select Azure Vs On-premises, we choose between VM, SQL serverInVM, Azure fileshare. In our case we choose VM,
We select policy, done. PS : VMs have to be in the same region **and** not already protected (backed up)

Restore VM :
Restore VM : will not overwrite the origin VM, it will create for us a new VM , we can choose resource group , Vnet for it .. Doest need to have the original VM
Restore Disks : will overwrite our existing disk , this requires the original VM to be existed.
TO perform backup : we go to rec service vault, menu backup items , we select our item : there is option to restore a file (file recovery) , in our case we choose restore VM, we choose restore point, then either create new or replace existing (replace disk)
We can monitor the backup operations, we go the menu backup jobs : we can see the status of backup and restore
To BACKip files : We need to do this step in the VM in question : we open portal there , we choose "File recovery" , select recovery point, download script , we run it : result : new disk mounted with the recovered files (if the disk is bitlocked you need key) , last step is to unmount the disk from same page in portal

--> **alerts - monitor , logs**

Azure monitor is the central repo for Alerts , Metrics and Logs
In azure there are 4 type of logs : activity logs , OS-level diagnostic logs, application logs, diagnostic logs
Activity logs : subscription logs and events (put post and delete event) , to check by who and when a resource was created
Application logs : used by web and desktop app to log events
OS-level diagnostic logs: an agent is gathering data from azure resource ex:VM
Diagnostic clogs : 2 types : tenant logs : for resources outside azure subscription ex :AAD , Resource logs : for resources within azure subscription ex : storage account

Diagnostic logs can either redirected to :OMS log analytics, Event hubs, or Azure storage



We can create diagnostic logs from a VM for exm in the menu "Diag setting" , we choose metrics.. And storage account where to save these logs, disk quota.

Or from Azure Monitor, Menu "diag settings" , here we see all our resources that have diag logs, we choose our resource and we configure it (in azure monitor we can choose to send our logs to either : storage acc , event hubs, OMS) , this is not feasible from the resource!

Spending limit : to remove it we go to the service "Cost management and billing" , select the subscription , in the top there is a message "…X $ remaining, click here to remove your spending limit".
If you reach the limit, resources in the production are removed, VMs are stopped and de-allocated

Azure adviser : get security, high availability, cost , performances recommendation about resources , ex :recom to resize or shutdown VMs if the memory and cpu (<5%) are every low for a period of time (14 days), we can changes and add this rules

Log Analytics: we can access it from Azure monitor side menu. Here we can write query to get some insights like search * | where (type=update) to get update logs for all resources. We can set alerts.

KQL : kusto query language: to query logs in azure monitor

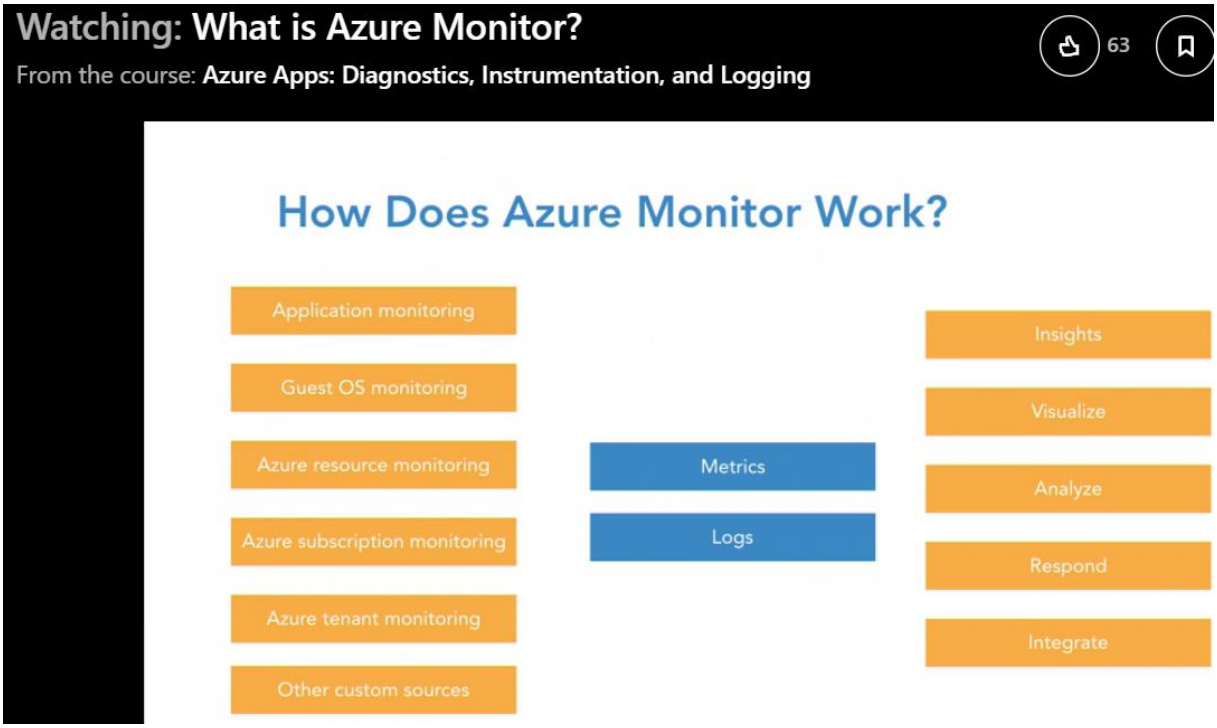
Create budget : get a notification whenever we exceed a budget , steps : in the service "Cost mang and billing" , left menu cost management, then left menu "Budget", we choose the scope (subscription or a resource group), we set a budget amount, and period of the invoice, we set the alert (action group ..)

Action groups : set of alert operations to be used for different alert and monitoring , ex : we set an action to email + sms owners then whenever we create alert we can set this action group so that the alert will be sent as email and sms to the owners , **steps** : monitor, left menu "Alerts", in the top menu "Manage action groups" , Add , we set resource group, action type : Email/sms, webhook , azure function ..

App insight: monitor logs, traces .. From azure or on-prem applications, you can see response time, sessions, perfs ..

Application diagnostic logs: from Visual studio we can log information/errors/warning : steps : in code we log the logs using the code System.Diagnostic.Trace.TraceError("exception …") , then we deploy the project in app service, and in left menu "Diagnostic logs" we set application logging (file system) to "On". Now if we go to the menu log system, we can see everything we log from the code

Metrics : point in time numbers , **Logs** : historical records



Azure Monitor menu :
Activity logs : azure resources logs , creation , listing resources … by who what time
Alerts : send notif whenever a condition is met (based on a metric : cpu , memory ..)
Metrics : point of time of a resource performances (cpu , response time ..) we can pin the chart into the dashboard
Logs : to work with logs we need to : enable this in the resource ex: in the VM m left menu "Diagnostics setting", choose metrics and logs to be collected, storage account . Then we need also to create log analytics workspace resource, and here in the left menu "Workspace data source -

VM" we need to connect our VM . Finally now we can see in our Azure Monitor the logs using KQL queries.
Service health: global azure , planned maintenance, service issues ..
Auto scale, we can get notif whenever auto-scale takes place
Usage and estimated cost: monitor and get and estimation for the month

---> Azure **SQL DB**

To create in azure : new SQL db , we set the server (unique name in azure nameX.database.windows.com) , set options : use elastic pool , and Pricing tier : in the end 2 resources are created : DB and Server
Pricing tiers : Basic (less demanding worklouad) , Standard (for most production workload) , Premium , PremiumRS
Server admin : can access to all db in the server, create users , create drop objects , set rules in the firewall. Sys role not exists , we have DBA and login roles.
We can set the admin passpowrd from the server blade.
To connect to the DB from SQL SSMS : first from the server we need to allow our ip address in the firewall , then we grab the FQDN (nameX.database.windows.com) we use it as server from our client SSMS with the admin password created.
There is also DB firewall level, we can limit access to a specific db in the server, we can do that using T-SQL : EXECUTE sp_set_database_firewall_rule nameOfRuleHEre , "ip_start" , "ip_end"
SQL server can contain many DBs, Pricing tier is related to the DB not the server, so we can have diffrent pricing tier in same server.
SQL **Elastic** pool : use elastic DTUs (eDTU) for cost effectivness , the DTUs are increasing wit the workload increases, where many DBs share same DTUs (same pool) for DBs in the same server.

Data at **REST** : **Transparent data encryption TDE** : by default is ON , encrypt our db , whenever an app is pulling data azure deycrpt it first then send it.
Data in **Transit and REST** : **enable always encrypted** : in the SSMS right click on the DB , choose **Encrypt columns ,** we select the column (sensitive data), select or create encryption key, select where to store the Master key (azure key vault or local computer). After ecryption is done, select * from will return ecrypted values. To see plain text we need to : disconnect, then to connect using the param : Column Encryption Settting = Enabled , this allow MSMS to use the master key in windows certificate folder then decypt the values
Dynamic Data Masking : to hide some part of the values of the columns like salary to set to 0 or email show it as xxx@xx.com for some users (exluding Admins) . Steps : in the DB in the portal , choose from the menu Dynamic data masking , ADD MASK to the column in qustion.
Alerts : receive alerts based on CPU DTU , blocked by firewall , failed connections .. We set treshhold, email receipants.
Migrating SQL server to Azure SQL DB (to **PaaS**) : the tool DMA (data migration assistant) to be installed in any client of the db : to check compatibility and recommendations, then we need to Create BACPAC File : containes all data and metadata of the DB, will be used to recreate the Db in azure , we run the tool **SQLPackage**.exe /Action:Export (this will generate the BACPAC file) , then we need to import it after creating a server in Azure : **SQLPackage**.exe /Action:Import
Other way to migrate data : SQLCMD (to coonect to server and create emtpy DB with tables strcucture) & BCP utility bulk (export to flat file, and import) --> no stored proc or relatioship is created.

Migration from on-prem SQL server to VM sql server (**IaaS**) : there are 4 options : 1-copy data using bcp utility, 2-backup db (.bak) then restore it . 3-detatach and attach the .mdf .ldf files to the target.

4-for big heavy DBs: lift and shift : export db to a drive and ship it to MS using Windows Import/Export service

SQL db **auto back**up : azure sql db have Point in Time automatic backups (run full:weekly, incremental : 1hr, log:5-10 mnts) , for max of 35 days retention for premium tier. (basic plan has only 7 days see creen below, and 12 days for standard) --> to **restore** : in the DB top menu click restore, we set name of db, we choose the tab Point-In-Time (long term is for below) , we set restore point (date and time) done.

SQL db **long term** backup : allows to backup weekly up to 10 years retention, steps : in the SQL server blade, we choose long-term backup, then we select the DB we click configure from the top, we select (must be pre-created services recovery vault) , we set a policy name if we dont already have, we choose retention period from 1 week to 10 years . **Restore** -> same as above then we choose the azure vault backups and restore point, done.

Capability	Basic tier	Standard tier	Premium tier
Point In Time Restore	Any restore point within 7 days	Any restore point within 14 days	Any restore point within 35 days
Geo-Restore	ERT < 12h, RPO < 1h	ERT < 12h, RPO < 1h	ERT < 12h, RPO < 1h
Active Geo-Replication	ERT < 30s, RPO < 5s	ERT < 30s, RPO < 5s	ERT < 30s, RPO < 5s

-**Policy vs initiative**: 2nd is set of 1st
Policy can be as result "Deny" or "Audit" , or "Append" (like to apply tags if doesnt mean the requirements)
Blueprint : add RBAC users/roles and resources to be created, we assign it to a scope (manag group or resource group) then it will run (user will be assigned to RBAC , VM will be created ..)
After creating the blueprint, we publish it , and we assign it to a subscsription. Blue print is a combination of : RBAC assignement, Policy, ARM tempplate

What's the difference between Azure Blueprint and Azure Policy?

Have you ever considered Azure Policy vs ARM? An azure policy is an access system that provided default allow or deny on new or existing resources to which the policy applied. But azure blueprint is a backage to create govern the implementation of Azure services, security and design.

From <https://www.cybrary.it/blog/0p3n/azure-blueprint-service-new-way-automate-subcription/>

--> web containeser (ACS vs AKS), service fabric

Docker concept : **Dockerfile** is used to copy the code to an image (also to specify the runtime) (typical cmds : push [image] , pull), Docker-**compose.yml** is to be configured which discribed how the app to be run in the container, we can run and stop the container from here also.
In docker-cmopose we specify the poert to acces to the app , 8084:80 : meeans use port 80 to access to the app from the container , and 8084 from local machine
The container can use difeerent OS than our dev env .

Kubernates:
3 nodes and 1 master node
Azure container registry :

AKS :
To run your applications and supporting services, you need a Kubernetes node. An AKS cluster has one or more nodes, which is an Azure virtual machine (VM) that runs the Kubernetes node components and container runtime

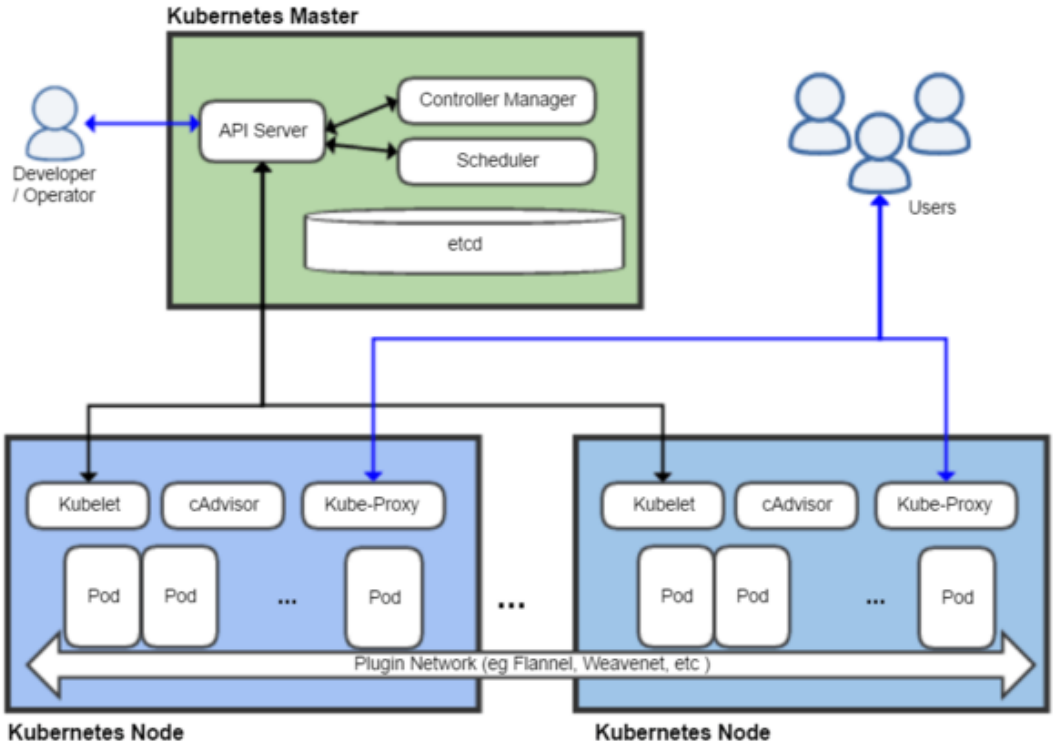
For high workload ,we either choose a big size of VM (CPU and memory) or we can scale up the number of the **nodes** in our AKS **cluster**.
In AKS, the VM image for the nodes in your cluster is currently based on Ubuntu Linux or Windows Server 2019.
Kubernetes uses **pods** to run an instance of your application. A pod represents a single instance of your application. Pods typically have a 1:1 mapping with a container
A **pod** is a logical resource, but the container(s) are where the application workloads run
A *Pod* is the basic execution unit of a Kubernetes application–the smallest and simplest unit in the Kubernetes object model that you create or deploy. A Pod represents processes running on your Cluster.

From <https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/>

Docker is the most common container runtime used in a Kubernetes Pod, but Pods support other container runtimes as well:
Docker
CRI-O
Containerd
Other CRI runtimes: frakti

Docker vs Kubernetes : Docker is the container runtime (which allow us to run the image in the container) , Kuber. Is the Orchestrator.
Kubernetes components into three main parts.

The Control Plane - The **Master**, items : scheduler , API server (for communication between the elements) , controller manager
Nodes - Where pods get scheduled. (and where actually containers get deployed to run , = the physical infra - VMS)
Pods - Holds containers. Here we define how much CPU/memory for our resources , Node can contain multiple Pods. **Pods like processes**



When a Pod gets created (directly by you, or indirectly by a Controller), it is scheduled to run on a Node in your cluster. The Pod remains on that Node until the process is terminated, the pod object is deleted, the Pod is evicted for lack of resources, or the Node fails.

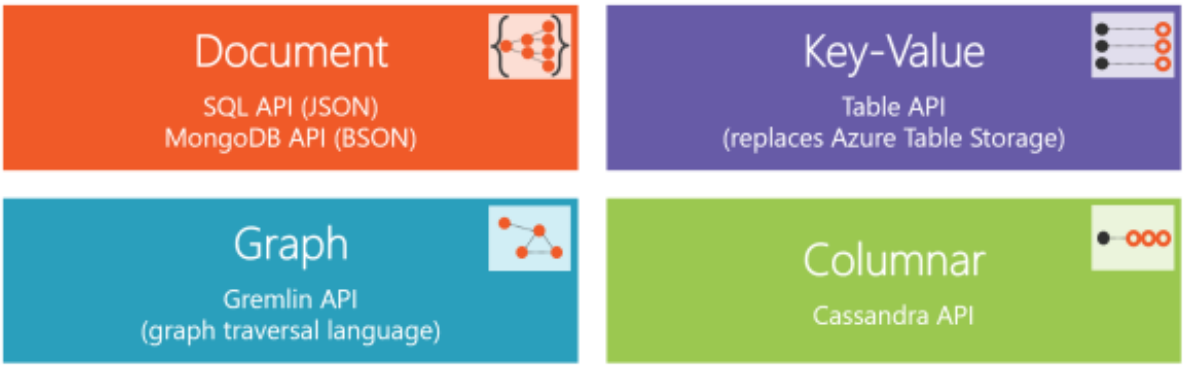
Service fabric :

For microservices , containers either Win or Linux , we can choose docker as container, by default we have a cluster of 5 nodes, when we deploy an app , service fabric deploys it in one of the nodes, ifthe node in question is down , Service F migrated it to another node seamlessly.
To create in portal azure : follow the wizard to specify nbr of nodes , nbr of instances, OS , service vault that will contain a certificate..
After creation these are the resources created : load balancer , public ip address , key vault , service fabric , VM scale set , Vnet , storage account
If we click on the service fabric resource , it represents the cluster , here we can see nbr of nodes and their status (up , disabled ..) and we can see the applications deployed (0 in the beginning)
To scale up the nbr of nodes of the azure SF : In the VM scale set , scaling ,set to x
In VS code : we create new service fabric , in files explorer it's service fabric + 1 service , we can add other service to the SF

--> **cosmos db**

Distributed database over many regions, low latency .

To create : we set our instance (DNS) Xdocument..azure.com , we choose the API (it's the data model) : Core (SQL) , MongoDB, Cassandra, Azure Table, Gremlin(Graph)
Table API : compatible with storage table , no code changes is needed to migrate to it.
SQP API : we create the cosmot DB with this API , then we create a database (we set an id [can be text]) , under the DB we create a collection: we specify a partition key (how we wanna our collection to be partitionned (filtred or organized, its like the DB scheme, it should not be UNIQUE value in the dataaset)) , we expand the collection created and we create an item from the top menu (we put te content as json format)



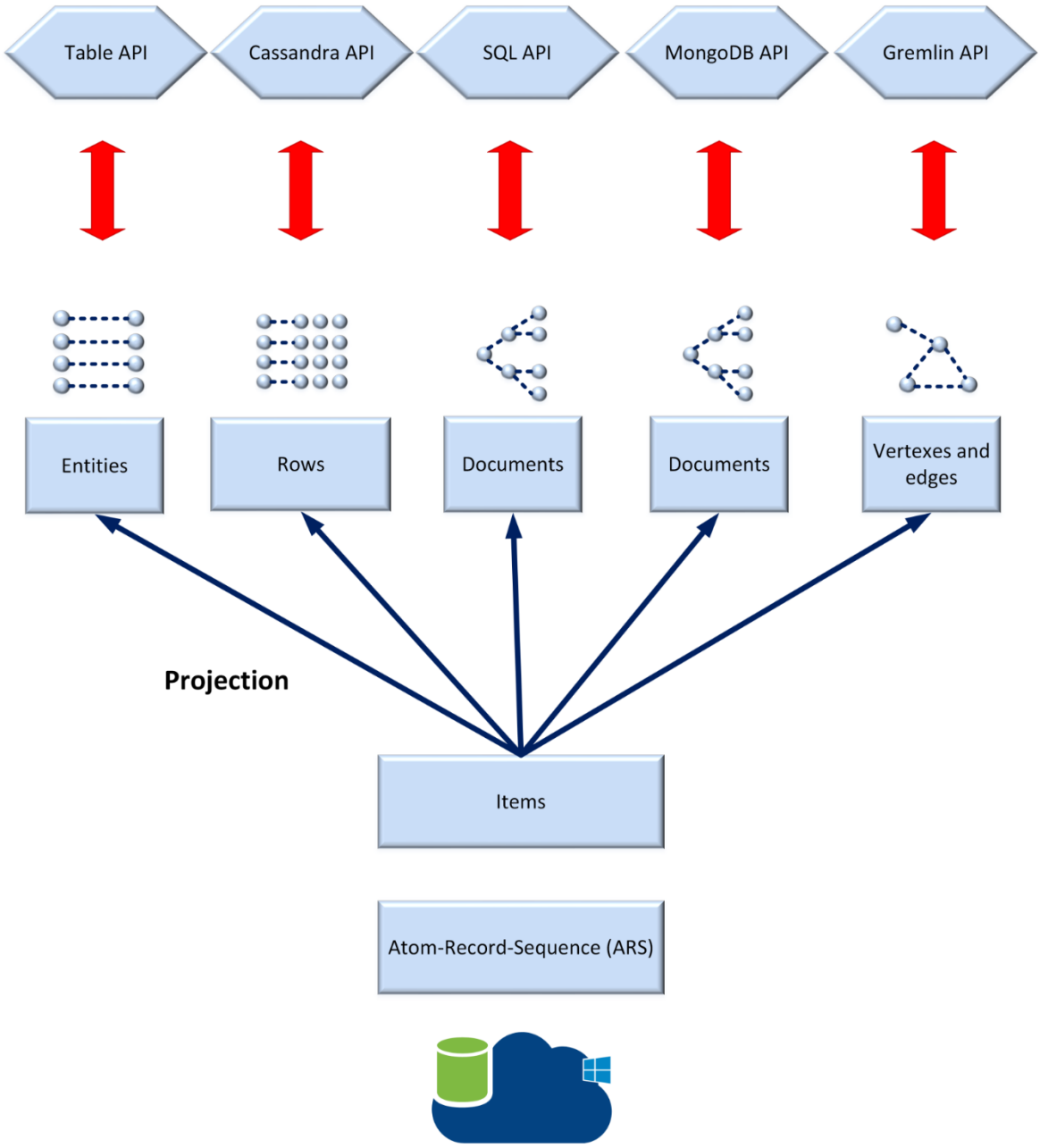
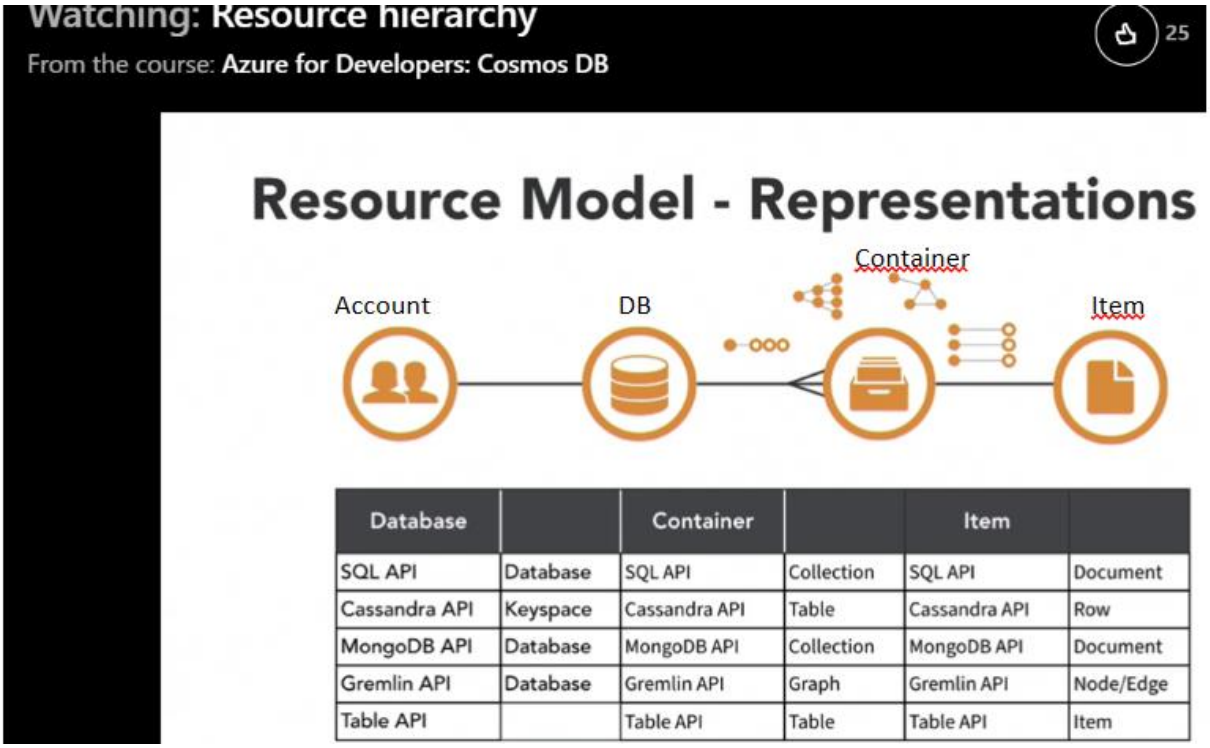


Table API
Cassandra API
SQL API
MongoDB API
Gremlin API
Entities
Rows
Documents
Documents
Vertexes and edges
Projection
Items
Atom-Record-Sequence (ARS)



Database		Container		Item	
SQL API	Database	SQL API	Collection	SQL API	Document
Cassandra API	Keyspace	Cassandra API	Table	Cassandra API	Row
MongoDB API	Database	MongoDB API	Collection	MongoDB API	Document
Gremlin API	Database	Gremlin API	Graph	Gremlin API	Node/Edge
Table API		Table API	Table	Table API	Item

Replicate data globalyly : by default : Single Write, Multi-Read, cosnos DB supports also multi-write multi-read . this is in the left menu of our cosmos DB accoumt, we can select additional regions where to replicat our DB to improve perfs . Multi regsions write is disabled, we can enable it from the right side in same page.
All documents are indexed by default.
Secrutiy : we can use key to access to our cosmos DB : 2 read-Write keys and 2 Read-only key
Consistency levels : describe how data is sync across different replicates in other regions . There are 5 : Strong : we will have same copy of data everywhere but operatoins are slow , eventual : operations are fast, data is immediat. Other lvels : Bounded Staleness, Sessions, Consistent Prefix. By default the Session is set , to change : left menu "Default Consistency"
We can scale up/down our container/database throughput (Rus [Request units per second] values)

--> **event Grid**
Event grid are events oriented where we have event and handlers
Service bus is more about Messages (transactions , financial ..) not events
Event-hub is for big data, usualy IOT devices

Event grid concepts : event : (smtng happened , max size of 64 KB), publisher (user of app who sends the event) , event source, Topics , event subscription, event handler (like http webhook).
Retry mechanism : the event grid will rety in 10 s then 30 s then 1 mnt then 5 until every 1h for 24h . Thisis default , we can curomise this , we can also set a dead letter : sent the event to storage account if the rety fails
Ex : storage acc as publisher , subscribor : azure function , topic : storage accounts
We create a function where the trigger is event grid , in the eventgrid we specify topic as storage account , we we choose our storage account in question (this is the topic) , the endpoint (handler) is selected (our azure function url)

Criteria	Storage queues	Service Bus queues
Ordering guarantee	**No** For more information, see the first note in the "Additional Information" section.	**Yes - First-In-First-Out (FIFO)** (through the use of messaging sessions)
Delivery guarantee	**At-Least-Once**	**At-Least-Once** (using PeekLock receive mode - this is the default) **At-Most-Once** (using ReceiveAndDelete receive mode) Learn more about various Receive modes
Atomic operation support	**No**	**Yes**
Receive behavior	**Non-blocking** (completes immediately if no new message is found)	**Blocking with/without timeout** (offers long polling, or the "Comet technique") **Non-blocking** (through the use of .NET managed API only)
Push-style API	**No**	**Yes** OnMessage and **OnMessage** sessions .NET API.
Receive mode	**Peek & Lease**	**Peek & Lock** **Receive & Delete**
Exclusive access mode	**Lease-based**	**Lock-based**
Lease/Lock duration	**30 seconds (default)** **7 days (maximum)** (You can renew or release a message lease using the UpdateMessage API.)	**60 seconds (default)** You can renew a message lock using the RenewLock API.
Lease/Lock precision	**Message level** (each message can have a different timeout value, which you can then update as needed while processing the message, by using the UpdateMessage API)	**Queue level** (each queue has a lock precision applied to all of its messages, but you can renew the lock using the RenewLock API.)
Batched receive	**Yes**	**Yes**

-->**ARM template:**
Makes resources deplyomenet easier, we can download it as zip file which contain a .ps1 , .sh and .rb , besides template.json and parametres.json

--> **app service**
Tiers app serice plan : Free, basic , standard , premium v2 (isolated , Vnet)
Cheapest tier to implement auto-scaling is Standard (not available for free/shared and basic)
The D1 (Shared) pricing tier does not support HTTPS.

App Service: Tiers vs. Scaling

Tier	Scaling
Free/Shared	N/A
Basic	Manual scaling up to 3 instances
Standard	Autoscaling up to 10 instances
Premium	Autoscaling up to 20 instances
Isolated	Autoscaling up to 100 instances

Custom domain are allowed from Basic tier
Azure CDN : allow caching of webpage for better perfs
AAD authentication authorization : out of the boksz with ZERO code : from the blade we can choose this option and set it to ON, we choose AAD (they are others like google , fb ..) , we can select express so that the wizard will register the app for us and do everything else.
Bakcup app : we can backup our back into zipe file , the backup includes the code and file config and in-app DBs
Monitor : we can monitor response time , CPU , memory .. Set alerts

Build the container image

From <https://docs.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-prepare-app>

In the Dockerfile
=
```
FROM node:8.9.3-alpine
RUN mkdir -p /usr/src/app
COPY ./app/ /usr/src/app/
WORKDIR /usr/src/app
RUN npm install
CMD node /usr/src/app/index.js
```

Then :
--> docker build ./aci-helloworld -t aci-tutorial-app

To see build image use the cmd :
--> docker images

Run the container locally:
--> docker run -d -p 8080:80 aci-tutorial-app

create Azure container registry

--> az acr create --resource-group myResourceGroup --name <acrName> --sku Basic

Log in to container registry

You must log in to your Azure Container Registry instance before pushing images to it

From <https://docs.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-prepare-acr>

--> az acr login --name mycontainerregistry082

From <https://docs.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-prepare-acr>

Push image to Azure Container Registry

From <https://docs.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-prepare-acr>

--> docker push <acrLoginServer>/aci-tutorial-app:v1

Deploy :

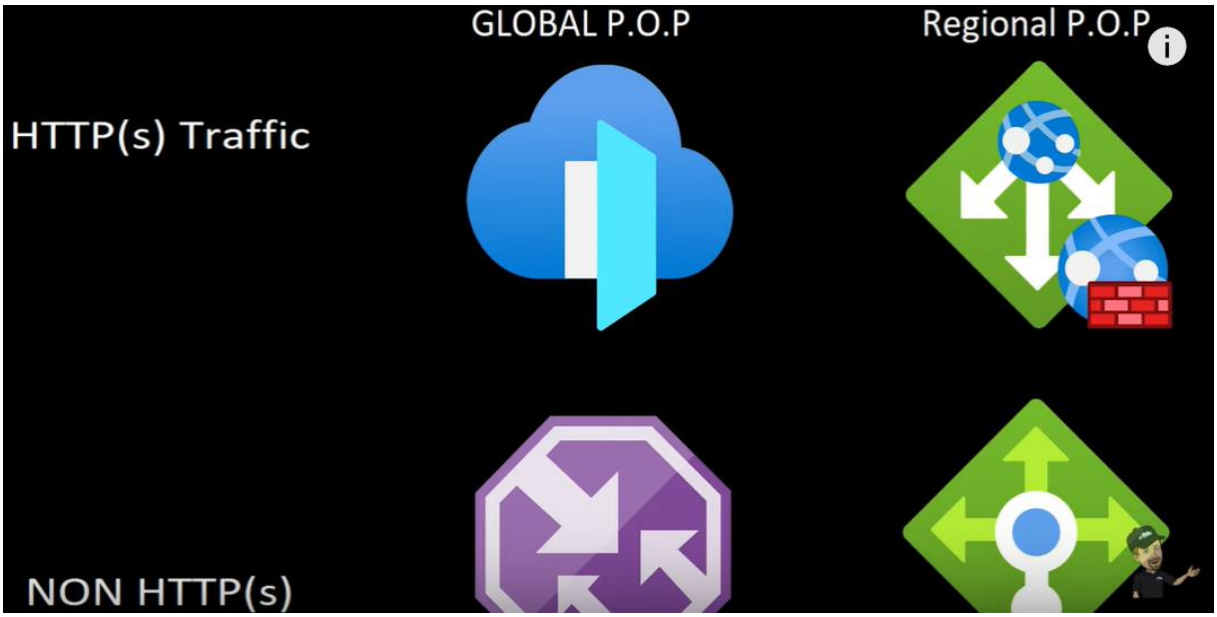
-->
az container create --resource-group myResourceGroup --name aci-tutorial-app --image <acrLoginServer>/aci-tutorial-app:v1 --cpu 1 --memory 1 --registry-login-server <acrLoginServer> --registry-username <service-principal-ID> --registry-password <service-principal-password> --dns-name-label <aciDnsLabel> --ports 8

From <https://docs.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-deploy-app>

Azure Bastion is a new fully platform-managed PaaS service you provision inside your virtual network. It provides secure and seamless RDP/SSH connectivity to your VMs directly in

the **Azure** portal over SSL. When you connect via **Azure Bastion**, your virtual machines do not need a public IP address.

From <https://www.google.com/search?q=azure+bastion&rlz=1C1NDCM_enHU854HU854&oq=azure+Bastion&aqs=chrome.0.0l8.2048j0j7&sourceid=chrome&ie=UTF-8>



RBAC:

Operations are specified with strings that have the following format:

- {Company}.{ProviderName}/{resourceType}/{action}

From <https://docs.microsoft.com/bs-cyrl-ba/azure/role-based-access-control/role-definitions>

Owner: full access to resources, and can delegate control
Contributor : full access but cannot delegate control
Reader : can only view resources in azure (except secrets)
User access Administrator : granted access to manage access to Azure resources
Security Adminsitrator role : access to azure security center , read edit security policies
Network Contributor : Lets you manage networks, but not access to them.

Deny assignement are applied after the Grant assignement !

Routing

User custom routes are prtioritized over system default routes
Longest prefix is prioritized

Source	Address prefixes	Next hop type
Default	Unique to the virtual network	Virtual network
Default	0.0.0.0/0	Internet
Default	10.0.0.0/8	None
Default	192.168.0.0/16	None
Default	100.64.0.0/10	None

Azure create automaticly system routes to each subnets created in the vnet, we cannot create system routes, not remove them . But we can ovvveride them by creating custom rules.
System routes in the table :
-a route to routes traffic to the different address range (so to diferent subnets in the vnet)
-Internet : azure routes any trafffic with the prefix 0.0.0.0/0 to Internet (except if the destination is azure service, where it goes thgough azure backbone)
-None: azure drops the traffic for some standard reserved prefixes.

Optionnal default routes:

- **Virtual network (VNet) peering**: When you create a virtual network peering between two virtual networks, a route is added for each address range within the address space of each virtual network a peering is created for. Learn more about virtual network peering.
- **Virtual network gateway**: One or more routes with *Virtual network gateway* listed as the next hop type are added when a virtual network gateway is added to a virtual network. The source is also *virtual network gateway*, because the gateway adds the routes to the subnet.

From <https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview>

Azure Service **endpoint** :
to limit access to some azure services access to only a specifiy vnet subnet. The connection is through azure backbone. The azure services to be configured to be accesible only by the subnet in question.

+ the traffic is force to be in the backbone, unline some forced-internet calls that might go thgough internet (usually public ip addresses for azure services are using backbone)
When enabled, the source ip of the trrafic is private instead of public, but the target (PaaS, service endpoint) is -public !! Unlike azure private link where the source and target are private.
https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-service-endpoints-overview#secure-azure-services-to-virtual-networks

Private Link vs Service Endpoints

One question that comes to mind when thinking about Private Link is how does this differ from Service Endpoints?
Service endpoints provide a way to lock down access to PaaS resources to a virtual network; however, you are still accessing a public endpoint; it is just locked down to specific traffic. With service endpoints, you are also only locking down access to a service, not to a specific resource. With Private Link, the private endpoints allow access only to the specific resource.
Because private Link creates an endpoint with a private IP, your traffic is flowing solely inside your virtual network and does not require NSG rules allowing outbound traffic beyond your virtual network, unlike service endpoints.

From <https://samcogan.com/wth-is-azure-private-link/>

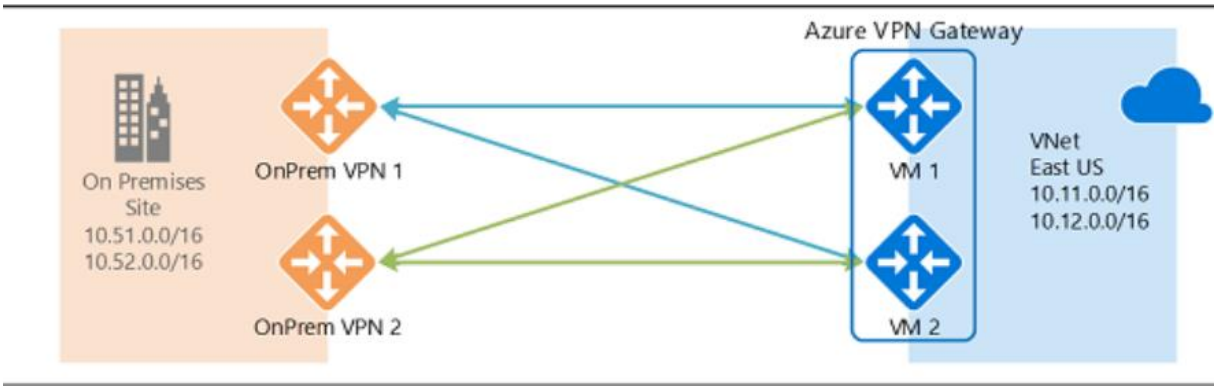
---> where we create a private link, a NIC (network interface card) is created which represents the pruvate of the PaaS service created.

Highly Available Cross-Premises Connectivity

From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-highlyavailable>

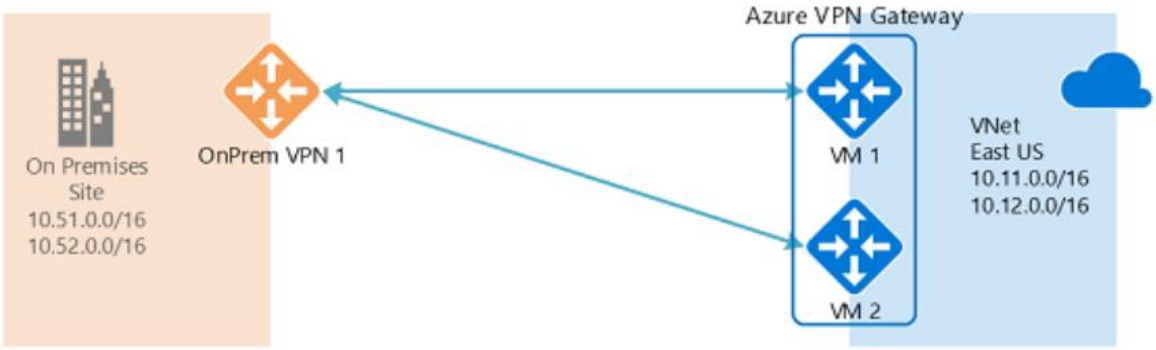
Dual-redundancy: active-active VPN gateways for both Azure and on-premises networks

From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-highlyavailable>



=
This topology will require two local network gateways and two s to support the pair of on-premises VPN devices : 2 public ip on-prem , 2 public ip azure for active-active, 2 connections TO BE CREATED in azure virtual gateway (1 from VNET to on-prem vpn1 and other from VNET to on-premvpn2) , 2 local network gateway

From <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-highlyavailable>



=
Active-active virtual gateway : 2 ip public in azure, 1 ip public on-prem, 1 connection to be created in azure vpn gateway (from vnet to onPrem vpn1) , 1 local gateway to be created (representes our on.prem vpn)

--VNET to VNEt (not peering method) : we create virtual netowrk gateway in each vnet, then in "connections" we link them like S2S vpn : 1 connection in each virtual network gateway.

-- BGP : for routes exchange between on-prem and azure gateway , for VPN its optional, for expressroute its mendatory.

--networks troublshooting : use network watcher (in the overview w emust enable it in the region in question) , we can acces it from azure monitr also. To check connectivity we can use in menu "Connection troublshoot" we set source VM for ex and target ip.
Extention AzureNetworkWatchedExtension has to ben installed in the VM beforehand.

AAD

Entreprise state roaming: only for premium, sync app and win10 setting across devices, to set up : from AAD , menu "Devices" , then menu "Entreprise state roaming" enable
Password **writeback** is premium: allow to write back the new password to on-prem whehenver its changed in azure. To config : in azure connect we add permissions , then we enable "password writbacks" under optional features, then in the portal in AAD , menu "password reset" , menu "on-prem integration" enable "write-back password to your on prem"
Service **service password reset** : in free tiers user can change not reset for cloud users only, basic : we can set for cloud users only , other tiers (p1 p2) can reset. Steps : under AAD menu "password reset" , Enable , then menu "authentication methods" we choose nbr or method and Email/Mobile phone/security questions ..

MFA : available for free tier but to be applied on all users , no options for granular conditions (The mobile authentication app is the only method). Admins have MFA enabled by default for all tiers
ADFS vs Pta : pass through the user name and password entered by user entered at the time stored in cloud whereas ADFs it never leaves on premise. In ADFS user enters password on to ADFs website whereas pass through stores the password in service bus.
AD Connect : to set ip up we need AAD global admin AND AD DS entrepeise Admin

Trusted Ips : we define them in MFA portal (range of IPs) , then the users ocnnecting from this range wont be asked for MFA
Conditionnal access : in AAD (need premium tier) , we create polivy by selecting user/group_users , cloud apps , condition (device plateform , location , exlucde some users or **trusted Ips** ..) , then we select Grant or deny , for grant we can select how to grant (grant with MFA ..)
Conditionnal access --> MFA settings apply last : meaning : if our conditional access policy grant access with MFA to all users except Ip range X , and in the MFA we don't have X as trusted IP , then MFA will be applied.
Access review : AAD p2, permissions : global admin or user admin, steps : from AAD menu "Acces reviews" we add new access review ,we set onetime date of frequency, we set users or group in questions (whom we cant to check access) , we choose the app , select **Reviewers :** group owners, selected users, myself . Then we set upon completion seetings : remove access if reviewser didnt reply , auto apply reviewers reply ..
https://docs.microsoft.com/en-us/azure/active-directory/governance/create-access-review
https://docs.microsoft.com/en-us/azure/active-directory/governance/perform-access-review