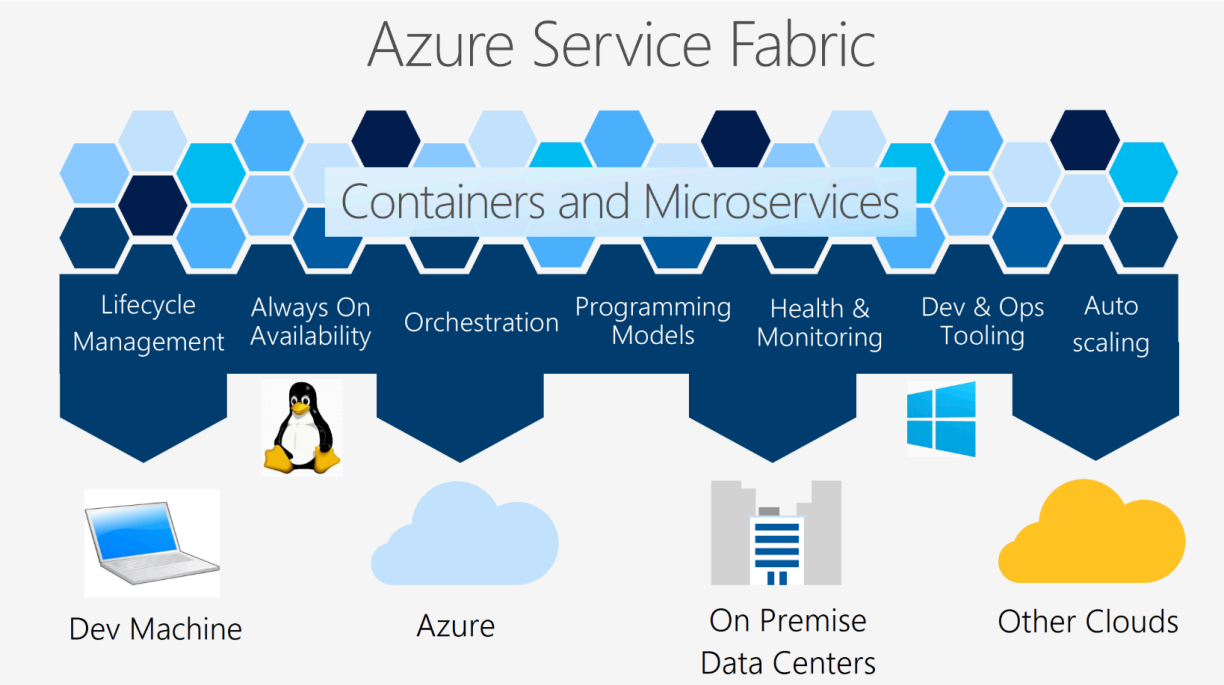
Azure Questions

Service Fabric (is like Kubernetes Service, an orchestration engine) and Service Fabric Mesh or Mesh (is like a Kube-Application-Cluser inside the Kubernetes Service).
Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices and containers. Service Fabric is an open source distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices. Service Fabric is the orchestrator that powers Service Fabric Mesh. Service Fabric provides options for how you can build and run your microservices applications. You can use any framework to write your services and choose where to run the application from multiple environment choices.
Service Fabric Mesh: A fully managed service for running Service Fabric applications in Microsoft Azure. Use Mesh to deploy and manage containerized applications in a secure, scalable environment without having to own or manage any infrastructure. Service Fabric Mesh, a managed Kuberntes clusters by providing fully managed containter orchestration solution without the need to manage the underlying Virtual Machines.
Service Fabric Mesh provides Docker Container Engine while Service Fabric provides Service Fabric Runtime.

Azure Service Fabric is a Platform-as-a-Service, a distributed systems platform that, now open source, provides access to the same tools that Microsoft itself use every day to build, run, and manage many of their services**. It is built to enable enterprises to create and manage scalable microservices and container-based solutions, as well as to benefit from cloud native applications.**

Service Fabric is a distributed systems platform for packaging, deploying, and managing stateless and stateful distributed applications and containers at large scale. Service Fabric runs on Windows and Linux, on any cloud, any datacenter, across geographic regions, or on your laptop.

Service Fabric runs everywhere. You can create clusters for Service Fabric in many environments, including Azure or on premises, on Windows Server, or on Linux. You can even create clusters on other public clouds. In addition, the development environment in the SDK is **identical** to the production environment, with no emulators involved. In other words, what runs on your local development cluster deploys to the clusters in other environments.



In Azure SQL Database, you can configure a single or a pooled database with a long-term backup retention policy (LTR) to automatically retain the database backups in separate Azure Blob storage containers for up to 10 years. You can then recover a database using these backups using the Azure portal or PowerShell. https://docs.microsoft.com/en-us/azure/sql-database/sql-database-long-term-backup-retention-configure

Auto-failover groups is a SQL Database feature that allows you to manage replication and failover of a group of databases on a SQL Database server or all databases in a managed instance to another region. It is a declarative abstraction on top of the existing active geo-replication feature, designed to simplify deployment and management of geo-replicated databases at scale. You can initiate failover manually or you can delegate it to the SQL Database service based on a user-defined policy. The latter option allows you to automatically recover multiple related databases in a secondary region after a catastrophic failure or other unplanned event that results in full or partial loss of the SQL Database service's availability in the primary region. A failover group can include one or multiple databases, typically used by the same application. Additionally, you can use the readable secondary databases to offload read-only query workloads. Because auto-failover groups involve multiple databases, these databases must be configured on the primary server. Auto-failover groups support replication of all databases in the group to only one secondary server in a different region.

When you are using auto-failover groups with automatic failover policy, any outage that impacts one or several of the databases in the group results in automatic failover. Typically these are incidents that cannot be self-mitigated by the built-in automatic high availability operations. The examples of failover triggers include an incident caused by a SQL tenant ring or control ring being down due to an OS kernel memory leak on several compute nodes, or an incident caused by one or more tenant rings being down because a wrong network cable was cut during routine hardware decommissioning.

High Availability and Azure SQL DB: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-high-availability
The high availability solution is designed to ensure that committed data is never lost due to failures, that maintenance operations do not affect your workload, and that the database will not be a single point of failure in your software architecture. There are no maintenance windows or downtimes that should require you to stop the workload while the database is upgraded or maintained.
There are two high-availability architectural models that are used in Azure SQL Database:

- Standard availability model that is based on a separation of compute and storage. It relies on high availability and reliability of the remote storage tier. This architecture targets budget-oriented business applications that can tolerate some performance degradation during maintenance activities.
- Premium availability model that is based on a cluster of database engine processes. It relies on the fact that there is always a quorum of available database engine nodes. This architecture targets mission critical applications with high IO performance, high transaction rate and guarantees minimal performance impact to your workload during maintenance activities.

Azure SQL Database runs on the latest stable version of SQL Server Database Engine and Windows OS, and most users would not notice that upgrades are performed continuously.

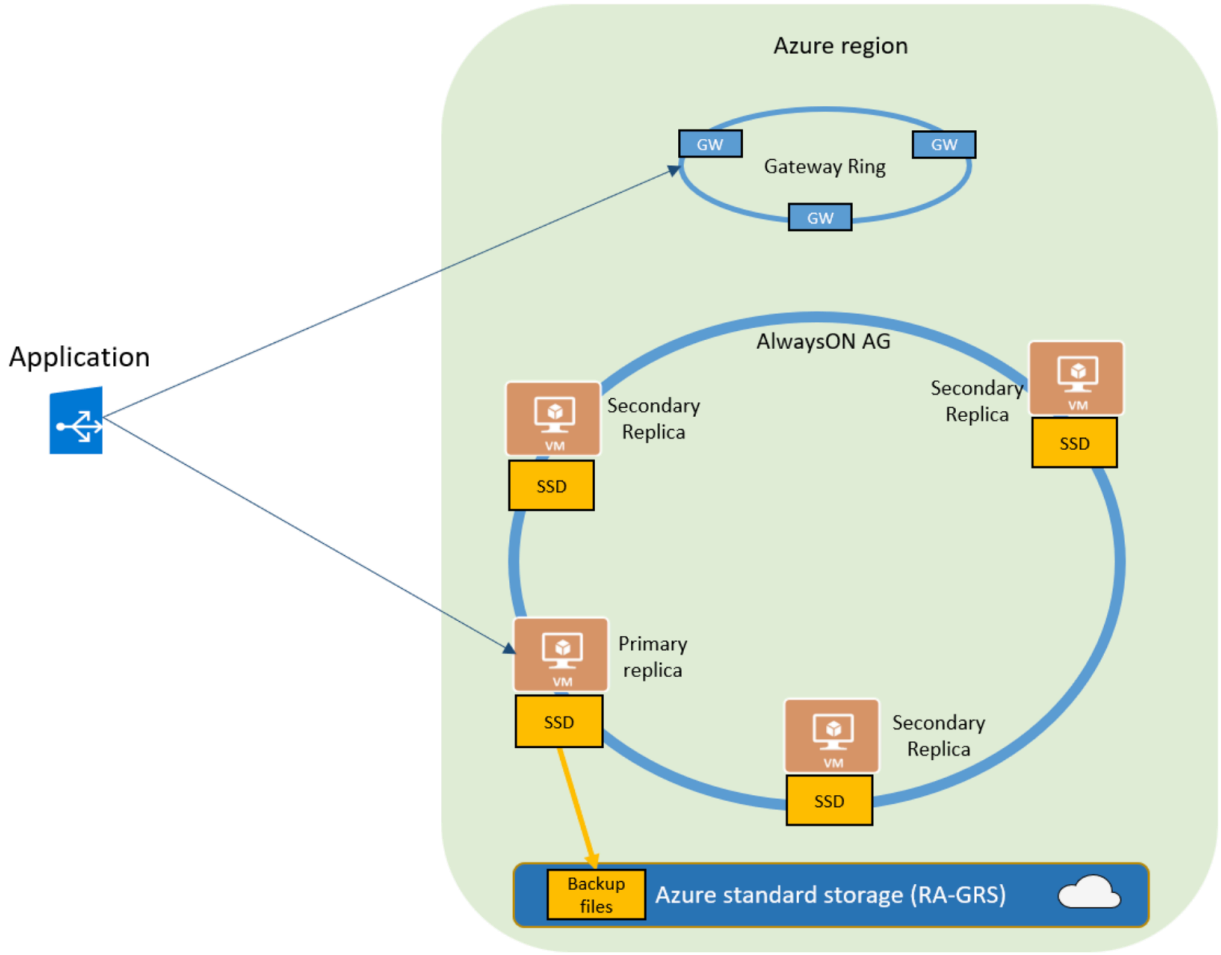
The standard availability model includes two layers:

- A stateless compute layer that runs the sqlservr.exe process and contains only transient and cached data, such as TempDB, model databases on the attached SSD, and plan cache, buffer pool, and columnstore pool in memory. This stateless node is operated by Azure Service Fabric that initializes sqlservr.exe, controls health of the node, and performs failover to another node if necessary.
- A stateful data layer with the database files (.mdf/.ldf) that are stored in Azure Blob storage. Azure blob storage has built-in data availability and redundancy feature. It guarantees that every record in the log file or page in the data file will be preserved even if SQL Server process crashes.

Whenever the database engine or the operating system is upgraded, or a failure is detected, Azure Service Fabric will move the stateless SQL Server process to another stateless compute node with sufficient free capacity. Data in Azure Blob storage is not affected by the move, and the data/log files are attached to the newly initialized SQL Server process. This process guarantees 99.99% availability, but a heavy workload may experience some performance degradation during the transition since the new SQL Server instance starts with cold cache.

Premium and Business Critical service tiers leverage the Premium availability model, which integrates compute resources (SQL Server Database Engine process) and storage (locally attached SSD) on a single node. High availability is achieved by replicating both compute and storage to additional nodes creating a three to four-node cluster.

The underlying database files (.mdf/.ldf) are placed on the attached SSD storage to provide very low latency IO to your workload. High availability is implemented using a technology similar to SQL Server Always On Availability Groups. The cluster includes a single primary replica (SQL Server process) that is accessible for read-write customer workloads, and up to three secondary replicas (compute and storage) containing copies of data. The primary node constantly pushes changes to the secondary nodes in order and ensures that the data is synchronized to at least one secondary replica before committing each transaction. This process guarantees that if the primary node crashes for any reason, there is always a fully synchronized node to fail over to. The failover is initiated by the Azure Service Fabric. Once the secondary replica becomes the new primary node, another secondary replica is created to ensure the cluster has enough nodes (quorum set). Once failover is complete, SQL connections are automatically redirected to the new primary node.



Site Recovery: In **Failover**, select a **Recovery Point** to which to fail over. https://docs.microsoft.com/en-us/azure/site-recovery/site-recovery-failover

- **Latest**: Use the latest point. This processes all the data that's been **sent to Site Recovery service**, and creates a recovery point for each machine. This option provides the lowest RPO (Recovery Point Objective) because the VM created after failover has all the data that's been replicated to Site Recovery when the failover was triggered.
- **Latest processed**: Use this option to fail over VMs to the latest recovery point already **processed by Site Recovery**. You can see the latest processed recovery point in the VM **Latest Recovery Points**. This option provides a low RTO as no time is spent to processing the unprocessed data
- **Latest app-consistent**: Use this option to fail VMs over to the latest application consistent recovery point that's been **processed by Site Recovery**.
- **Latest multi-VM processed**: With this option VMs that are part of a replication group failover to the latest common multi-VM consistent recovery point. Other virtual machines fail over to their latest processed recovery point.

This option is only for recovery plans that have at least one VM with multi-VM consistency enabled.

- **Latest multi-VM app-consistent**: With this option VMs that are part of a replication group fail over to the latest common multi-VM application-consistent recovery point. Other virtual machines failover to their latest application-consistent recovery point. Only for recovery plans that have at least one VM with multi-VM consistency enabled.
- **Custom**: Not available for recovery plans. This option is only for failover of individual VMs.

Azure Batch: https://docs.microsoft.com/en-us/azure/batch/batch-technical-overview
Use Azure Batch to run large-scale parallel and high-performance computing (HPC) batch jobs efficiently in Azure. Azure Batch creates and manages a pool of compute nodes (virtual machines), installs the applications you want to run, and schedules jobs to run on the nodes. There is no cluster or job scheduler software to install, manage, or scale. Instead, you use Batch APIs and tools, command-line scripts, or the Azure portal to configure, manage, and monitor your jobs.

Developers can use Batch as a platform service to build SaaS applications or client apps where large-scale execution is required. For example, build a service with Batch to run a Monte Carlo risk simulation for a financial services company, or a service to process many images.
There is no additional charge for using Batch. You only pay for the underlying resources consumed, such as the virtual machines, storage, and networking.

Batch works well with intrinsically parallel (also known as "embarrassingly parallel") workloads. Intrinsically parallel workloads are those where the applications can run independently, and each instance completes part of the work. When the applications are executing, they might access some common data, but they do not communicate with other instances of the application. Intrinsically parallel workloads can therefore run at a large scale, determined by the amount of compute resources available to run applications simultaneously.

You can also use Batch to run tightly coupled workloads; these are workloads where the applications you run need to communicate with each other, as opposed to run independently. Tightly coupled applications normally use the Message Passing Interface (MPI) API. You can run your tightly coupled workloads with Batch using Microsoft MPI or Intel MPI. Improve application performance with specialized HPC and GPU-optimized VM sizes.

Monitoring and diagnostics

Azure Monitor —> Infrastructure, and Clusters —> Like Azure VMs, Containers, Azure Kubernetes, Service Fabric Clusters.
Application Insights —> Apps & Services —> Like Web Apps, Docker Apps, Functions, Service Fabric Apps.

Monitoring and diagnostics are critical to developing, testing, and deploying workloads in any cloud environment. For example, you can track how your applications are used, the actions taken by the Service Fabric platform, your resource utilization with performance counters, and the overall health of your cluster. You can use this information to diagnose and correct issues, and prevent them from occurring in the future.

Application monitoring
Application monitoring tracks how features and components of your application are being used. Monitor your applications to make sure issues that impact your users are caught. Application monitoring is the responsibility of those developing the application and its services because it is unique to the business logic of your application. It is recommended that you set up application monitoring with Application Insights, Azure's application monitoring tool.

Cluster monitoring
One of Service Fabric's goals is to make applications resilient to hardware failures. This goal is achieved through the platform's system services' ability to detect infrastructure issues and rapidly failover workloads to other nodes in the cluster. But what if the system services themselves have issues? Or if in attempting to deploy or move a workload, rules for the placement of services are violated? Service Fabric provides diagnostics for these, and other issues, to make sure you are informed about how the Service Fabric platform interacts with your applications, services, containers, and nodes.

For Windows clusters, it is recommended that you set up cluster monitoring with Diagnostics Agent and Azure Monitor logs.

For Linux clusters, Azure Monitor logs is also the recommended tool for Azure platform and infrastructure monitoring. Linux platform diagnostics require different configuration as noted in Service Fabric Linux cluster events in Syslog.

Infrastructure monitoring
Azure Monitor logs is recommended for monitoring cluster level events. Once you configure the Log Analytics agent with your workspace as described in previous link, you will be able to collect performance metrics such as CPU Utilization, .NET performance counters such as process level CPU utilization, Service Fabric performance counters such as # of exceptions from a reliable service, and

container metrics such as CPU Utilization. You will need to write container logs to stdout or stderr so that they will be available in Azure Monitor logs.

Watchdogs
Generally, a watchdog is a separate service that watches health and load across services, pings endpoints, and reports unexpected health events in the cluster. This can help prevent errors that may not be detected based only on the performance of a single service. Watchdogs are also a good place to host code that performs remedial action that don't require user interaction such as cleaning up log files in storage at certain time intervals. See a sample watchdog service implementation in Service Fabric Linux cluster events in Syslog.

Enable Application Insights
There are two ways to enable application monitoring for Azure App Services hosted applications:

- **Agent-based application monitoring** (ApplicationInsightsAgent).
  - This method is the easiest to enable, and no advanced configuration is required. It is often referred to as "runtime" monitoring. For Azure App Services we recommend at a minimum enabling this level of monitoring, and then based on your specific scenario you can evaluate whether more advanced monitoring through manual instrumentation is needed.
- **Manually instrumenting the application through code** by installing the Application Insights SDK.
  - This approach is much more customizable, but it requires adding a dependency on the Application Insights SDK NuGet packages. This method, also means you have to manage the updates to the latest version of the packages yourself.
  - If you need to make custom API calls to track events/dependencies not captured by default with agent-based monitoring, you would need to use this method. Check out the API for custom events and metrics article to learn more.

Lifecycle events and performance counters from Docker containers can be charted on Application Insights. Install the Application Insights image in a container in your host, and it will display performance counters for the host, as well as for the other images.
With Docker, you distribute your apps in lightweight containers complete with all dependencies. They'll run on any host machine that runs a Docker Engine.
When you run the Application Insights image on your Docker host, you get these benefits:

- Lifecycle telemetry about all the containers running on the host - start, stop, and so on.

- Performance counters for all the containers. CPU, memory, network usage, and more.
- If you installed Application Insights SDK for Java in the apps running in the containers, all the telemetry of those apps will have additional properties identifying the container and host machine. So for example, if you have instances of an app running in more than one host, you can easily filter your app telemetry by host.

- Backups, Availability, Replication: on Managed Services vs Non-Managed Services. Managed services are easier to configure and administer. You don't need to provision VMs, set up VNets, manage patches and updates, and all of the other overhead associated with running software on a VM.

Instead of running...	**Consider using...**
Active Directory	Azure Active Directory Domain Services
Elasticsearch	Azure Search
Hadoop	HDInsight
IIS	App Service
MongoDB	Cosmos DB
Redis	Azure Cache for Redis
SQL Server	Azure SQL Database

- App Service: Traffic Manager, VNET integration, Local Caching, Deployment Slots, Testing in Production, Backup/Restore, and Auto-scale features only available in Standard, Premium and Isolated. Azure App Service does not support the installation of custom software. App Service minimizes administrative burdens, such as auto-scale, high availability, and support for various development tools and services.
- Just In Time (JIT) access on VNET or on the VM. Azure Privileged Identity Management does not provide time-bound network access. It can provide time-bound privilege elevation. Azure Privileged Identity Management (PIM) is a service that provides a range of features to protect against a range of security concerns for users who are assigned permissions for Azure AD or Azure resources. PIM provides features like just-in-time access, permissions and approval request workflows, and access reviews. Azure Security Center can be used to enable just-in-time (JIT) virtual machine (VM) access. This ensures that network access is only provided as and when required.

Azure Active Directory (AD) Privileged Identity Management (PIM) to manage elevated access for users who have privileged roles for Azure services. Use the Azure Active Directory (Azure AD) Privileged Identity Management (PIM) feature of just-in-time role activation (JIT) to temporarily elevate the role-based access as needed for a defined time. Elevated access includes job roles that need greater access, including support, resource administrators, resource owners, service administrators, and global administrators. We manage role-based access at the resource level. Because elevated access accounts could be misused if they're compromised, we rationalize new requests for elevated access and perform regular re-attestation for elevated roles.

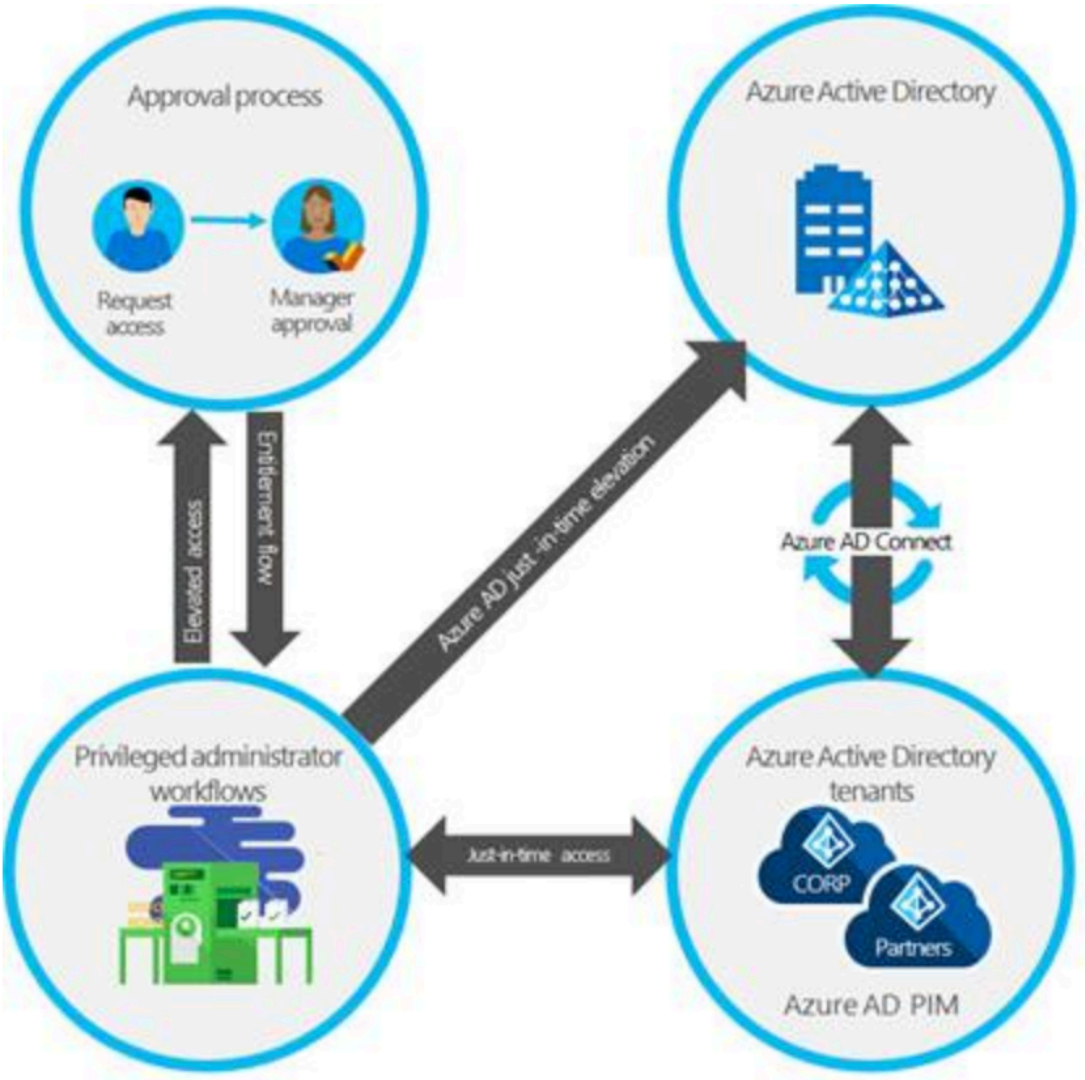


Figure 1. Azure AD PIM elevated access workflow

JIT administrator access

Historically, we could assign an employee to an administrative role through the

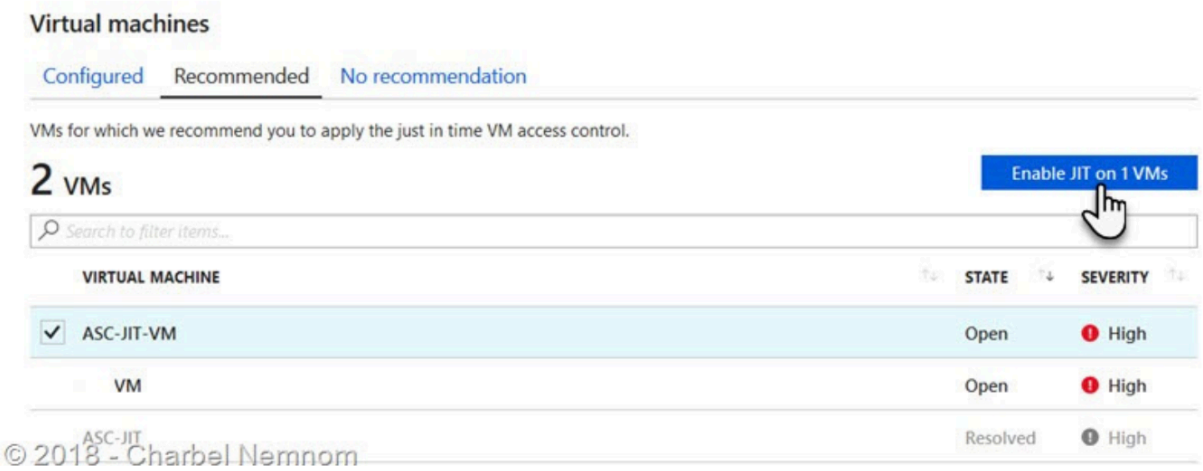
Azure portal or through Windows PowerShell and that employee would be a permanent administrator; their elevated access would remain active in the assigned role.
Azure AD PIM introduced the concept of permanent and eligible administrators in Azure AD and Azure. Permanent administrators have persistent elevated role connections; whereas, eligible administrators have privileged access only when they need it. The eligible administrator role is inactive until the employee needs access, then they complete an activation process and become an active administrator for a set amount of time. We've stopped using permanent administrators for named individual accounts, although we do have some automated service accounts that still use the role.

Role activation in Azure Active Directory
Azure AD PIM uses administrative roles, such as tenant admin and global admin, to manage temporary access to various roles. With Azure AD PIM, you can manage the administrators by adding or removing permanent or eligible administrators to each role. Azure AD PIM includes a number of built-in Azure AD roles as well as Azure that we manage.
To activate a role, an eligible admin will initialize Azure AD PIM in the Azure portal and request a time-limited role activation. The activation is requested using the **Activate my role** option in Azure AD PIM. Users requesting activation must satisfy conditional access policies to ensure that they are coming from authorized devices and locations, and their identities must be verified through multi-factor authentication.

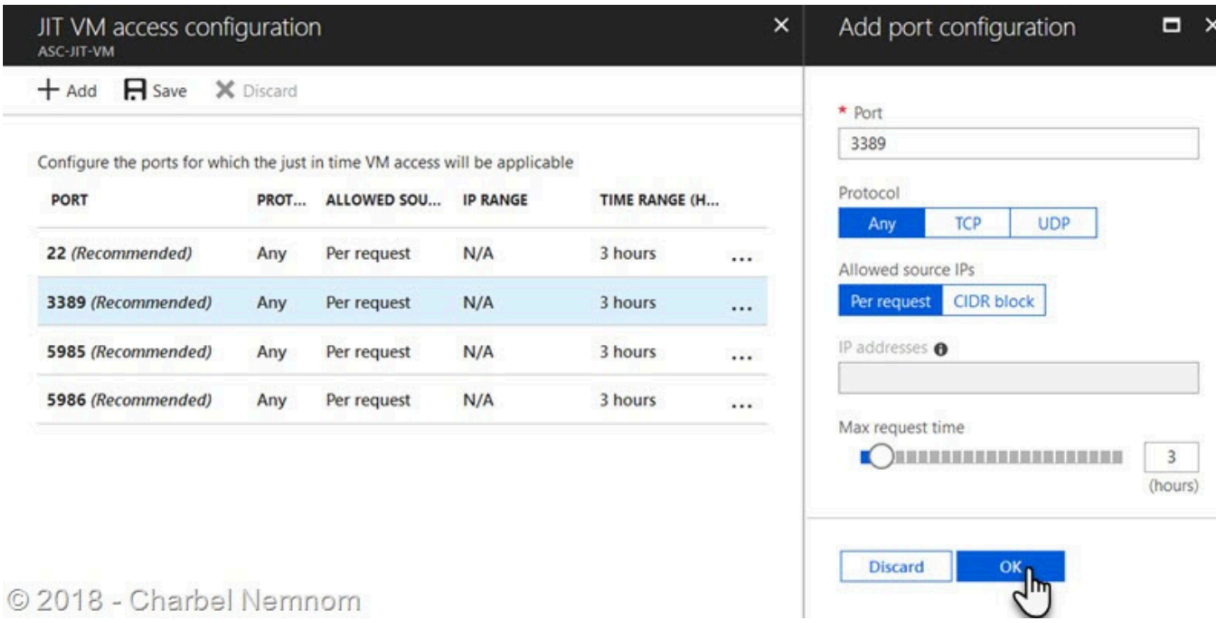
Just-in-Time VM access is one of many features that is included in Azure Security Center which is something you should consider for your virtual machines. You can specify rules for how users can connect to virtual machines. When needed, access can be requested from Azure Security Center or via PowerShell. As long as the request complies with the rules, access is automatically granted for the requested time only.

How Just in Time VM Access Works

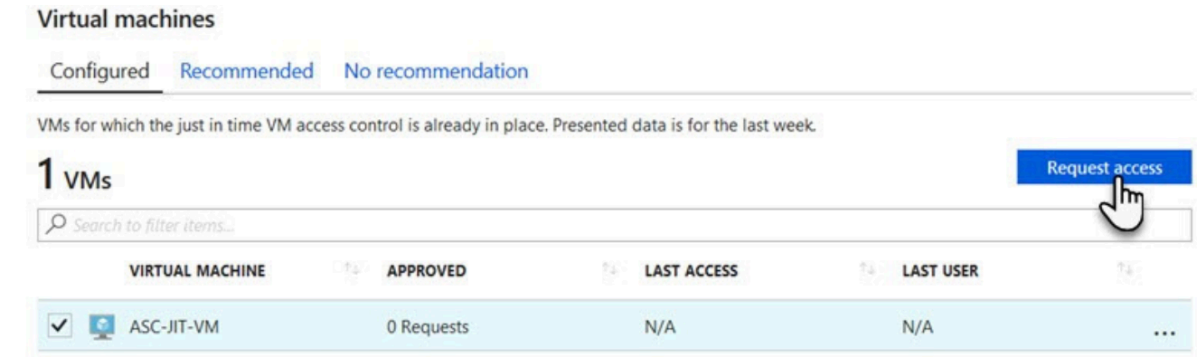
So what the just-in-time VM access feature actually does, is it really automates the Network Security Group (NSG) exception to let me connect in, by default the VM is locked and it's blocking any RDP or SSH remote management, and I cannot connect to it, when it's time for me to do a connection, what actually happens is, I go to Azure Portal, then I go to the Azure Security Center, and I **Enable just-in-time** VM access for that VM.



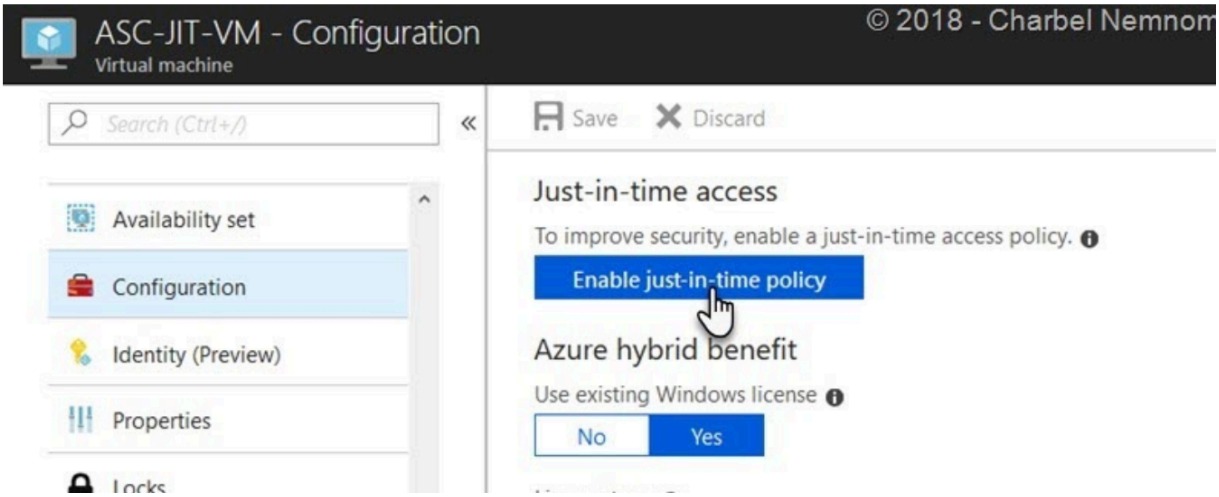
Then I specify for how many hours I want to allow access and click **OK**.



When I need to access my Azure virtual machine, I go to Azure Portal again, then I go to the Azure Security Center, and I **Request access** for that VM. At this point it work out what is my public facing IP address, and it will go and modify the Network Security Group to allow an exception for whichever protocol I'm selecting be an RDP or SSH or WS management, it will add that exception just for my IP address that I want access just for that period of time, when I enable it now I can go and RDP in from the Internet, when that time expires, it will close that exception so I cannot access it anymore.



And Microsoft recently announced a new configuration option for Just-In-Time VM Access from the virtual machine blade directly to make it even easier for you to reduce your exposure to threats.



Just-In-Time access for Azure Firewall

To learn more about Just-In-Time (JIT) VM access, please check the following article. Just like JIT on Network Security Groups (NSG), when using Just-In-Time with Azure Firewall, Azure Security Center allows inbound traffic to your Azure VMs only per confirmed request, by creating an Azure Firewall NAT rule (if needed – in addition to NSG rules). If you are new to Azure Firewall, please check Microsoft documentation here.

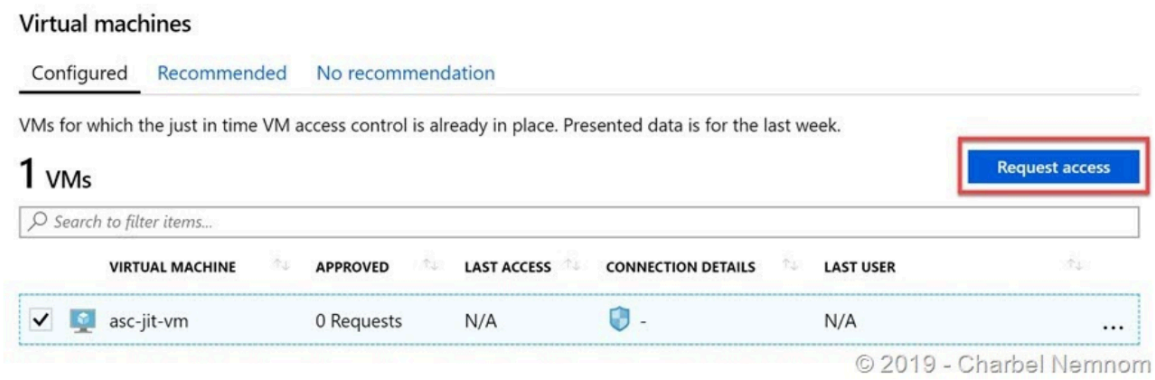
When a user **requests access** to a VM, Azure Security Center checks that the user has **Role-Based Access Control (RBAC)** permissions that permit them to successfully **request access** to a VM. If the request is approved, Azure Security Center automatically configures the Azure Firewall (and NSGs) to allow inbound traffic to the selected ports and requested source IP addresses or ranges, for the amoun of time that was specified. After the time has expired, Azure Security Center restores the firewalls and NSGs to their previous states. However, for the connections that are already established (connected) won't be interrupted. In addition, when requesting access, Azure Security Center provides you with the right connection details to your virtual machine.

Use Just-In-Time access with Azure Firewall

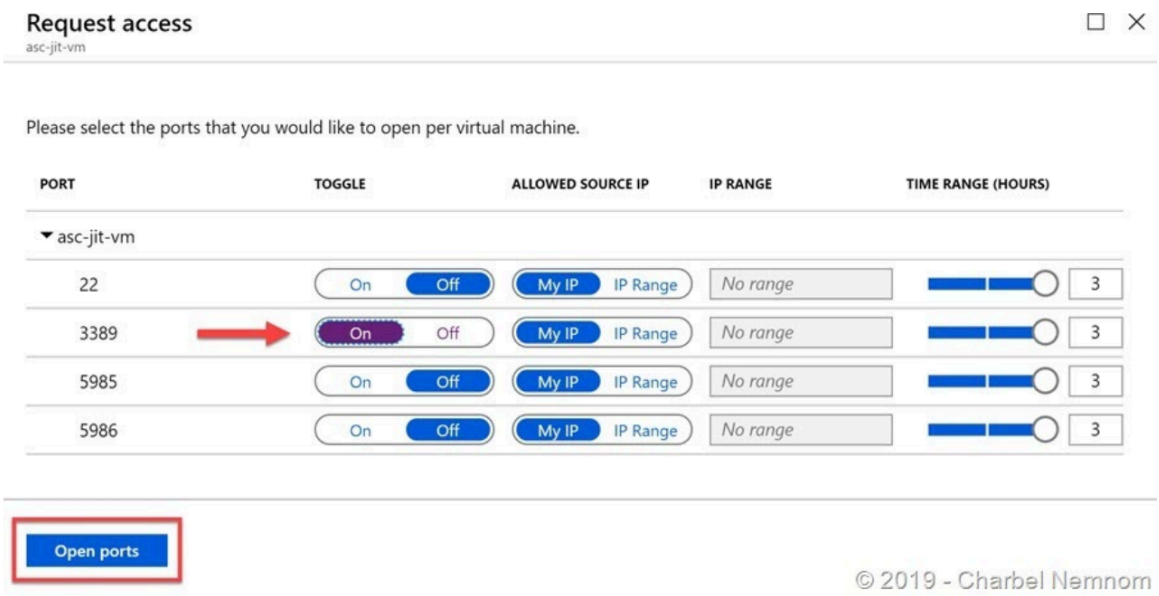
To use Just-In-Time (JIT) VM access with Azure Firewall, you need first to configure and deploy Azure Firewall. Microsoft has a great tutorial on **how to deploy and configure Azure Firewall using the Azure portal**.

Once you have Azure Firewall configured and you **enabled Just-In-Time access for your virtual machine**, then you can take the following easy steps:

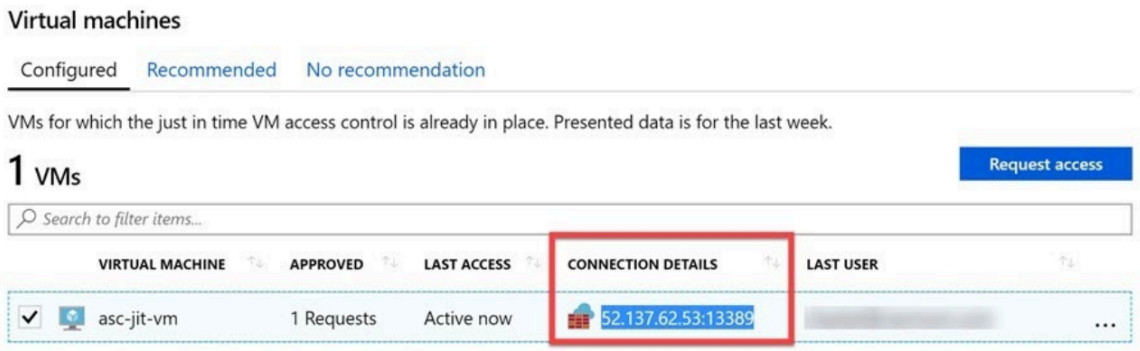
1. Open the Azure Portal, then go to **Security Center**, under **Just in time VM access**, select **Configured**.
2. Under **VMs**, select the VM that you want to request just-in-time access for, and then select **Request access**.



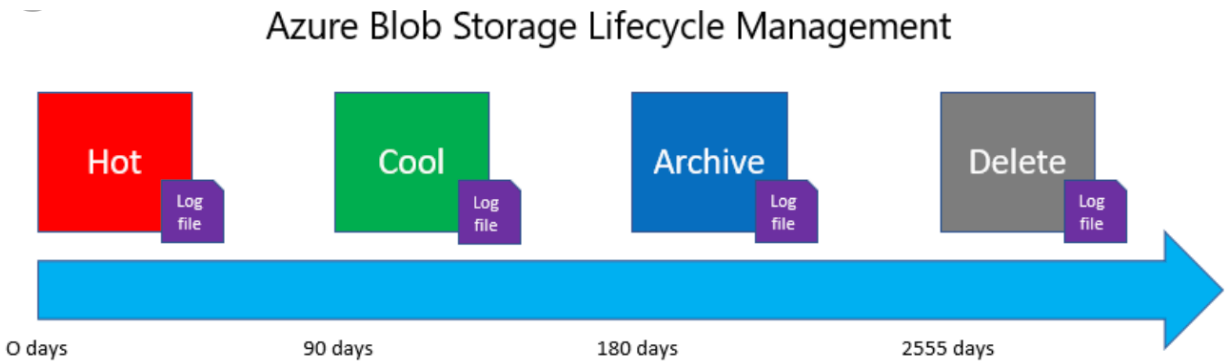
3. Under **Request access**, for each selected VM, configure the ports that you want to open and the source IP addresses that the port is opened on and the time window for which the port will be open. Please note that it is only possible to request access to the ports that are configured in the just-in-time policy. Each port has a maximum allowed time derived from the just-in-time policy. Select **Open ports**.

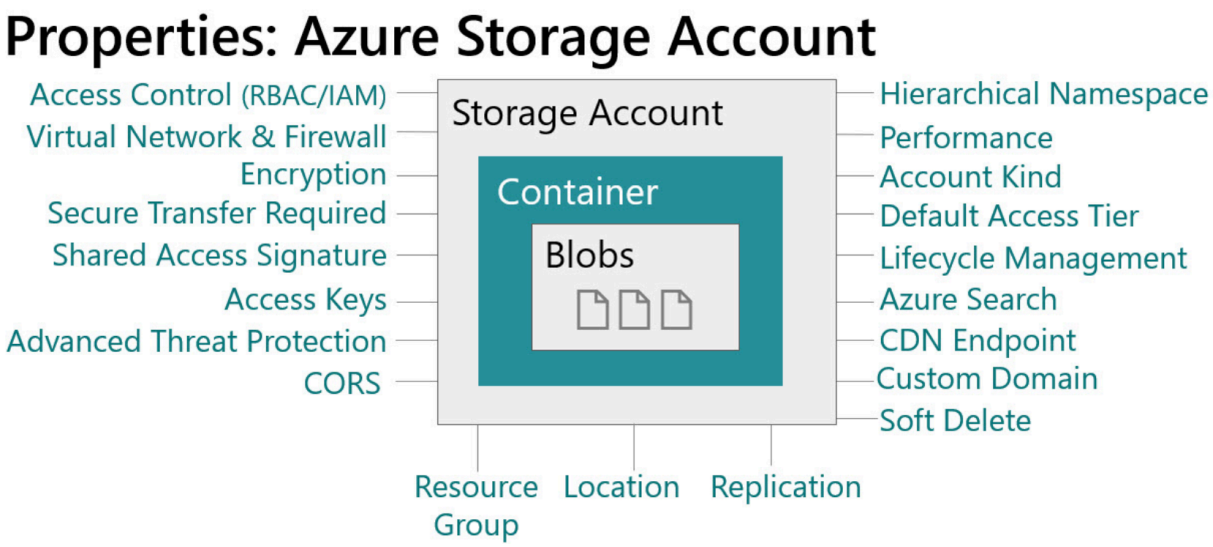


4. The icon in the '**Connection Details**' column indicates whether JIT is enabled on the NSG or FW. If it's enabled on both, only the Firewall icon appears. The '**Connection Details**' column provides the correct information required to connect the VM, as well as indicates the opened ports. In this example, since we are using Azure Firewall with JIT, the Firewall icon only appears.

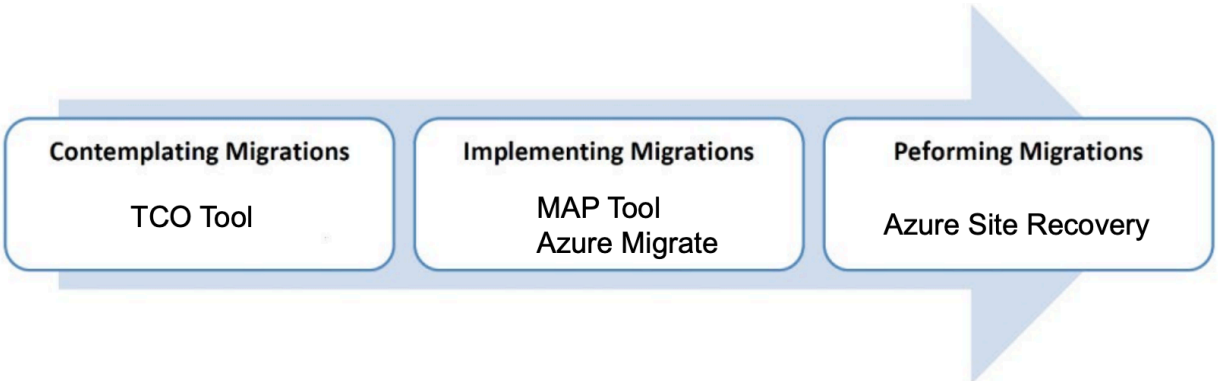


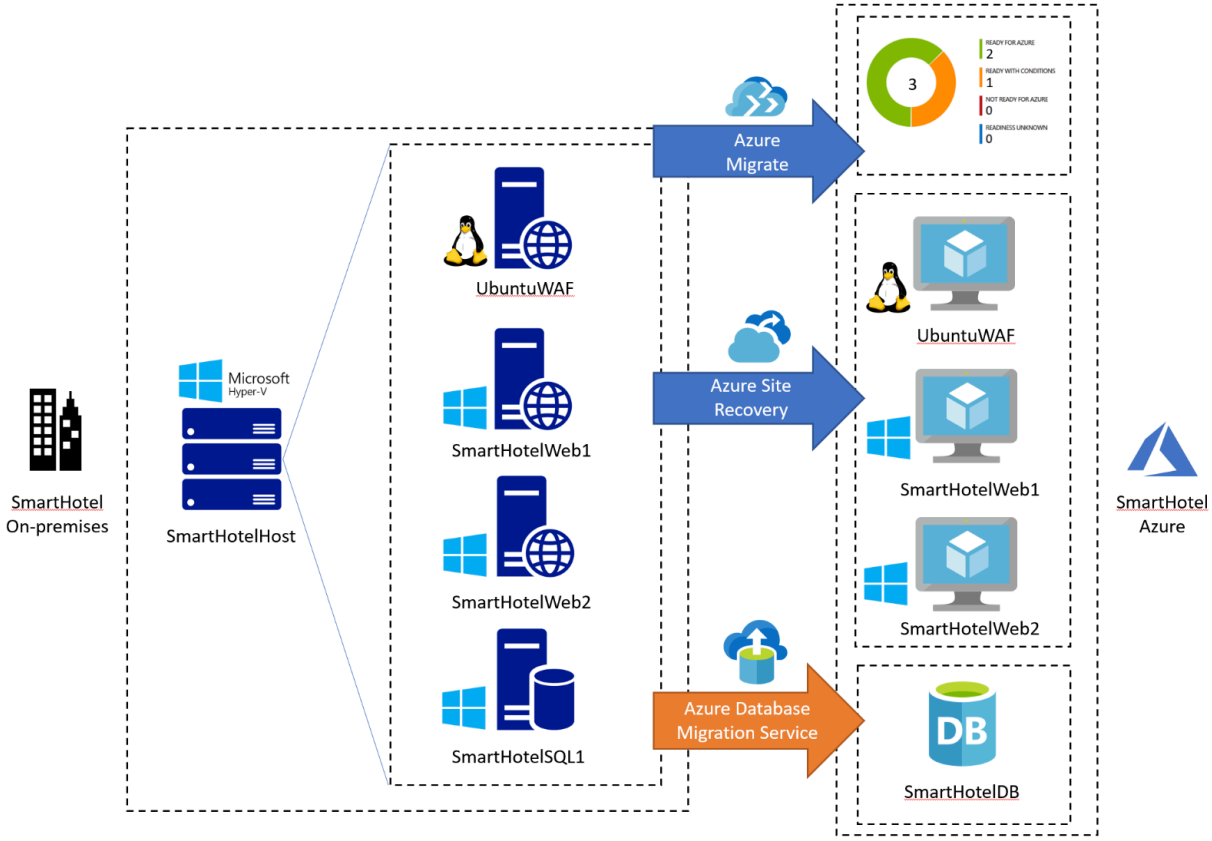
- Storage: Object Life Cycle Management

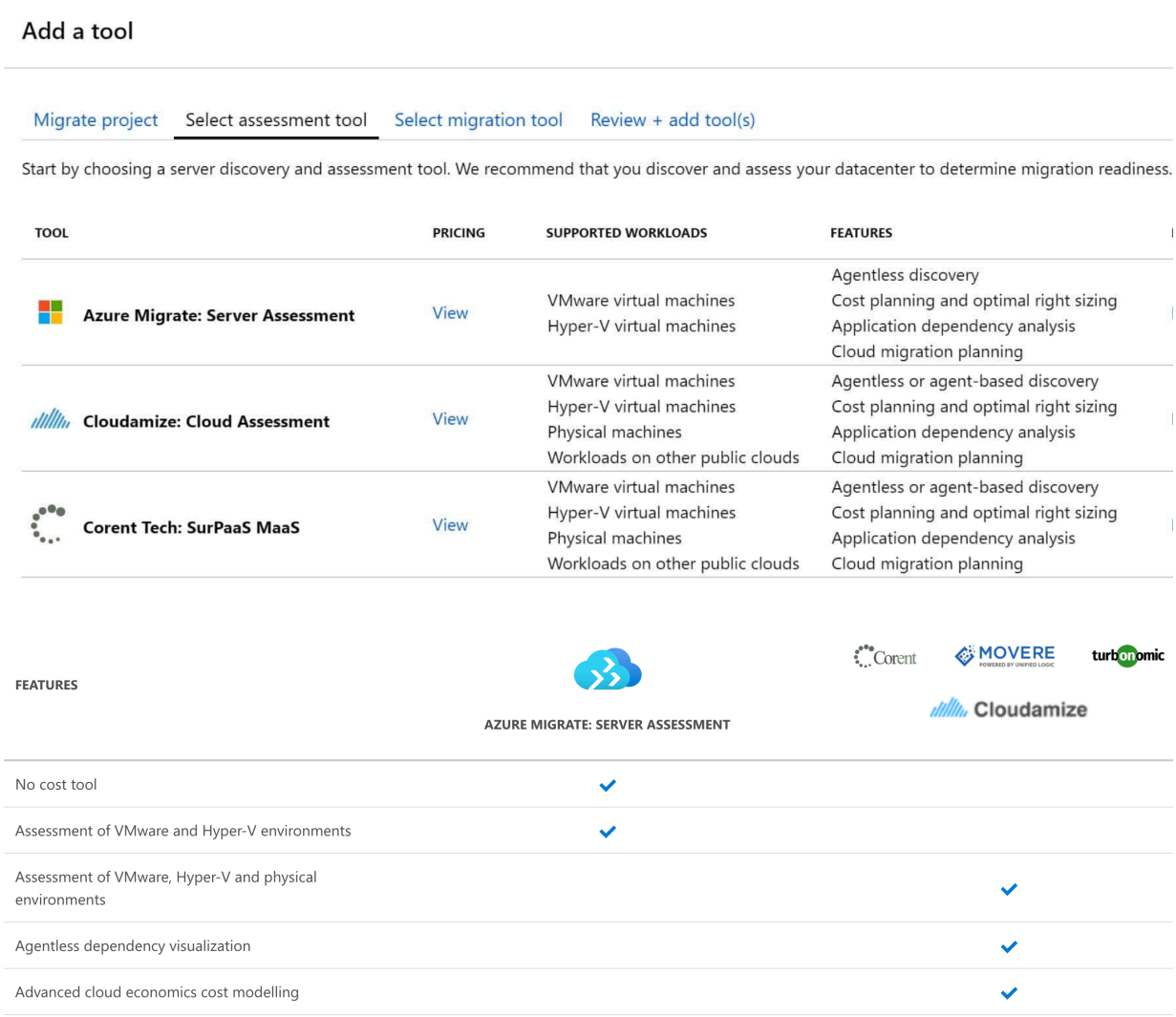




- Pricing tools: Azure Migrate (On-prem and Cloud cost comparison), TCO Calculator, Azure Pricing Calculator, Cost Management. Microsoft Assessment and Planning Toolkit provides performance metrics, Cloud readiness details. It is an executable on the platform. It provides no dependency or cost details. The TCO calculator - web based tool, requires manual entry of information. It does not provide an automated approach to analyzing workloads and estimating costs. The Azure TCO Calculator (total cost of ownership) is a tool which helps to estimate and compare the total cost of operating an on-premises solution, to the total cost of operating that same solution within Azure.
  Azure Migrate provides a range of tools to assess workloads for migration, and perform actual migrations. Azure Migrate - Assess Azure readiness, provide size recommendations, Estimates monthly cost, and provides decency mappings. It is only capable of analyzing VMs, not physical servers. It installs two agents (Monitoring and Dependency Agents) to collect the details. Azure Site Recovery - Used for DR and migration purpose. Replicate Workloads on the cloud, Failover and Failback capabilities, Application consistent snapshots, Non-disruptive testing, Customize recovery plans using runbacks.


Contemplating Migrations
TCO Tool
Implementing Migrations
MAP Tool
Azure Migrate
Peforming Migrations
Azure Site Recovery


SmartHotel
On-premises
Microsoft
Hyper-V
SmartHotelHost
UbuntuWAF
SmartHotelWeb1
SmartHotelWeb2
SmartHotelSQL1
Azure
Migrate
Azure Site
Recovery
Azure Database
Migration Service
3
READY FOR AZURE
2
READY WITH CONDITIONS
1
NOT READY FOR AZURE
0
READINESS UNKNOWN
0
UbuntuWAF
SmartHotelWeb1
SmartHotelWeb2
DB
SmartHotelDB
SmartHotel
Azure



Add a tool

Migrate project | Select assessment tool | Select migration tool | Review + add tool(s)

Start by choosing a server discovery and assessment tool. We recommend that you discover and assess your datacenter to determine migration readiness.

TOOL	PRICING	SUPPORTED WORKLOADS	FEATURES
Azure Migrate: Server Assessment	View	VMware virtual machines Hyper-V virtual machines	Agentless discovery Cost planning and optimal right sizing Application dependency analysis Cloud migration planning
Cloudamize: Cloud Assessment	View	VMware virtual machines Hyper-V virtual machines Physical machines Workloads on other public clouds	Agentless or agent-based discovery Cost planning and optimal right sizing Application dependency analysis Cloud migration planning
Corent Tech: SurPaaS MaaS	View	VMware virtual machines Hyper-V virtual machines Physical machines Workloads on other public clouds	Agentless or agent-based discovery Cost planning and optimal right sizing Application dependency analysis Cloud migration planning

FEATURES	AZURE MIGRATE: SERVER ASSESSMENT	Corent / MOVERE / turbonomic / Cloudamize
No cost tool	✓	
Assessment of VMware and Hyper-V environments	✓	
Assessment of VMware, Hyper-V and physical environments		✓
Agentless dependency visualization		✓
Advanced cloud economics cost modelling		✓

- Site Recovery, Recovery Services Vault. Azure Data Factory. Azure Data Factory is a managed cloud service that's built for these complex hybrid extract-transform-load (ETL), extract-load-transform (ELT), and data integration projects.

Azure Backup

Azure backup prevents data on-premises and into the cloud. It is designed to maximize backup efficiency while minimizing storage consumption. Azure backup is mostly used in cases with accidental deletion, patch testing, alternative location recovery and for security. It is a part of OMS add on as well as sold standalone. It is a flexible long term retention policy and provides granular recovery. Azure backups vary in the acceptable RPO (Recovery Point Objective). VM backups take usually one day while database backups might take 15 days only. It provides exceptional security and reliability, with six copies of encrypted data distributed over two azure datacenters with a 99.9% service availability SLA.

The AzureBackup data is typically retained for 30 days or less. The amount of data that a backup solution needs to process is very high due to a larger RPO (Recovery Point Objective). This leads to a higher RTO (Recovery Time Objective).

AZURE SITE RECOVERY

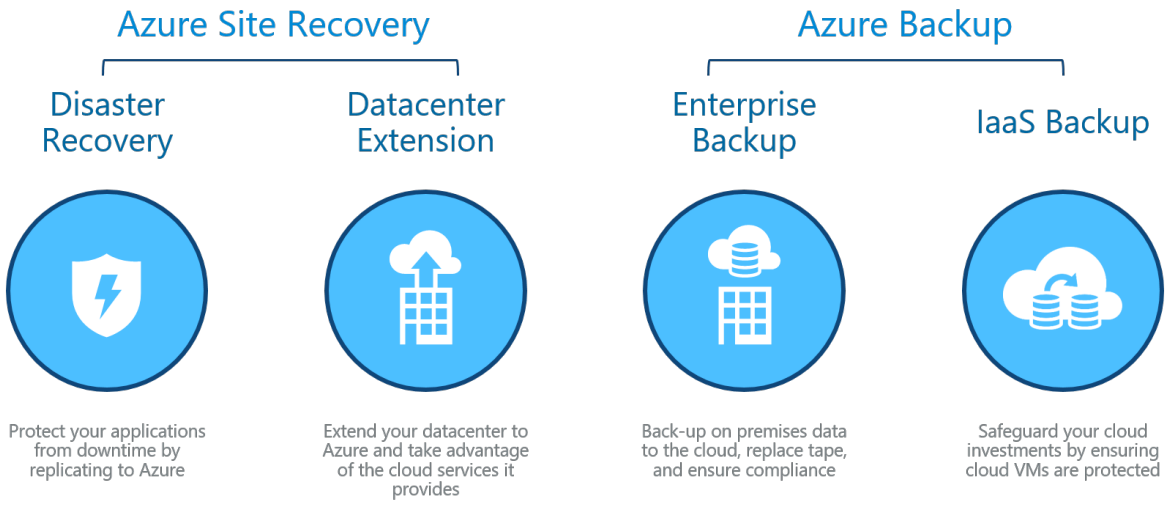
Azure SiteRecovery coordinates replication, failover, and failback. It provides flexible automated protection and recovery for your local servers and workstations.

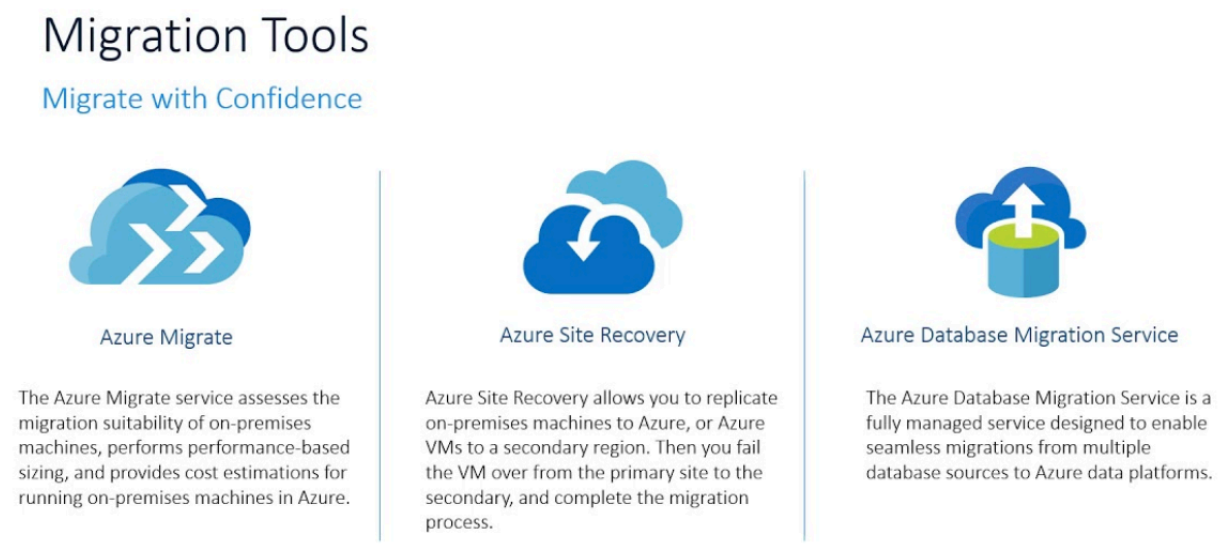
Azure Site Recovery supports multiple scenarios of disaster recovery, migration and Dev/Test environment. It has a very low Recovery point objective and Recovery time Objective. The RPO might be as low as 30 seconds while the RTO might be of 5-30 minutes. The smaller RTO is because it is in sync with the source. ASR needs only operational recovery data. It allows us to make customizable recovery plans and create on-demand test copies.

Conclusion

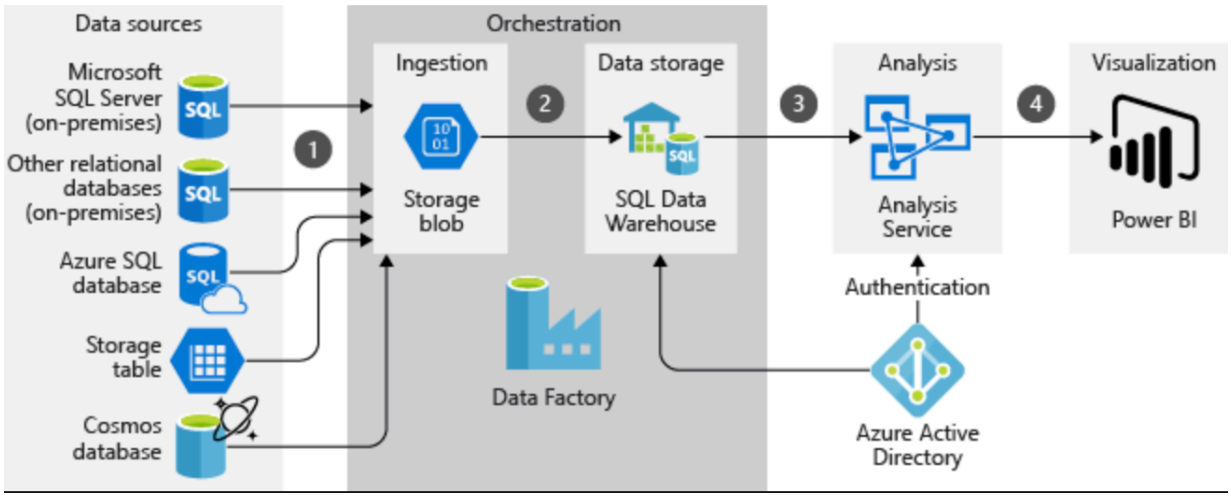
Both Azure Backup and Azure Site Recovery, are effective disaster solutions that back up and restore data. They both work together to ensure complete business continuity, but they have slightly different purposes.

While Azure backup can help you to restore a corrupted file, Azure Site Recovery will replicate the data and configuration of a system to another datacenter.



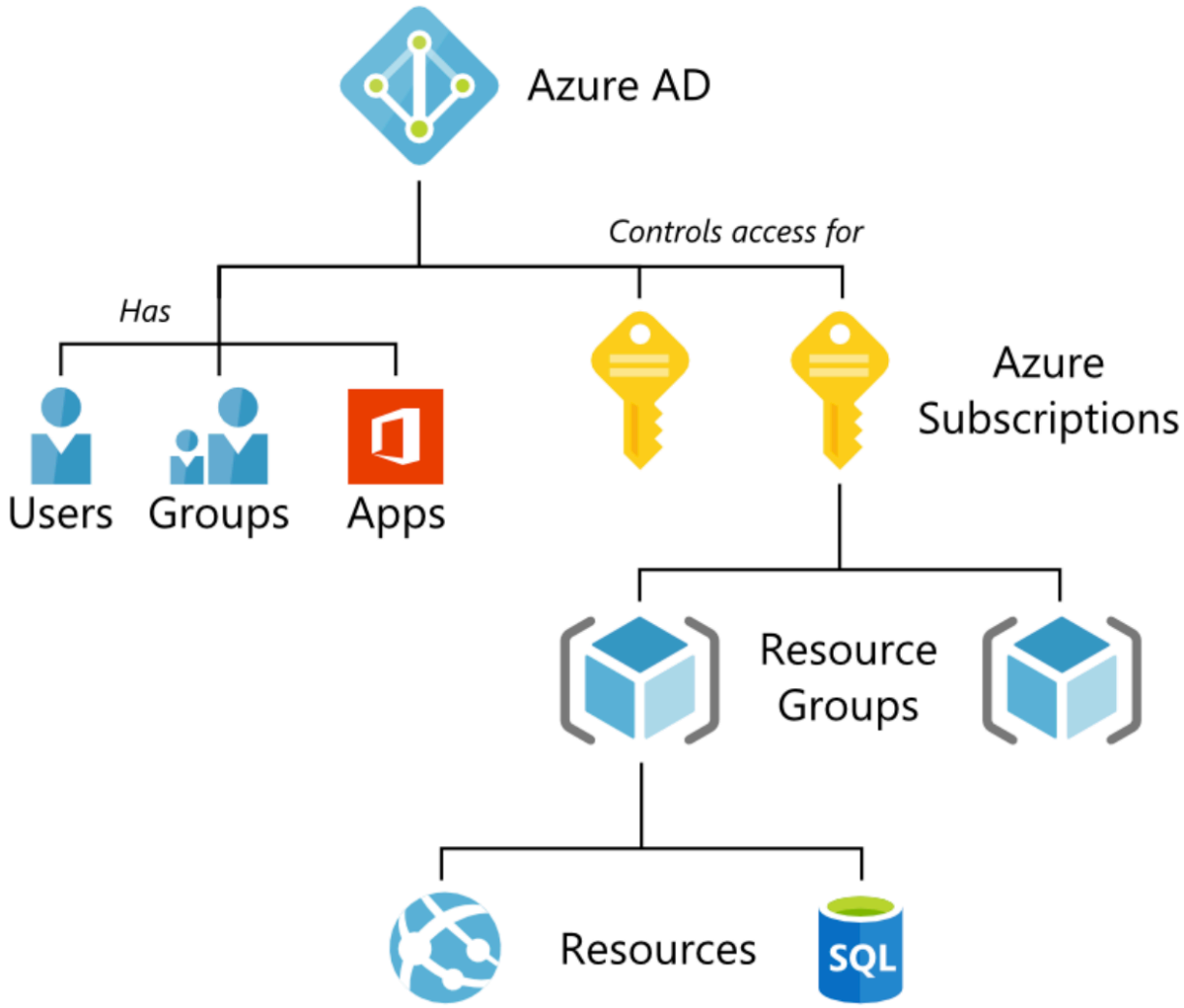


Azure Data Factory is a managed cloud service that's built for these complex hybrid extract-transform-load (ETL), extract-load-transform (ELT), and data integration projects.



- APIs Caching. Operations in API Management can be configured for response caching. Response caching can significantly reduce API latency, bandwidth consumption, and web service load for data that does not change frequently.

- Azure SQL has a management plane and a data plane. The data plane is for actual database data (tables, columns, values, etc). Creating any RBAC roles or assignments would only help in controlling access to administer the Azure SQL resources. Not the database tables.



- Action Groups does not monitor subscription level activities, they are for creating alerts. Action Groups can be used to create a range of alerts or actions, including email/SMS, trigger automation runbooks, and more. Action Groups are used within the Azure Monitor platform in response to certain conditions.
  The Azure Activity Log provides details about what, who, and when to write operations at the subscription level.
  Instrumentation is required to ensure telemetry is sent from your web application to Application Insights.
  In this scenario, single sign-on is already established using (ADFS).

- Linked sign-on mode is used to support applications that already use another identity provider for single sign-on. Choose linked sign-on when the application is configured for single sign-on in another identity provider service. This option doesn't add single sign-on to the application. However, the application might already have single sign-on implemented using another

service such as Active Directory Federation Services.
A user changing their own password can only do so if they know their current password. This is different from password reset, where the user's current password is not known.
Azure AD Password Protection is a feature of Azure AD which provides protection against weak passwords, and a smart lockout system to protect against brute-force attacks.

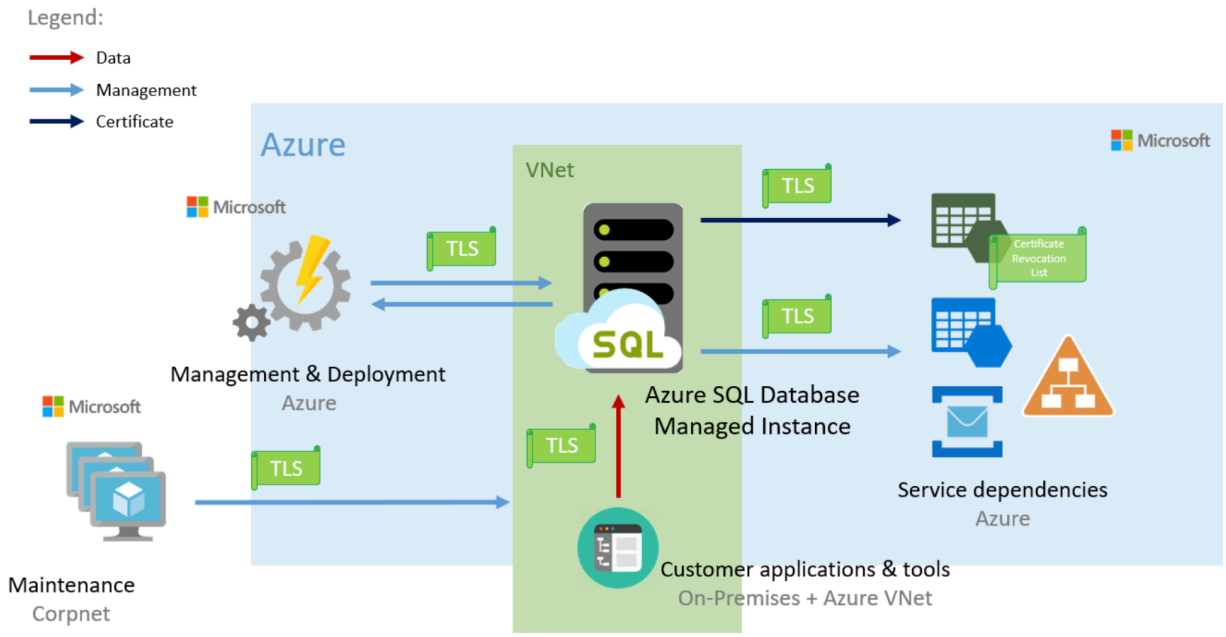
- Service endpoints would not help in establishing connectivity between resources in different subscriptions. Service endpoints provide Azure resources with a direct route by using Microsoft Azure network backbones. This avoids public internet and provides a more secure and direct connection. Note that this does not change firewall rules (only a route is created), nor does it provide private IP addressing to Azure resources. Today, Azure service traffic from a virtual network uses public IP addresses as source IP addresses. With service endpoints, service traffic switches to use virtual network private addresses as the source IP addresses when accessing the Azure service from a virtual network. This switch allows you to access the services without the need for reserved, public IP addresses used in IP firewalls.

- Azure SQL managed instances are accessible by private IP but do not provide ALL SQL Server capabilities. It supports almost all instance-level and database-level capabilities. Only SQL virtual machines will provide ALL SQL Server capabilities.

The SQL Database managed instance is placed inside the Azure virtual network and the subnet that's dedicated to managed instances. This deployment provides:
- A secure private IP address.
- The ability to connect an on-premises network to a managed instance.
- The ability to connect a managed instance to a linked server or another on-premises data store.
- The ability to connect a managed instance to Azure resources.

Communication overview

The following diagram shows entities that connect to a managed instance. It also shows the resources that need to communicate with the managed instance. The communication process at the bottom of the diagram represents customer applications and tools that connect to the managed instance as data sources.



A managed instance is a platform as a service (PaaS) offering. Microsoft uses automated agents (management, deployment, and maintenance) to manage this service based on telemetry data streams. Because Microsoft is responsible for management, customers can't access the managed instance virtual cluster machines through Remote Desktop Protocol (RDP).

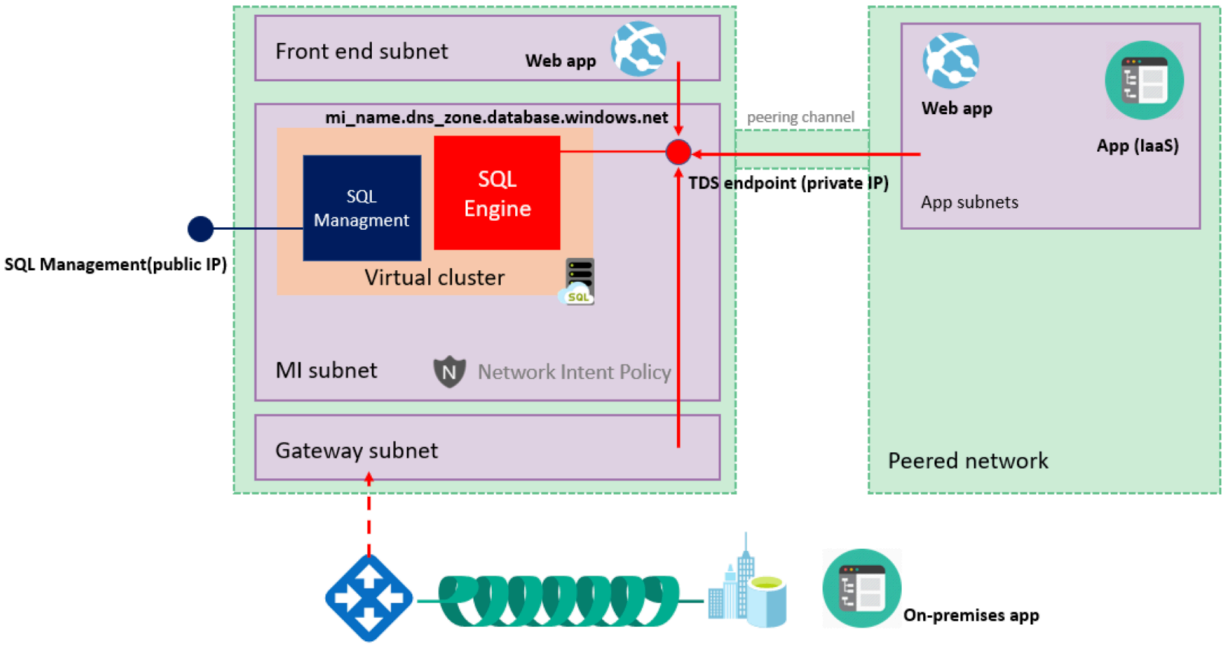
Some SQL Server operations started by end users or applications might require managed instances to interact with the platform. One case is the creation of a managed instance database. This resource is exposed through the Azure portal, PowerShell, Azure CLI, and the REST API.

Managed instances depend on Azure services such as Azure Storage for backups, Azure Event Hubs for telemetry, Azure Active Directory for authentication, Azure Key Vault for Transparent Data Encryption (TDE) and a couple of Azure platform services that provide security and supportability features. The managed instances makes connections to these services.

High-level connectivity architecture

At a high level, a managed instance is a set of service components. These components are hosted on a dedicated set of isolated virtual machines that run inside the customer's virtual network subnet. These machines form a virtual cluster.
A virtual cluster can host multiple managed instances. If needed, the cluster automatically expands or contracts when the customer changes the number of

provisioned instances in the subnet.
Customer applications can connect to managed instances and can query and update databases inside the virtual network, peered virtual network, or network connected by VPN or Azure ExpressRoute. This network must use an endpoint and a private IP address.



Microsoft management and deployment services run outside the virtual network. A managed instance and Microsoft services connect over the endpoints that have public IP addresses. When a managed instance creates an outbound connection, on receiving end Network Address Translation (NAT) makes the connection look like it’s coming from this public IP address.

Important
To improve customer experience and service availability, Microsoft applies a network intent policy on Azure virtual network infrastructure elements. The policy can affect how the managed instance works. This platform mechanism transparently communicates networking requirements to users. The policy's main goal is to prevent network misconfiguration and to ensure normal managed instance operations. When you delete a managed instance, the network intent policy is also removed.

Management and deployment services connect to a managed instance by using a management endpoint that maps to an external load balancer. Traffic is routed to the nodes only if it's received on a predefined set of ports that only the managed instance's management components use. A built-in firewall on the nodes is set up

to allow traffic only from Microsoft IP ranges. Certificates mutually authenticate all communication between management components and the management plane.

Management endpoint

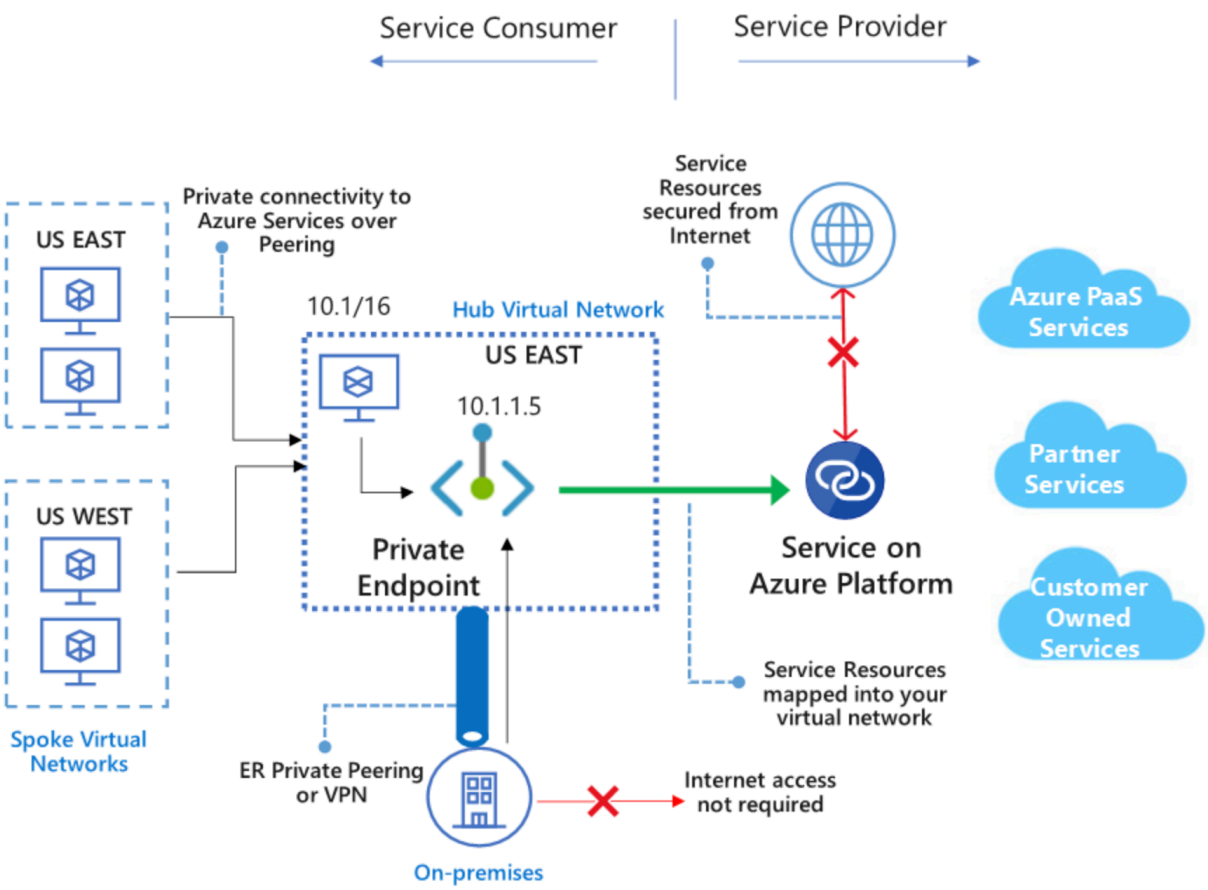
Microsoft manages the managed instance by using a management endpoint. This endpoint is inside the instance's virtual cluster. The management endpoint is protected by a built-in firewall on the network level. On the application level, it's protected by mutual certificate verification.

A managed instance is created in own VNet with no public endpoint. For client application access, you can either **create a VM in the same VNet (different subnet)** or **create a point-to-site VPN connection to the VNet from your client computer** using one of these quickstarts:

- Enable public endpoint on your Managed Instance in order to access your data directly from your environment.
- Create Azure Virtual Machine in the managed instance VNet for client application connectivity, including SQL Server Management Studio.
- Set up point-to-site VPN connection to your managed instance from your client computer on which you have SQL Server Management Studio and other client connectivity applications. This is other of two options for connectivity to your managed instance and to its VNet.

What is Azure Private Link? (Preview)

Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage, Azure Cosmos DB, and SQL Database) and Azure hosted customer/ partner services over a Private Endpoint in your virtual network. Traffic between your virtual network and the service traverses over the Microsoft backbone network, eliminating exposure from the public Internet. You can also create your own Private Link Service in your virtual network (VNet) and deliver it privately to your customers. The setup and consumption experience using Azure Private Link is consistent across Azure PaaS, customer-owned, and shared partner services.



Key benefits

Azure Private Link provides the following benefits:

- **Privately access services on the Azure platform**: Connect your virtual network to services running in Azure privately without needing a public IP address at the source or destination. Service providers can render their services privately in their own virtual network and consumers can access those services privately in their local virtual network. The Private Link platform will handle the connectivity between the consumer and services over the Azure backbone network.
- **On-premises and peered networks**: Access services running in Azure from on-premises over ExpressRoute private peering/VPN tunnels (from on-premises) and peered virtual networks using private endpoints. There is no need to set up public peering or traverse the internet to reach the service. This ability provides a secure way to migrate workloads to Azure.
- **Protection against data exfiltration**: With Azure Private Link, the private endpoint in the VNet is mapped to a specific instance of the customer's PaaS resource as opposed to the entire service. Using the private endpoint consumers can only connect to the specific resource and not to any other resource in the service. This in built mechanism provides protection against

data exfiltration risks.

- **Global reach**: Connect privately to services running in other regions. This means that the consumer's virtual network could be in region A and it can connect to services behind Private Link in region B.
- **Extend to your own services**: Leverage the same experience and functionality to render your own service privately to your consumers in Azure. By placing your service behind a Standard Load Balancer you can enable it for Private Link. The consumer can then connect directly to your service using a Private Endpoint in their own VNet. You can manage these connection requests using a simple approval call flow. Azure Private Link works for consumers and services belonging to different Active Directory tenants as well.

Availability

The following table lists the Private Link services and the regions where they are available.

Scenario	Supported services	Available regions	Status
Private Link for customer-owned services	Private Link services behind Standard Load Balancer	All public regions	Preview
Private Link for Azure PaaS services	Azure Storage	All public regions	Preview Learn more.
	Azure Data Lake Storage Gen2	All public regions	Preview Learn more.
	Azure SQL Database	All public regions	Preview
	Azure SQL Data Warehouse	All public regions	Preview
	Azure Cosmos DB	West Central US, WestUS, North Central US	Preview

Logging and monitoring

Azure Private Link is integrated with Azure Monitor which allows you to archive logs to a storage account, stream events to your Event Hub, or send them to Azure Monitor logs. You can access the following information on Azure Monitor:

- **Private endpoint**: Data processed by the Private Endpoint (IN/OUT)
- **Private Link service**:
    - Data processed by the Private Link service (IN/OUT)
    - NAT port availability