

Recommendation	Description	Severity
AWS WAF Classic global web ACL logging should be enabled ↴	<p>This control checks whether logging is enabled for an AWS WAF global Web ACL. This control fails if logging is not enabled for the web ACL.</p> <p>Logging is an important part of maintaining the reliability, availability, and performance of AWS WAF globally. It is a business and compliance requirement in many organizations, and allows you to troubleshoot application behavior. It also provides detailed information about the traffic that is analyzed by the web ACL that is attached to AWS WAF.</p>	Medium
CloudFront distributions should have a default root object configured ↴	<p>This control checks whether an Amazon CloudFront distribution is configured to return a specific object that is the default root object. The control fails if the CloudFront distribution does not have a default root object configured.</p> <p>A user might sometimes request the distributions root URL instead of an object in the distribution. When this happens, specifying a default root object can help you to avoid exposing the contents of your web distribution.</p>	High
CloudFront distributions should have origin access identity enabled ↴	<p>This control checks whether an Amazon CloudFront distribution with Amazon S3 Origin type has Origin Access Identity (OAI) configured. The control fails if OAI is not configured.</p> <p>CloudFront OAI prevents users from accessing S3 bucket content directly. When users access an S3 bucket directly, they effectively bypass the CloudFront distribution and any permissions that are applied to the underlying S3 bucket content.</p>	Medium
CloudTrail log file validation should be enabled ↴	<p>To ensure additional integrity checking of CloudTrail logs, we recommend enabling file validation on all CloudTrails.</p>	Low
CloudTrail should be enabled ↴	<p>AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. Not all services enable logging by default for all APIs and events.</p> <p>You should implement any additional audit trails other than CloudTrail and review the documentation for each service in CloudTrail Supported Services and Integrations.</p>	High

Recommendation	Description	Severity
CloudTrail trails should be integrated with CloudWatch Logs ↴	<p>In addition to capturing CloudTrail logs within a specified S3 bucket for long term analysis, real-time analysis can be performed by configuring CloudTrail to send logs to CloudWatch Logs.</p> <p>For a trail that is enabled in all regions in an account, CloudTrail sends log files from all those regions to a CloudWatch Logs log group. We recommended that CloudTrail logs will be sent to CloudWatch Logs to ensure AWS account activity is being captured, monitored, and appropriately alarmed on.</p> <p>Sending CloudTrail logs to CloudWatch Logs facilitates real-time and historic activity logging based on user, API, resource, and IP address, and provides opportunity to establish alarms and notifications for anomalous or sensitivity account activity.</p>	Low
Database logging should be enabled ↴	<p>This control checks whether the following logs of Amazon RDS are enabled and sent to CloudWatch Logs:</p> <ul style="list-style-type: none"> - Oracle: (Alert, Audit, Trace, Listener) - PostgreSQL: (Postgresql, Upgrade) - MySQL: (Audit, Error, General, SlowQuery) - MariaDB: (Audit, Error, General, SlowQuery) - SQL Server: (Error, Agent) - Aurora: (Audit, Error, General, SlowQuery) - Aurora-MySQL: (Audit, Error, General, SlowQuery) - Aurora-PostgreSQL: (Postgresql, Upgrade). <p>RDS databases should have relevant logs enabled. Database logging provides detailed records of requests made to RDS. Database logs can assist with security and access audits and can help to diagnose availability issues.</p>	Medium
Disable direct internet access for Amazon SageMaker notebook instances ↴	<p>Direct internet access should be disabled for an SageMaker notebook instance.</p> <p>This checks whether the 'DirectInternetAccess' field is disabled for the notebook instance.</p> <p>Your instance should be configured with a VPC and the default setting should be Disable - Access the internet through a VPC.</p> <p>In order to enable internet access to train or host models from a notebook, make sure that your VPC has a NAT gateway and your security group allows outbound connections. Ensure access to your SageMaker configuration is limited to only authorized users, and restrict users' IAM permissions to modify SageMaker settings and resources.</p>	High

Recommendation	Description	Severity
Do not setup access keys during initial user setup for all IAM users that have a console password ↴	<p>AWS console defaults the checkbox for creating access keys to enabled. This results in many access keys being generated unnecessarily.</p> <p>In addition to unnecessary credentials, it also generates unnecessary management work in auditing and rotating these keys.</p> <p>Requiring that additional steps be taken by the user after their profile has been created will give a stronger indication of intent that access keys are [a] necessary for their work and [b] once the access key is established on an account that the keys may be in use somewhere in the organization</p>	Medium
Ensure a support role has been created to manage incidents with AWS Support ↴	<p>AWS provides a support center that can be used for incident notification and response, as well as technical support and customer services.</p> <p>Create an IAM Role to allow authorized users to manage incidents with AWS Support.</p> <p>By implementing least privilege for access control, an IAM Role will require an appropriate IAM Policy to allow Support Center Access in order to manage Incidents with AWS Support.</p>	Low
Ensure access keys are rotated every 90 days or less ↴	<p>Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS.</p> <p>AWS users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services.</p> <p>It is recommended that all access keys be regularly rotated. Rotating access keys will reduce the window of opportunity for an access key that is associated with a compromised or terminated account to be used.</p> <p>Access keys should be rotated to ensure that data cannot be accessed with an old key which might have been lost, cracked, or stolen.</p>	Medium

Recommendation	Description	Severity
Ensure AWS Config is enabled in all regions	<p>AWS Config is a web service that performs configuration management of supported AWS resources within your account and delivers log files to you.</p> <p>The recorded information includes the configuration item (AWS resource), relationships between configuration items (AWS resources), any configuration changes between resources.</p> <p>It is recommended to enable AWS Config be enabled in all regions.</p> <p>The AWS configuration item history captured by AWS Config enables security analysis, resource change tracking, and compliance auditing.</p>	Medium
Ensure CloudTrail is enabled in all regions	<p>AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you.</p> <p>The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service. CloudTrail provides a history of AWS API calls for an account, including API calls made via the Management Console, SDKs, command line tools, and higher-level AWS services (such as CloudFormation).</p> <p>The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.</p> <p>Additionally,</p> <ul style="list-style-type: none"> * ensuring that a multi-regions trail exists will ensure that unexpected activity occurring in otherwise unused regions is detected * ensuring that a multi-regions trail exists will ensure that "Global Service Logging" is enabled for a trail by default to capture recording of events generated on AWS global services * for a multi-regions trail, ensuring that management events configured for all type of Read/Writes ensures recording of management operations that are performed on all resources in an AWS account. 	High
Ensure credentials unused for 90 days or greater are disabled	<p>AWS IAM users can access AWS resources using different types of credentials, such as passwords or access keys.</p> <p>It is recommended that all credentials that have been unused in 90 or greater days be removed or deactivated.</p> <p>Disabling or removing unnecessary credentials will reduce the window of opportunity for credentials associated with a compromised or abandoned account to be used.</p>	Medium

Recommendation	Description	Severity
Ensure IAM password policy expires passwords within 90 days or less ↗	<p>IAM password policies can require passwords to be rotated or expired after a given number of days.</p> <p>It is recommended that the password policy expire passwords after 90 days or less.</p> <p>Reducing the password lifetime increases account resiliency against brute force login attempts. Additionally, requiring regular password changes help in the following scenarios:</p> <ul style="list-style-type: none"> * Passwords can be stolen or compromised sometimes without your knowledge. This can happen via a system compromise, software vulnerability, or internal threat. * Certain corporate and government web filters or proxy servers have the ability to intercept and record traffic even if it's encrypted. * Many people use the same password for many systems such as work, email, and personal. * Compromised end user workstations might have a keystroke logger. 	Low
Ensure IAM password policy prevents password reuse ↗	<p>IAM password policies can prevent the reuse of a given password by the same user.</p> <p>It is recommended that the password policy prevent the reuse of passwords.</p> <p>Preventing password reuse increases account resiliency against brute force login attempts.</p>	Low
Ensure IAM password policy requires at least one lowercase letter ↗	<p>Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets.</p> <p>It is recommended that the password policy require at least one lowercase letter.</p> <p>Setting a password complexity policy increases account resiliency against brute force login attempts</p>	Medium
Ensure IAM password policy requires at least one number ↗	<p>Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets.</p> <p>It is recommended that the password policy require at least one number.</p> <p>Setting a password complexity policy increases account resiliency against brute force login attempts.</p>	Medium

Recommendation	Description	Severity
Ensure IAM password policy requires at least one symbol ↴	<p>Password policies are, in part, used to enforce password complexity requirements.</p> <p>IAM password policies can be used to ensure password are comprised of different character sets.</p> <p>It is recommended that the password policy require at least one symbol.</p> <p>Setting a password complexity policy increases account resiliency against brute force login attempts.</p>	Medium
Ensure IAM password policy requires at least one uppercase letter ↴	<p>Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets.</p> <p>It is recommended that the password policy require at least one uppercase letter.</p> <p>Setting a password complexity policy increases account resiliency against brute force login attempts.</p>	Medium
Ensure IAM password policy requires minimum length of 14 or greater ↴	<p>Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are at least a given length.</p> <p>It is recommended that the password policy require a minimum password length '14'.</p> <p>Setting a password complexity policy increases account resiliency against brute force login attempts.</p>	Medium
Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password ↴	<p>Multi-Factor Authentication (MFA) adds an extra layer of protection on top of a user name and password.</p> <p>With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device.</p> <p>It is recommended that MFA be enabled for all accounts that have a console password.</p> <p>Enabling MFA provides increased security for console access as it requires the authenticating principal to possess a device that emits a time-sensitive key and have knowledge of a credential.</p>	Medium
GuardDuty should be enabled ↴	<p>To provide additional protection against intrusions, GuardDuty should be enabled on your AWS account and region.</p> <p>Note: GuardDuty might not be a complete solution for every environment</p>	Medium

Recommendation	Description	Severity
Hardware MFA should be enabled for the "root" account ↗	<p>The root account is the most privileged user in an account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they're prompted for their user name and password and for an authentication code from their AWS MFA device.</p> <p>For Level 2, it is recommended that you protect the root account with a hardware MFA. A hardware MFA has a smaller attack surface than a virtual MFA. For example, a hardware MFA doesn't suffer the attack surface introduced by the mobile smartphone that a virtual MFA resides on.</p> <p>Using hardware MFA for many, many accounts might create a logistical device management issue. If this occurs, consider implementing this Level 2 recommendation selectively to the highest security accounts. You can then apply the Level 1 recommendation to the remaining accounts.</p>	Low
IAM authentication should be configured for RDS clusters ↗	<p>This control checks whether an RDS DB cluster has IAM database authentication enabled.</p> <p>IAM database authentication allows for password-free authentication to database instances. The authentication uses an authentication token. Network traffic to and from the database is encrypted using SSL. For more information, see IAM database authentication in the Amazon Aurora User Guide.</p>	Medium
IAM authentication should be configured for RDS instances ↗	<p>This control checks whether an RDS DB instance has IAM database authentication enabled.</p> <p>IAM database authentication allows authentication to database instances with an authentication token instead of a password. Network traffic to and from the database is encrypted using SSL. For more information, see IAM database authentication in the Amazon Aurora User Guide.</p>	Medium

Recommendation	Description	Severity
IAM customer managed policies should not allow decryption actions on all KMS keys	<p>Checks whether the default version of IAM customer managed policies allow principals to use the AWS KMS decryption actions on all resources. This control uses Zelkova, an automated reasoning engine, to validate and warn you about policies that may grant broad access to your secrets across AWS accounts. This control fails if the "kms:Decrypt" or "kms:ReEncryptFrom" actions are allowed on all KMS keys. The control evaluates both attached and unattached customer managed policies. It does not check inline policies or AWS managed policies.</p> <p>With AWS KMS, you control who can use your KMS keys and gain access to your encrypted data. IAM policies define which actions an identity (user, group, or role) can perform on which resources. Following security best practices, AWS recommends that you allow least privilege. In other words, you should grant to identities only the "kms:Decrypt" or "kms:ReEncryptFrom" permissions and only for the keys that are required to perform a task. Otherwise, the user might use keys that are not appropriate for your data.</p> <p>Instead of granting permissions for all keys, determine the minimum set of keys that users need to access encrypted data. Then design policies that allow users to use only those keys. For example, do not allow "kms:Decrypt" permission on all KMS keys. Instead, allow "kms:Decrypt" only on keys in a particular Region for your account. By adopting the principle of least privilege, you can reduce the risk of unintended disclosure of your data.</p>	Medium

Recommendation	Description	Severity
IAM customer managed policies that you create should not allow wildcard actions for services ↴	<p>This control checks whether the IAM identity-based policies that you create have Allow statements that use the * wildcard to grant permissions for all actions on any service. The control fails if any policy statement includes 'Effect': 'Allow' with 'Action': 'Service:'. <i>For example, the following statement in a policy results in a failed finding.</i></p> <pre data-bbox="457 428 742 720">'Statement': [{ 'Sid': 'EC2-Wildcard', 'Effect': 'Allow', 'Action': 'ec2:', 'Resource': '' }]</pre> <p><i>The control also fails if you use 'Effect': 'Allow' with 'NotAction': 'service:'. In that case, the NotAction element provides access to all of the actions in an AWS service, except for the actions specified in NotAction.</i></p> <p>This control only applies to customer managed IAM policies. It does not apply to IAM policies that are managed by AWS. When you assign permissions to AWS services, it is important to scope the allowed IAM actions in your IAM policies. You should restrict IAM actions to only those actions that are needed. This helps you to provision least privilege permissions. Overly permissive policies might lead to privilege escalation if the policies are attached to an IAM principal that might not require the permission.</p> <p>In some cases, you might want to allow IAM actions that have a similar prefix, such as <code>DescribeFlowLogs</code> and <code>DescribeAvailabilityZones</code>. In these authorized cases, you can add a suffixed wildcard to the common prefix. For example, <code>ec2:Describe*</code>.</p> <p>This control passes if you use a prefixed IAM action with a suffixed wildcard. For example, the following statement in a policy results in a passed finding.</p> <pre data-bbox="457 1641 790 1933">'Statement': [{ 'Sid': 'EC2-Wildcard', 'Effect': 'Allow', 'Action': 'ec2:Describe*', 'Resource': '*' }]</pre> <p>When you group related IAM actions in this way, you can also avoid exceeding the IAM policy size limits.</p>	Low

Recommendation	Description	Severity
IAM policies should be attached only to groups or roles	<p>By default, IAM users, groups, and roles have no access to AWS resources. IAM policies are the means by which privileges are granted to users, groups, or roles.</p> <p>It is recommended that IAM policies be applied directly to groups and roles but not users.</p> <p>Assigning privileges at the group or role level reduces the complexity of access management as the number of users grow. Reducing access management complexity may in-turn reduce opportunity for a principal to inadvertently receive or retain excessive privileges.</p>	Low
IAM policies that allow full ":" administrative privileges should not be created	<p>IAM policies are the means by which privileges are granted to users, groups, or roles.</p> <p>It is recommended and considered a standard security advice to grant least privilege—that is, granting only the permissions required to perform a task.</p> <p>Determine what users need to do and then craft policies for them that let the users perform only those tasks, instead of allowing full administrative privileges.</p> <p>It's more secure to start with a minimum set of permissions and grant additional permissions as necessary, rather than starting with permissions that are too lenient and then trying to tighten them later.</p> <p>Providing full administrative privileges instead of restricting to the minimum set of permissions that the user is required to do exposes the resources to potentially unwanted actions.</p> <p>IAM policies that have a statement with "Effect": "Allow" with "Action": "" over "Resource": "" should be removed.</p>	High

Recommendation	Description	Severity
IAM principals should not have IAM inline policies that allow decryption actions on all KMS keys ↗	<p>Checks whether the inline policies that are embedded in your IAM identities (role, user, or group) allow the AWS KMS decryption actions on all KMS keys. This control uses Zelkova ↗, an automated reasoning engine, to validate and warn you about policies that may grant broad access to your secrets across AWS accounts.</p> <p>This control fails if "kms:Decrypt" or "kms:ReEncryptFrom" actions are allowed on all KMS keys in an inline policy.</p> <p>With AWS KMS, you control who can use your KMS keys and gain access to your encrypted data. IAM policies define which actions an identity (user, group, or role) can perform on which resources. Following security best practices, AWS recommends that you allow least privilege. In other words, you should grant to identities only the permissions they need and only for keys that are required to perform a task. Otherwise, the user might use keys that are not appropriate for your data.</p> <p>Instead of granting permission for all keys, determine the minimum set of keys that users need to access encrypted data. Then design policies that allow the users to use only those keys. For example, do not allow "kms:Decrypt" permission on all KMS keys. Instead, allow them only on keys in a particular Region for your account. By adopting the principle of least privilege, you can reduce the risk of unintended disclosure of your data.</p>	Medium
Lambda functions should restrict public access ↗	<p>Lambda function resource-based policy should restrict public access. This recommendation does not check access by internal principals.</p> <p>Ensure access to the function is restricted to authorized principals only by using least privilege resource-based policies.</p>	High
MFA should be enabled for all IAM users ↗	<p>All IAM users should have multi-factor authentication (MFA) enabled.</p>	Medium
MFA should be enabled for the "root" account ↗	<p>The root account is the most privileged user in an account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they're prompted for their user name and password and for an authentication code from their AWS MFA device.</p> <p>When you use virtual MFA for root accounts, it is recommended that the device used is not a personal device. Instead, use a dedicated mobile device (tablet or phone) that you manage to keep charged and secured independent of any individual personal devices.</p> <p>This lessens the risks of losing access to the MFA due to device loss, device trade-in, or if the individual owning the device is no longer employed at the company.</p>	Low

Recommendation	Description	Severity
Password policies for IAM users should have strong configurations ↴	<p>Checks whether the account password policy for IAM users uses the following minimum configurations.</p> <ul style="list-style-type: none"> * <code>RequireUppercaseCharacters</code>- Require at least one uppercase character in password. (Default = true) * <code>RequireLowercaseCharacters</code>- Require at least one lowercase character in password. (Default = true) * <code>RequireNumbers</code>- Require at least one number in password. (Default = true) * <code>MinimumPasswordLength</code>- Password minimum length. (Default = 7 or longer) * <code>PasswordReusePrevention</code>- Number of passwords before allowing reuse. (Default = 4) * <code>MaxPasswordAge</code>- Number of days before password expiration. (Default = 90) 	Medium
Root account access key shouldn't exist ↴	<p>The root account is the most privileged user in an AWS account. AWS Access Keys provide programmatic access to a given AWS account.</p> <p>It is recommended that all access keys associated with the root account be removed.</p> <p>Removing access keys associated with the root account limits vectors by which the account can be compromised.</p> <p>Additionally, removing the root access keys encourages the creation and use of role based accounts that are least privileged.</p>	High
S3 Block Public Access setting should be enabled ↴	<p>Enabling Block Public Access setting for your S3 bucket can help prevent sensitive data leaks and protect your bucket from malicious actions.</p>	Medium
S3 Block Public Access setting should be enabled at the bucket level ↴	<p>This control checks whether S3 buckets have bucket-level public access blocks applied. This control fails if any of the following settings are set to false:</p> <ul style="list-style-type: none"> * <code>ignorePublicAcls</code> * <code>blockPublicPolicy</code> * <code>blockPublicAcls</code> * <code>restrictPublicBuckets</code> <p>Block Public Access at the S3 bucket level provides controls to ensure that objects never have public access. Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both.</p> <p>Unless you intend to have your S3 buckets publicly accessible, you should configure the bucket level Amazon S3 Block Public Access feature.</p>	High

Recommendation	Description	Severity
S3 buckets public read access should be removed ↴	Removing public read access to your S3 bucket can help protect your data and prevent a data breach.	High
S3 buckets public write access should be removed ↴	Allowing public write access to your S3 bucket can leave you vulnerable to malicious actions such as storing data at your expense, encrypting your files for ransom, or using your bucket to operate malware.	High
Secrets Manager secrets should have automatic rotation enabled ↴	<p>This control checks whether a secret stored in AWS Secrets Manager is configured with automatic rotation. Secrets Manager helps you improve the security posture of your organization. Secrets include database credentials, passwords, and third-party API keys. You can use Secrets Manager to store secrets centrally, encrypt secrets automatically, control access to secrets, and rotate secrets safely and automatically. Secrets Manager can rotate secrets. You can use rotation to replace long-term secrets with short-term ones. Rotating your secrets limits how long an unauthorized user can use a compromised secret. For this reason, you should rotate your secrets frequently. To learn more about rotation, see Rotating your AWS Secrets Manager secrets ↴ in the AWS Secrets Manager User Guide.</p>	Medium
Stopped EC2 instances should be removed after a specified time period ↴	This control checks whether any EC2 instances have been stopped for more than the allowed number of days. An EC2 instance fails this check if it is stopped for longer than the maximum allowed time period, which by default is 30 days. A failed finding indicates that an EC2 instance has not run for a significant period of time. This creates a security risk because the EC2 instance is not being actively maintained (analyzed, patched, updated). If it is later launched, the lack of proper maintenance could result in unexpected issues in your AWS environment. To safely maintain an EC2 instance over time in a nonrunning state, start it periodically for maintenance and then stop it after maintenance. Ideally this is an automated process.	Medium

AWS Networking recommendations

There are **36** AWS recommendations in this category.

Recommendation	Description	Severity
----------------	-------------	----------

Recommendation	Description	Severity
Amazon EC2 should be configured to use VPC endpoints	<p>This control checks whether a service endpoint for Amazon EC2 is created for each VPC. The control fails if a VPC does not have a VPC endpoint created for the Amazon EC2 service.</p> <p>To improve the security posture of your VPC, you can configure Amazon EC2 to use an interface VPC endpoint. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to access Amazon EC2 API operations privately. It restricts all network traffic between your VPC and Amazon EC2 to the Amazon network. Because endpoints are supported within the same Region only, you cannot create an endpoint between a VPC and a service in a different Region. This prevents unintended Amazon EC2 API calls to other Regions.</p> <p>To learn more about creating VPC endpoints for Amazon EC2, see Amazon EC2 and interface VPC endpoints in the Amazon EC2 User Guide for Linux Instances.</p>	Medium
Amazon ECS services should not have public IP addresses assigned to them automatically	<p>A public IP address is an IP address that is reachable from the internet.</p> <p>If you launch your Amazon ECS instances with a public IP address, then your Amazon ECS instances are reachable from the internet.</p> <p>Amazon ECS services should not be publicly accessible, as this may allow unintended access to your container application servers.</p>	High
Amazon EMR cluster master nodes should not have public IP addresses	<p>This control checks whether master nodes on Amazon EMR clusters have public IP addresses.</p> <p>The control fails if the master node has public IP addresses that are associated with any of its instances. Public IP addresses are designated in the PublicIp field of the NetworkInterfaces configuration for the instance.</p> <p>This control only checks Amazon EMR clusters that are in a RUNNING or WAITING state.</p>	High
Amazon Redshift clusters should use enhanced VPC routing	<p>This control checks whether an Amazon Redshift cluster has EnhancedVpcRouting enabled.</p> <p>Enhanced VPC routing forces all COPY and UNLOAD traffic between the cluster and data repositories to go through your VPC. You can then use VPC features such as security groups and network access control lists to secure network traffic. You can also use VPC Flow Logs to monitor network traffic.</p>	High
Application Load Balancer should be configured to redirect all HTTP requests to HTTPS	<p>To enforce encryption in transit, you should use redirect actions with Application Load Balancers to redirect client HTTP requests to an HTTPS request on port 443.</p>	Medium

Recommendation	Description	Severity
Application load balancers should be configured to drop HTTP headers	<p>This control evaluates AWS Application Load Balancers (ALB) to ensure they are configured to drop invalid HTTP headers. The control fails if the value of <code>routing.http.drop_invalid_header_fields.enabled</code> is set to false. By default, ALBs are not configured to drop invalid HTTP header values. Removing these header values prevents HTTP desync attacks.</p>	Medium
Configure Lambda functions to a VPC	<p>This control checks whether a Lambda function is in a VPC. It does not evaluate the VPC subnet routing configuration to determine public reachability.</p> <p>Note that if Lambda@Edge is found in the account, then this control generates failed findings. To prevent these findings, you can disable this control.</p>	Low
EC2 instances should not have a public IP address	<p>This control checks whether EC2 instances have a public IP address. The control fails if the "publicIp" field is present in the EC2 instance configuration item. This control applies to IPv4 addresses only.</p> <p>A public IPv4 address is an IP address that is reachable from the internet. If you launch your instance with a public IP address, then your EC2 instance is reachable from the internet. A private IPv4 address is an IP address that is not reachable from the internet. You can use private IPv4 addresses for communication between EC2 instances in the same VPC or in your connected private network.</p> <p>IPv6 addresses are globally unique, and therefore are reachable from the internet. However, by default all subnets have the IPv6 addressing attribute set to false. For more information about IPv6, see IP addressing in your VPC in the Amazon VPC User Guide.</p> <p>If you have a legitimate use case to maintain EC2 instances with public IP addresses, then you can suppress the findings from this control. For more information about front-end architecture options, see the AWS Architecture Blog or the This Is My Architecture series.</p>	High
EC2 instances should not use multiple ENIs	<p>This control checks whether an EC2 instance uses multiple Elastic Network Interfaces (ENIs) or Elastic Fabric Adapters (EFAs). This control passes if a single network adapter is used. The control includes an optional parameter list to identify the allowed ENIs. Multiple ENIs can cause dual-homed instances, meaning instances that have multiple subnets. This can add network security complexity and introduce unintended network paths and access.</p>	Low

Recommendation	Description	Severity
EC2 instances should use IMDSv2 ↴	<p>This control checks whether your EC2 instance metadata version is configured with Instance Metadata Service Version 2 (IMDSv2). The control passes if "HttpTokens" is set to "required" for IMDSv2. The control fails if "HttpTokens" is set to "optional". You use instance metadata to configure or manage the running instance. The IMDS provides access to temporary, frequently rotated credentials. These credentials remove the need to hard code or distribute sensitive credentials to instances manually or programmatically. The IMDS is attached locally to every EC2 instance. It runs on a special 'link local' IP address of 169.254.169.254. This IP address is only accessible by software that runs on the instance.</p> <p>Version 2 of the IMDS adds new protections for the following types of vulnerabilities. These vulnerabilities could be used to try to access the IMDS.</p> <ul style="list-style-type: none"> * Open website application firewalls * Open reverse proxies * Server-side request forgery (SSRF) vulnerabilities * Open Layer 3 firewalls and network address translation (NAT) <p>Security Hub recommends that you configure your EC2 instances with IMDSv2.</p>	High
EC2 subnets should not automatically assign public IP addresses ↴	<p>This control checks whether the assignment of public IPs in Amazon Virtual Private Cloud (Amazon VPC) subnets have "MapPublicIpOnLaunch" set to "FALSE". The control passes if the flag is set to "FALSE".</p> <p>All subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address. Instances that are launched into subnets that have this attribute enabled have a public IP address assigned to their primary network interface.</p>	Medium
Ensure a log metric filter and alarm exist for AWS Config configuration changes ↴	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for detecting changes to CloudTrail's configurations.</p> <p>Monitoring changes to AWS Config configuration will help ensure sustained visibility of configuration items within the AWS account.</p>	Low

Recommendation	Description	Severity
Ensure a log metric filter and alarm exist for AWS Management Console authentication failures ↴	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for failed console authentication attempts.</p> <p>Monitoring failed console logins may decrease lead time to detect an attempt to brute force a credential, which may provide an indicator, such as source IP, that can be used in other event correlation.</p>	Low
Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL) ↴	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. NACLs are used as a stateless packet filter to control ingress and egress traffic for subnets within a VPC.</p> <p>It is recommended that a metric filter and alarm be established for changes made to NACLs.</p> <p>Monitoring changes to NACLs will help ensure that AWS resources and services are not unintentionally exposed.</p>	Low
Ensure a log metric filter and alarm exist for changes to network gateways ↴	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Network gateways are required to send/receive traffic to a destination outside of a VPC.</p> <p>It is recommended that a metric filter and alarm be established for changes to network gateways.</p> <p>Monitoring changes to network gateways will help ensure that all ingress/egress traffic traverses the VPC border via a controlled path.</p>	Low
Ensure a log metric filter and alarm exist for CloudTrail configuration changes ↴	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for detecting changes to CloudTrail's configurations.</p> <p>Monitoring changes to CloudTrail's configuration will help ensure sustained visibility to activities performed in the AWS account.</p>	Low
Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs ↴	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for customer created CMKs which have changed state to disabled or scheduled deletion.</p> <p>Data encrypted with disabled or deleted keys will no longer be accessible.</p>	Low

Recommendation	Description	Severity
Ensure a log metric filter and alarm exist for IAM policy changes	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for changes made to Identity and Access Management (IAM) policies.</p> <p>Monitoring changes to IAM policies will help ensure authentication and authorization controls remain intact.</p>	Low
Ensure a log metric filter and alarm exist for Management Console sign-in without MFA	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for console logins that are not protected by multi-factor authentication (MFA).</p> <p>Monitoring for single-factor console logins will increase visibility into accounts that are not protected by MFA.</p>	Low
Ensure a log metric filter and alarm exist for route table changes	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Routing tables are used to route network traffic between subnets and to network gateways.</p> <p>It is recommended that a metric filter and alarm be established for changes to route tables.</p> <p>Monitoring changes to route tables will help ensure that all VPC traffic flows through an expected path.</p>	Low
Ensure a log metric filter and alarm exist for S3 bucket policy changes	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for changes to S3 bucket policies.</p> <p>Monitoring changes to S3 bucket policies may reduce time to detect and correct permissive policies on sensitive S3 buckets.</p>	Low
Ensure a log metric filter and alarm exist for security group changes	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Security Groups are a stateful packet filter that controls ingress and egress traffic within a VPC.</p> <p>It is recommended that a metric filter and alarm be established for changes to Security Groups.</p> <p>Monitoring changes to security group will help ensure that resources and services are not unintentionally exposed.</p>	Low

Recommendation	Description	Severity
Ensure a log metric filter and alarm exist for unauthorized API calls 	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for unauthorized API calls.</p> <p>Monitoring unauthorized API calls will help reveal application errors and may reduce time to detect malicious activity.</p>	Low
Ensure a log metric filter and alarm exist for usage of 'root' account 	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is recommended that a metric filter and alarm be established for root login attempts.</p> <p>Monitoring for root account logins will provide visibility into the use of a fully privileged account and an opportunity to reduce the use of it.</p>	Low
Ensure a log metric filter and alarm exist for VPC changes 	<p>Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms.</p> <p>It is possible to have more than 1 VPC within an account, in addition it is also possible to create a peer connection between 2 VPCs enabling network traffic to route between VPCs. It is recommended that a metric filter and alarm be established for changes made to VPCs.</p> <p>Monitoring changes to IAM policies will help ensure authentication and authorization controls remain intact.</p>	Low
Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389 	<p>Security groups provide stateful filtering of ingress/egress network traffic to AWS resources. It is recommended that no security group allows unrestricted ingress access to port 3389.</p> <p>Removing unfettered connectivity to remote console services, such as RDP, reduces a server's exposure to risk.</p>	High
Management ports of EC2 instances should be protected with just-in-time network access control 	<p>Microsoft Defender for Cloud has identified some overly-permissive inbound rules for management ports in your network.</p> <p>Enable just-in-time access control to protect your Instances from internet-based brute-force attacks. Learn more.</p>	High

Recommendation	Description	Severity
RDS databases and clusters should not use a database engine default port	<p>This control checks whether the RDS cluster or instance uses a port other than the default port of the database engine. If you use a known port to deploy an RDS cluster or instance, an attacker can guess information about the cluster or instance. The attacker can use this information in conjunction with other information to connect to an RDS cluster or instance or gain additional information about your application.</p> <p>When you change the port, you must also update the existing connection strings that were used to connect to the old port. You should also check the security group of the DB instance to ensure that it includes an ingress rule that allows connectivity on the new port.</p>	Low
RDS instances should be deployed in a VPC	<p>VPCs provide a number of network controls to secure access to RDS resources. These controls include VPC Endpoints, network ACLs, and security groups. To take advantage of these controls, we recommend that you move EC2-Classic RDS instances to EC2-VPC.</p>	Low
S3 buckets should require requests to use Secure Socket Layer	<p>We recommend to require requests to use Secure Socket Layer (SSL) on all Amazon S3 bucket. S3 buckets should have policies that require all requests ('Action: S3:*') to only accept transmission of data over HTTPS in the S3 resource policy, indicated by the condition key 'aws:SecureTransport'.</p>	Medium
Security groups should not allow ingress from 0.0.0.0/0 to port 22	<p>To reduce the server's exposure, it is recommended not to allow unrestricted ingress access to port '22'.</p>	High

Recommendation	Description	Severity
Security groups should not allow unrestricted access to ports with high risk	<p>This control checks whether unrestricted incoming traffic for the security groups is accessible to the specified ports that have the highest risk. This control passes when none of the rules in a security group allow ingress traffic from 0.0.0.0/0 for those ports. Unrestricted access (0.0.0.0/0) increases opportunities for malicious activity, such as hacking, denial-of-service attacks, and loss of data.</p> <p>Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. No security group should allow unrestricted ingress access to the following ports:</p> <ul style="list-style-type: none"> - 3389 (RDP) - 20, 21 (FTP) - 22 (SSH) - 23 (Telnet) - 110 (POP3) - 143 (IMAP) - 3306 (mySQL) - 8080 (proxy) - 1433, 1434 (MSSQL) - 9200 or 9300 (Elasticsearch) - 5601 (Kibana) - 25 (SMTP) - 445 (CIFS) - 135 (RPC) - 4333 (ahsp) - 5432 (postgresql) - 5500 (fcp-addr-srvr1) 	Medium

Recommendation	Description	Severity
Security groups should only allow unrestricted incoming traffic for authorized ports	<p>This control checks whether the security groups that are in use allow unrestricted incoming traffic. Optionally the rule checks whether the port numbers are listed in the "authorizedTcpPorts" parameter.</p> <ul style="list-style-type: none"> - If the security group rule port number allows unrestricted incoming traffic, but the port number is specified in "authorizedTcpPorts", then the control passes. The default value for "authorizedTcpPorts" is 80, 443. - If the security group rule port number allows unrestricted incoming traffic, but the port number is not specified in authorizedTcpPorts input parameter, then the control fails. - If the parameter is not used, then the control fails for any security group that has an unrestricted inbound rule. <p>Security groups provide stateful filtering of ingress and egress network traffic to AWS. Security group rules should follow the principle of least privileged access. Unrestricted access (IP address with a /0 suffix) increases the opportunity for malicious activity such as hacking, denial-of-service attacks, and loss of data.</p> <p>Unless a port is specifically allowed, the port should deny unrestricted access.</p>	High
Unused EC2 EIPs should be removed	<p>Elastic IP addresses that are allocated to a VPC should be attached to Amazon EC2 instances or in-use elastic network interfaces (ENIs).</p>	Low
Unused network access control lists should be removed	<p>This control checks whether there are any unused network access control lists (ACLs).</p> <p>The control checks the item configuration of the resource "AWS::EC2::NetworkAcl" and determines the relationships of the network ACL.</p> <p>If the only relationship is the VPC of the network ACL, then the control fails.</p> <p>If other relationships are listed, then the control passes.</p>	Low
VPC's default security group should restricts all traffic	<p>Security group should restrict all traffic to reduce resource exposure.</p>	Low

Next steps

For related information, see the following:

- [Connect your AWS accounts to Microsoft Defender for Cloud](#)

- What are security policies, initiatives, and recommendations?
- Review your security recommendations

Security alerts and incidents

Article • 05/29/2023

This article describes security alerts and notifications in Microsoft Defender for Cloud.

What are security alerts?

Security alerts are the notifications generated by Defender for Cloud's workload protection plans when threats are identified in your Azure, hybrid, or multicloud environments.

- Security alerts are triggered by advanced detections available when you enable [Defender plans](#) for specific resource types.
- Each alert provides details of affected resources, issues, and remediation steps.
- Defender for Cloud classifies alerts and prioritizes them by severity.
- Alerts are displayed in the portal for 90 days, even if the resource related to the alert was deleted during that time. This is because the alert might indicate a potential breach to your organization that needs to be further investigated.
- Alerts can be exported to CSV format.
- Alerts can also be streamed directly to a Security Information and Event Management (SIEM) such as Microsoft Sentinel, Security Orchestration Automated Response (SOAR), or IT Service Management (ITSM) solution.
- Defender for Cloud leverages the [MITRE Attack Matrix](#) to associate alerts with their perceived intent, helping formalize security domain knowledge.

How are alerts classified?

Alerts have a severity level assigned to help prioritize how to attend to each alert.

Severity is based on:

- The specific trigger
- The confidence level that there was malicious intent behind the activity that led to the alert

Severity	Recommended response
High	There is a high probability that your resource is compromised. You should look into it right away. Defender for Cloud has high confidence in both the malicious intent and in the findings used to issue the alert. For example, an alert that detects the execution of a known malicious tool such as Mimikatz, a common tool used for credential theft.

Severity	Recommended response
Medium	This is probably a suspicious activity might indicate that a resource is compromised. Defender for Cloud's confidence in the analytic or finding is medium and the confidence of the malicious intent is medium to high. These would usually be machine learning or anomaly based detections, for example a sign-in attempt from an unusual location.
Low	This might be a benign positive or a blocked attack. Defender for Cloud isn't confident enough that the intent is malicious and the activity might be innocent. For example, log clear is an action that might happen when an attacker tries to hide their tracks, but in many cases is a routine operation performed by admins. Defender for Cloud doesn't usually tell you when attacks were blocked, unless it's an interesting case that we suggest you look into.
Informational	An incident is typically made up of a number of alerts, some of which might appear on their own to be only informational, but in the context of the other alerts might be worthy of a closer look.

What are security incidents?

A security incident is a collection of related alerts.

Incidents provide you with a single view of an attack and its related alerts, so that you can quickly understand the actions an attacker took, and the affected resources.

As the breath of threat coverage grows, so does the need to detect even the slightest compromise. It's challenging for security analysts to triage different alerts and identify an actual attack. By correlating alerts and low fidelity signals into security incidents, Defender for Cloud helps analysts cope with this alert fatigue.

In the cloud, attacks can occur across different tenants, Defender for Cloud can combine AI algorithms to analyze attack sequences that are reported on each Azure subscription. This technique identifies the attack sequences as prevalent alert patterns, instead of just being incidentally associated with each other.

During an investigation of an incident, analysts often need extra context to reach a verdict about the nature of the threat and how to mitigate it. For example, even when a network anomaly is detected, without understanding what else is happening on the network or with regard to the targeted resource, it's difficult to understand what actions to take next. To help, a security incident can include artifacts, related events, and information. The additional information available for security incidents varies, depending on the type of threat detected and the configuration of your environment.

Correlating alerts into incidents

Defender for Cloud correlates alerts and contextual signals into incidents.

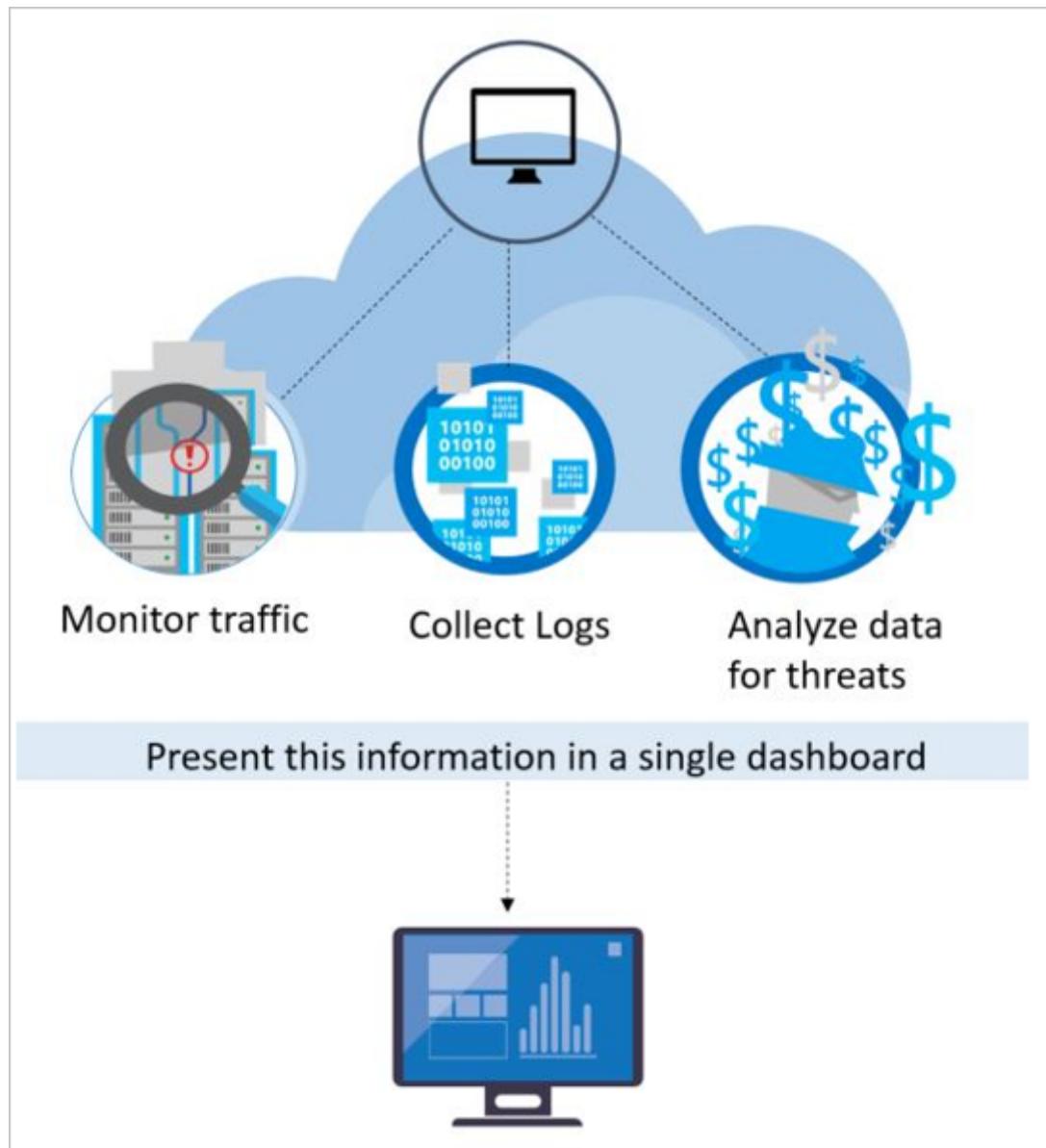
- Correlation looks at different signals across resources and combines security knowledge and AI to analyze alerts, discovering new attack patterns as they occur.
- By using the information gathered for each step of an attack, Defender for Cloud can also rule out activity that appears to be steps of an attack, but actually isn't.

💡 Tip

In the [incidents reference](#), review the list of security incident that can be produced by incident correlation.

How does Defender for Cloud detect threats?

To detect real threats and reduce false positives, Defender for Cloud monitors resources, collects, and analyzes data for threats, often correlating data from multiple sources.



Microsoft initiatives

Microsoft Defender for Cloud benefits from having security research and data science teams throughout Microsoft who continuously monitor for changes in the threat landscape. This includes the following initiatives:

- **Microsoft security specialists:** Ongoing engagement with teams across Microsoft that work in specialized security fields, like forensics and web attack detection.
- **Microsoft security research:** Our researchers are constantly on the lookout for threats. Because of our global presence in the cloud and on-premises, we have access to an expansive set of telemetry. The wide-reaching and diverse collection of datasets enables us to discover new attack patterns and trends across our on-premises consumer and enterprise products, as well as our online services. As a result, Defender for Cloud can rapidly update its detection algorithms as attackers release new and increasingly sophisticated exploits. This approach helps you keep pace with a fast moving threat environment.

- **Threat intelligence monitoring:** Threat intelligence includes mechanisms, indicators, implications, and actionable advice about existing or emerging threats. This information is shared in the security community and Microsoft continuously monitors threat intelligence feeds from internal and external sources.
- **Signal sharing:** Insights from security teams across Microsoft's broad portfolio of cloud and on-premises services, servers, and client endpoint devices are shared and analyzed.
- **Detection tuning:** Algorithms are run against real customer data sets and security researchers work with customers to validate the results. True and false positives are used to refine machine learning algorithms.

These combined efforts culminate in new and improved detections, which you can benefit from instantly – there's no action for you to take.

Security analytics

Defender for Cloud employs advanced security analytics, which go far beyond signature-based approaches. Breakthroughs in big data and [machine learning](#) technologies are leveraged to evaluate events across the entire cloud fabric – detecting threats that would be impossible to identify using manual approaches and predicting the evolution of attacks. These security analytics include:

Integrated threat intelligence

Microsoft has an immense amount of global threat intelligence. Telemetry flows in from multiple sources, such as Azure, Microsoft 365, Microsoft CRM online, Microsoft Dynamics AX, outlook.com, MSN.com, the Microsoft Digital Crimes Unit (DCU), and Microsoft Security Response Center (MSRC). Researchers also receive threat intelligence information that is shared among major cloud service providers and feeds from other third parties. Microsoft Defender for Cloud can use this information to alert you to threats from known bad actors.

Behavioral analytics

Behavioral analytics is a technique that analyzes and compares data to a collection of known patterns. However, these patterns are not simple signatures. They are determined through complex machine learning algorithms that are applied to massive datasets. They are also determined through careful analysis of malicious behaviors by expert analysts. Microsoft Defender for Cloud can use behavioral analytics to identify

compromised resources based on analysis of virtual machine logs, virtual network device logs, fabric logs, and other sources.

Anomaly detection

Defender for Cloud also uses anomaly detection to identify threats. In contrast to behavioral analytics that depends on known patterns derived from large data sets, anomaly detection is more "personalized" and focuses on baselines that are specific to your deployments. Machine learning is applied to determine normal activity for your deployments and then rules are generated to define outlier conditions that could represent a security event.

Exporting alerts

You have a range of options for viewing your alerts outside of Defender for Cloud, including:

- **Download CSV report** on the alerts dashboard provides a one-time export to CSV.
- **Continuous export** from Environment settings allows you to configure streams of security alerts and recommendations to Log Analytics workspaces and Event Hubs. [Learn more](#).
- **Microsoft Sentinel connector** streams security alerts from Microsoft Defender for Cloud into Microsoft Sentinel. [Learn more](#).

Learn about [streaming alerts to a SIEM, SOAR, or IT Service Management solution](#) and how to [continuously export data](#).

Next steps

In this article, you learned about the different types of alerts available in Defender for Cloud. For more information, see:

- [Security alerts in Azure Activity log](#) - In addition to being available in the Azure portal or programmatically, Security alerts and incidents are audited as events in Azure Activity Log
- [Reference table of Defender for Cloud alerts](#)
- [Respond to security alerts](#)
- Learn how to [manage security incidents in Defender for Cloud](#).

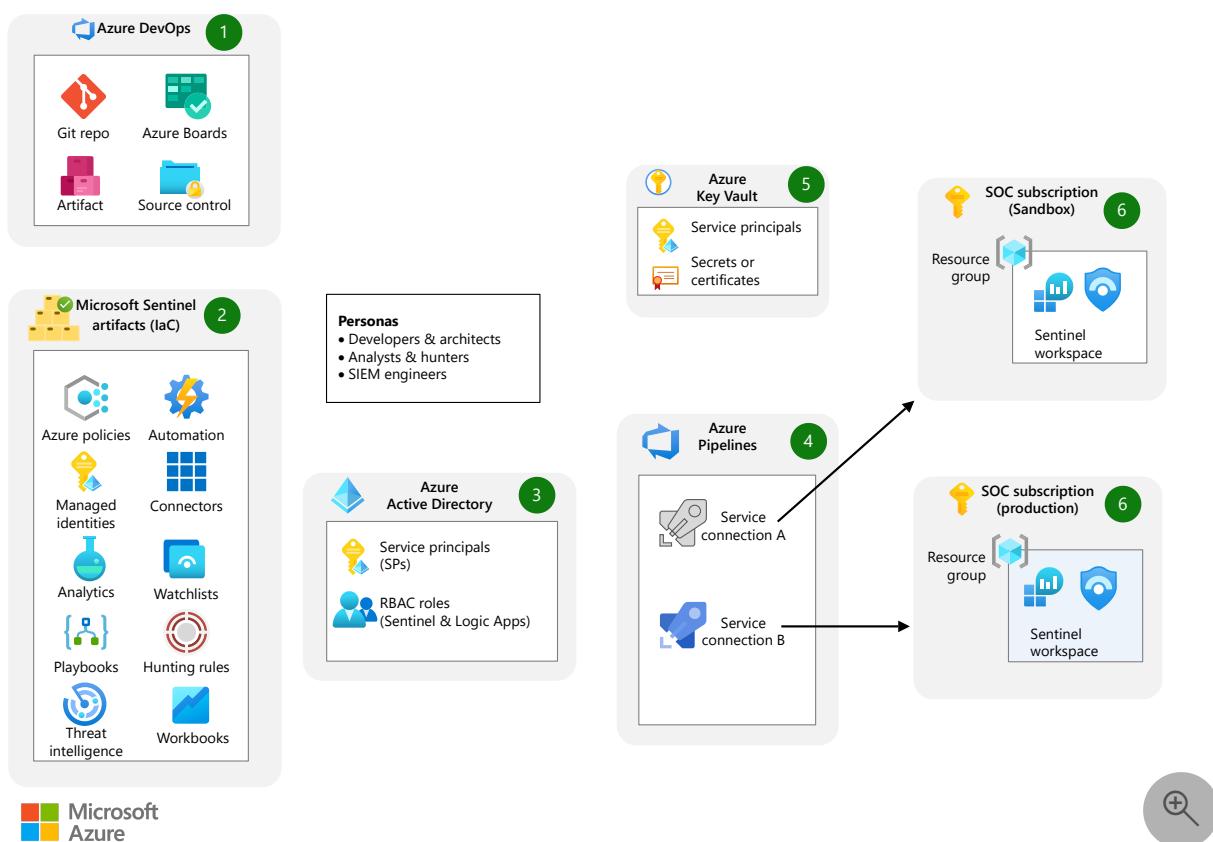
Automate Sentinel integration with Azure DevOps

Microsoft Entra ID Azure DevOps Azure Key Vault Microsoft Defender for Cloud Microsoft Sentinel

This article describes how to automate Microsoft Sentinel integration and deployment operations with Azure DevOps. You implement Azure DevOps by using Microsoft Sentinel capabilities to help secure your deployment. You then use a DevSecOps framework to manage and deploy Microsoft Sentinel artifacts at scale.

Architecture

The following diagram shows an Azure DevOps and Microsoft Sentinel IaC setup.



Download a [Visio file](#) of this architecture.

Dataflow

1. The scrum master and product management use Azure DevOps to define epics, user stories, and product backlog items as part of the project backlog.

- The scrum master and product management use Azure Boards to create the backlog, schedule work in sprints, review the project board, create the repository structure, and set security rules like approval workflows and branches.
- The Azure Git repository stores the scripts and the permits to manage Microsoft Sentinel artifacts in the infrastructure as code.
- Artifacts and source control maintain the extensions and update packages or components of the DevSecOps workflow that are used in the solution, such as Azure Resource Manager Template Toolkit and PowerShell Pester.

2. Microsoft Sentinel artifacts:

- Policies. SIEM engineers use Azure policies in the reference architecture, to configure and scale the diagnostic settings of the Azure services. The policies help automate deployment of the Microsoft Sentinel data connectors, such as Azure Key Vault. The policies are dependent on the OMSIntegration API.
- Connectors. Microsoft Sentinel uses logical connectors, the Azure Data Connectors, to ingest security data, as in audits or metrics, from supported data sources, such as Microsoft Entra ID, Azure resources, Microsoft Defender, or third-party solutions. The main list of data connectors is managed by the SecurityInsights API. Others rely on the OMSIntegration API and are managed with the Azure Policy diagnostic settings.
- Managed identity. Microsoft Sentinel uses managed identity to act on behalf of the Managed service identity (MSI) while interacting with playbooks, logic apps, or automation runbooks and the key vault.
- Automation. SOC teams use automation during investigations. SOC teams run digital forensics data acquisition procedures with Azure Automation, such as Azure virtual machine (VM) chain of custody or eDiscovery (Premium) for Microsoft Defender.
- Analytics. SOC analysts or threat hunters use built-in or custom analytics rules to analyze and correlate data in Microsoft Sentinel or to trigger playbooks if a threat and incident are identified.
- Playbooks. Logic apps run the SecOps repeatable actions, such as assigning an incident, updating an incident, or taking remediation actions, like isolating or containing a VM, revoking a token, or resetting a user password.
- Threat hunting. Threat hunters use proactive threat hunting capabilities that can be coupled with Jupyter notebooks for advanced use cases, such as data processing, data manipulation, data visualization, machine learning, or deep learning.
- Workbooks. SIEM engineers use Workbooks dashboards to visualize trends and statistics and to view the status of a Microsoft Sentinel instance and its subcomponents.
- Threat intelligence. A specific data connector that fuses threat intelligence platforms feeds into Microsoft Sentinel. Two connectivity methods are supported:

TAXII and Graph API. Both methods serve as *indicators*, or threat intelligence indicators, in security APIs.

3. Microsoft Entra ID. Identity and access management capabilities are delivered to components that are used in the reference architecture, such as managed identities, service principals, Azure role-based access controls (RBACs) for Microsoft Sentinel, logic apps, and automation runbooks.
4. Azure Pipelines. DevOps engineers use pipelines to create service connections for managing the different Azure subscriptions like the sandbox and production environments with continuous integration and continuous delivery (CI/CD) pipelines. We recommend using approval workflows to prevent unexpected deployments and separated service principals if you target multiple subscriptions per Azure environment.
5. Azure Key Vault. SOC engineers use the key vault to securely store service principal secrets and certificates. This component of the architecture helps enforce the DevSecOps principle of *no secrets in code* when used by Azure Pipeline service connections.
6. Azure subscription. The SOC teams use two instances of Microsoft Sentinel in this reference architecture, separated within two logical Azure subscriptions to simulate production and sandbox environments. You can scale for your needs with other environments, such as testing, dev, preproduction, and so on.

Dataflow example

1. An administrator adds, updates, or deletes an entry in their fork of the Microsoft 365 configuration file.
2. The administrator commits and syncs the changes to their forked repository.
3. The administrator then creates a pull request (PR) to merge the changes to the main repository.
4. The build pipeline runs on the PR.

Components

- [Microsoft Entra ID](#) is a multi-tenant, cloud-based service to manage your identity and access controls.
- [Azure DevOps](#) is a cloud service to collaborate on code, build and deploy apps, or plan and track your work.
- [Azure Key Vault](#) is a cloud service for securely storing and accessing secrets. A secret is anything that you want to tightly control access to, such as API keys, passwords, certificates, or cryptographic keys.
- [Azure Policy](#) is a service to create, assign, and manage policy definitions in your Azure environment.
- [Microsoft Sentinel](#) is a scalable, cloud-native, SIEM and security orchestration, automation, and response (SOAR) solution.

- [Azure Automation](#) is a service for simplifying cloud management through process automation. Use Azure Automation to automate long-running, manual, error-prone, and frequently repeated tasks. Automation helps improve reliability, efficiency, and time to value for your company.

Scenario details

Security operations center (SOC) teams sometimes experience challenges when they integrate Microsoft Sentinel with Azure DevOps. The process involves many steps, and the setup can take days and involve repetition. You can automate this part of the development.

To modernize for the cloud, engineers must constantly learn new skills and techniques for securing and protecting vital business assets. Engineers must build robust and scalable solutions that keep pace with the changing security landscape and with business needs. A security solution must be flexible, agile, and carefully planned from the earliest stages of development. This early-planning methodology is known as *shift-left*.

This article describes how to automate Microsoft Sentinel integration and deployment operations with Azure DevOps. You can expand the solution for complex organizations that have multiple entities, subscriptions, and various operating models. Some of the operating models supported by this solution include local SOC, global SOC, cloud service provider (CSP), and managed security service provider (MSSP).

This article is intended for the following audiences:

- SOC specialists, like analysts and threat hunters
- Security information and event management (SIEM) engineers
- Cybersecurity architects
- Developers

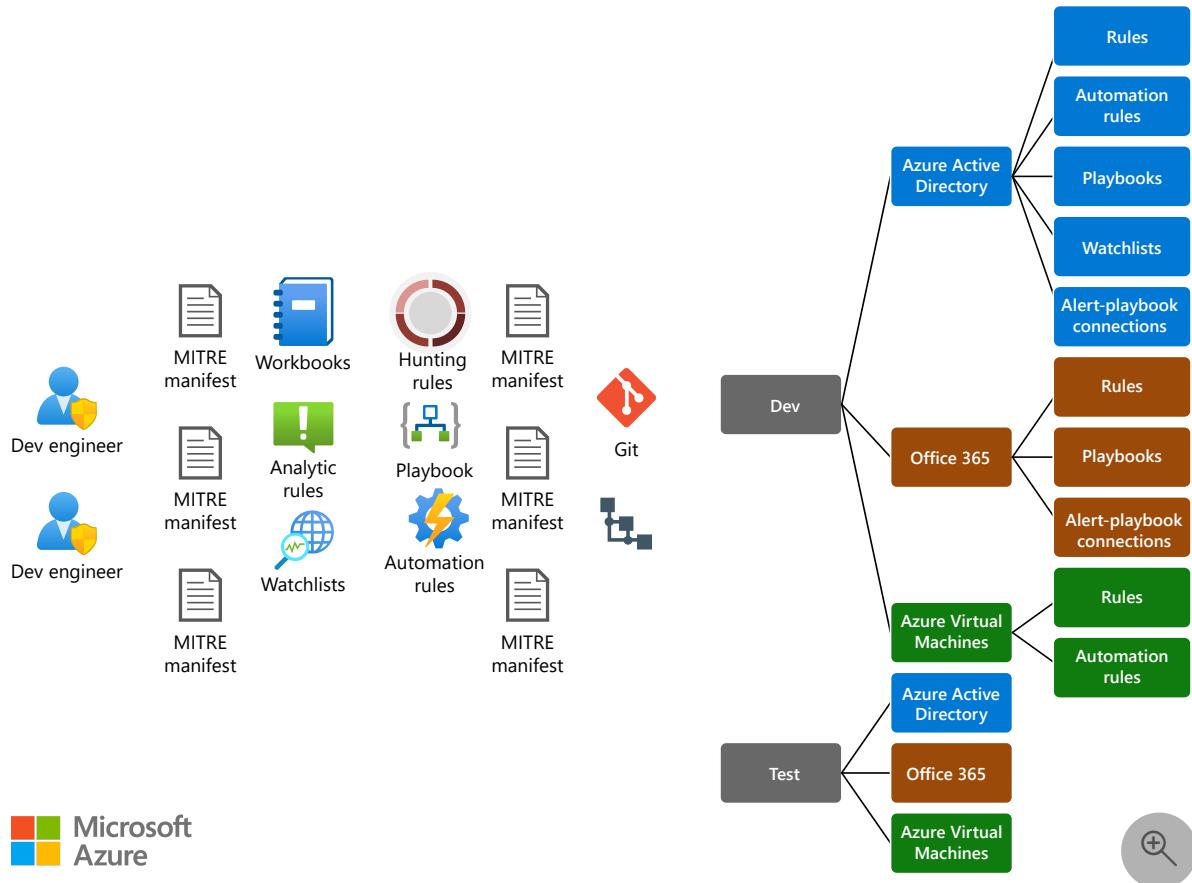
Potential use cases

Following are the typical use cases for this architecture:

- Rapid prototyping and proof of concept. This solution is ideal for security organizations and SOC teams that want to improve cloud threat coverage or modernize their SIEM infrastructure with infrastructure as code (IaC) and Microsoft Sentinel.
- Microsoft Sentinel as a service. This development framework integrates service lifecycle management principles. These principles suit simple or complex teams like MSSPs who run repeatable, standardized actions across multiple customer tenants while combining the power of Azure DevOps and Azure Lighthouse. For example, a team that needs to publish Microsoft Sentinel use cases for a new threat actor or ongoing campaign could use this solution.

- Building SOC use cases for threat detection. Many groups and threat intelligence platforms rely on MITRE Att&ck content and taxonomy to analyze their security posture against advanced tradecraft or techniques and tactics procedures. The solution defines a structured approach for developing threat detection engineering practices by incorporating MITRE Att&ck terminology within Microsoft Sentinel artifacts development.

The following illustration shows a MITRE Att&ck cloud scenario.



[Download a Visio file](#) of this architecture.

Threat definition attack scenarios based on MITRE

This table shows you the terms, definitions, and details of important aspects of attack scenarios.

[Expand table](#)

Data item	Description	Microsoft Sentinel artifacts
Title	Descriptive name for the attack scenario, based on attack vector characteristics or technique descriptions.	MITRE manifest
MITRE ATT&CK tactics	MITRE ATT&CK tactics related to attack scenario	MITRE manifest
MITRE ATT&CK techniques	MITRE ATT&CK techniques, including the technique or sub-technique ID, related to the attack scenario.	MITRE manifest
Data connector sources	Source of information collected by a sensor or logging system that might be used to collect information relevant to identifying the action being performed, sequence of actions, or the results of those actions by an adversary.	Microsoft Sentinel data connector or Custom log source
Description	Information about the technique, what it is, what it's typically used for, how an adversary can take advantage of it, and variations on how it could be used. Includes references to authoritative articles describing technical information related to the technique as well as in the wild use references as appropriate.	
Detection	High-level analytic process, sensors, data, and detection strategies useful in identifying a technique that's been used by an adversary. This section informs those responsible for detecting adversary behavior, such as network defenders, so they can take an action such as writing an analytic or deploying a sensor. There should be enough information and references to point toward useful defensive methodologies. Detection might not always be possible for a certain technique and should be documented as such.	Analytics threat hunting
Mitigation	Configurations, tools, or processes that prevent a technique from working or having the desired outcome for an adversary. This section informs those responsible for mitigating against adversaries, such as network defenders or policymakers, to let them take an action such as changing a policy or deploying a tool. Mitigation might	

Data item	Description	Microsoft Sentinel artifacts
	not always be possible for a given technique and should be documented as such.	
Mitigation	Configurations, tools, or processes that prevent a technique from working or having the desired outcome for an adversary. This section describes how to lessen the effects of adversary attacks for network defenders or policymakers. It covers steps for changing a policy or deploying a tool. Mitigation might not always be possible for a certain technique and should be documented as such.	Playbooks, automation runbooks

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, a set of guiding tenets that you can use to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

With security, in general terms, automation increases operations efficiency while saving time for more complex use cases, such as threat detection engineering, threat intelligence, SOC, and SOAR use cases. DevOps teams need to know where they can use IaC securely in the context of Microsoft Sentinel CI/CD. This process introduces the use of specific identities that are used by non-human accounts in Microsoft Entra ID called [service principals](#) and [managed identities](#).

The following table summarizes security considerations for service principals and the main use cases that are covered by Microsoft Sentinel and Azure DevOps release pipelines.

[] [Expand table](#)

Use case	Requirements (least privilege)	Role assignment duration	Permission scope	Trustee	Security considerations
Enable Microsoft Sentinel connectors	<p>Security administrator**</p> <p>Microsoft Sentinel contributor</p> <p>Reader</p>	<p>JIT (one-time activation)</p> <p>On purpose (every time a new subscription and connector deploys)</p>	Tenant	SPN	<p>Use the key vault to store service principal name (SPN) secrets and certificate.</p> <p>Enable SPN auditing.</p> <p>Periodically, review the permission assignment (Azure Privileged Identity Management for SPN) or suspicious activity for SPN.</p> <p>Use Microsoft Entra certificate authorities and multifactor authentication (when supported) for privileged accounts.</p> <p>Use Microsoft Entra Custom Roles for more granularity.</p>
Deploy Microsoft Sentinel artifacts, such as workbooks, analytics, rules,	Microsoft Sentinel Contributor	Permanent	Microsoft Sentinel's Workspace or Resource Group	SPN	Use Azure DevOps (ADO) workflow approval and checks to secure pipeline deployment with this SPN.

Use case threat hunting queries, notebooks, and playbooks	Requirements (least privilege)	Role assignment duration	Permission scope	Trustee	Security considerations
Assign a policy to configure log streaming features to Microsoft Sentinel	Resource Policy Contributor **	On purpose (every time a new subscription and connector deploys)	All subscriptions to be monitored	SPN	Use Microsoft Entra ID, CA, and MFA, when supported, for privileged accounts.

* Only concerns Microsoft Entra diagnostics settings.

** Specific connectors need additional permissions like "security administrator" or "resource policy contributor" to allow streaming data to Microsoft Sentinel workspace, Microsoft Entra ID, Microsoft 365 or Microsoft Defender, and Platform as a service (PaaS) resources like Azure Key Vault.

Privileged access model

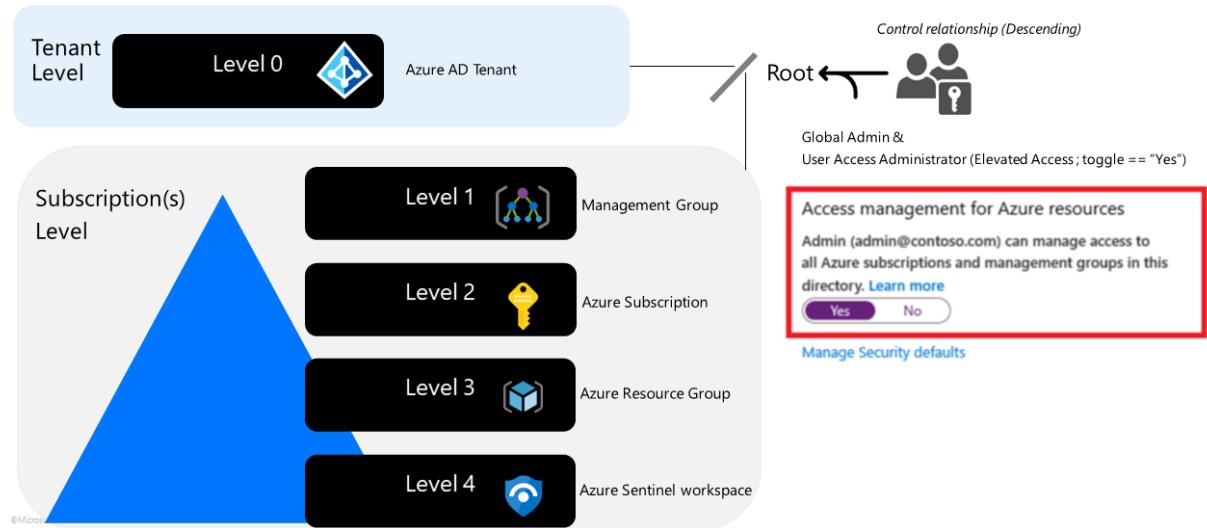
We recommend adopting a privileged access model strategy to rapidly lower the risks to your company from high-impact and high-likelihood attacks on privileged access. In the case of automatic processes in a DevOps model, base the identity on [service principal](#) identities.

Privileged access should be the top security priority at every company. Any compromise of these identities creates highly negative impacts. Privileged identities have access to business-critical assets, which nearly always causes major impacts when attackers compromise these accounts.

Security of privileged access is critically important because it's foundational to all other security assurances. An attacker in control of your privileged accounts can undermine all other security assurances.

For that reason, we recommend logically spreading the service principals into different levels or tiers by following a minimum privilege principle. The following illustration shows how to classify the service principals, depending on the type of access and where the access is required.

Azure Sentinel – Layered Architecture



Level 0 service principals

Level 0 service principals have the highest level of permissions. These service principals entitle someone to perform tenant-wide or root management group administration tasks as a global administrator.

For security reasons and manageability, we recommend that you have only one service principal for this level. The permissions for this service principal persist, so we recommend that you grant only the minimum permissions that are required and keep the account monitored and secured.

Store the secret or certificate for this account securely in Azure Key Vault. We strongly recommended that you locate the key vault in a dedicated administrative subscription if possible.

Level 1 service principals

Level 1 service principals are elevated permissions that are limited and scoped to management groups at the business organization level. These service principals entitle someone to create subscriptions under the management group that's in scope.

For security reasons and manageability, we recommend that you have only one service principal for this level. The permissions for this service principal persist, so we highly recommended that you grant only the minimum permissions that are required and keep the account monitored and secured.

Store the secret or certificate for this account securely in Azure Key Vault. We strongly recommended that you locate the key vault in a dedicated administrative subscription if

possible.

Level 2 service principals

Level 2 service principals are limited to the subscription level. These service principals entitle someone to perform administrative tasks under a subscription, acting as the subscription owner.

For security reasons and manageability, we recommend that you have only one service principal for this level. The permissions for this service principal persist, so we highly recommend that you grant only the minimum permissions that are required and keep the account monitored and secured.

Store the secret or certificate for this account securely in Azure Key Vault. We strongly recommended that you locate the key vault in a dedicated administrative resource group.

Level 3 service principals

Level 3 service principals are limited to the Workload Administrator. In a typical scenario, every workload is contained inside the same resource group. This structure limits the service principal permissions to just this resource group.

For security reasons and manageability, we recommend that you have only one service principal per workload. The permissions for this service principal persist, so we highly recommend that you grant only the minimum permissions that are required and keep the account monitored and secured.

Store the secret or certificate for this account securely in Azure Key Vault. We strongly recommended that you locate the key vault in a dedicated administrative resource group.

Level 4 service principals

Level 4 service principals have the most limited permissions. These service principals entitle someone to perform administrative tasks that are limited to one resource.

We recommended using managed identities where possible. In the case of non-managed identities, store the secret or certificate securely in Azure Key Vault where Level 3 secrets are stored.

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

Microsoft Sentinel solutions are composed of three blocks, which ensure complete and successful operations.

The first block is the environment definition, which makes up the essential architecture elements. Your main concern with this block is to consider the number of production and non-production environments to be deployed, and then ensure the setup is homogeneous in all cases.

The second block is the Microsoft Sentinel connector deployment, where you consider the kind of connectors that are required by your team and the security requirements to enable them.

The third block is the Microsoft Sentinel artifacts lifecycle management, which covers coding, deployment, and use or destruction of the components. For example, this block contains the analytic rules, playbooks, workbooks, threat hunting, and so on.

Consider these dependencies between artifacts:

- Automation rules that are defined in an analytics rule
- Workbooks or analytics that require a new data source or connector
- Managing the updates of existing components
 - How to version your artifacts
 - How to identify, test, and deploy an updated or entirely new analytics rule

Build, test, and deploy infrastructure

In managing Microsoft Sentinel solutions and DevOps, it's important to consider the connectivity and security aspects of your enterprise architecture.

Azure DevOps can use Microsoft-hosted agents or self-hosted agents for build, test, and deploy activities. Depending on your company's requirements, you can use Microsoft-hosted, self-hosted, or a combination of both models.

- Microsoft-hosted agents. This option is the fastest way to work with Azure DevOps agents, because it's a shared infrastructure for your entire organization. For more information on using Microsoft-hosted agents in your pipeline, see [Microsoft-hosted agents](#). Microsoft-hosted agents can work in hybrid-networking environments, granting access for the IP ranges. To download the IP ranges that these agents grant access to, see [Azure IP Ranges and Service Tags – Public Cloud](#).
- Self-hosted agents. This option gives you dedicated resources and more control when installing dependent software for your builds and deployments. Self-hosted agents can work over VMs, scale sets, and containers on Azure. For more information on self-hosted agents, see [Azure Pipelines agents](#).

GitHub runners

GitHub can use GitHub-hosted runners or self-hosted runners for activities that are related to building, testing, and deploying. Depending on your company's needs, you can use GitHub-hosted, self-hosted, or a combination of both models.

GitHub-hosted runners

This option is the fastest way to work with GitHub workflows, since it's a shared infrastructure for an entire organization. For more information, see [About GitHub-hosted runners](#).

GitHub-hosted agents work in hybrid-networking environments, according to certain network requirements. For more information on the network requirements, see [Supported runners and hardware resources](#).

Self-hosted runners

This option gives your company a dedicated resources infrastructure. Self-hosted runners work over VMs and containers on Azure and support auto-scaling.

Considerations for choosing runners

When choosing options for the agents and runners in your Microsoft Sentinel solution, consider the following needs:

- Does your company need dedicated resources for running processes on your Microsoft Sentinel environments?
- Do you want to isolate resources for production environment DevOps activities from the rest of the environments?
- Do you need to test certain cases that require access to critical resources or resources that are available only on an internal network?

Orchestration and automation of release processes

You can set up the deployment process with Azure DevOps or GitHub. Azure DevOps supports using a YAML pipeline or a release pipeline. For more information on using a YAML pipeline in Azure DevOps, see [Use Azure Pipelines](#). For more information on using a release pipeline in Azure DevOps, see [Release pipelines](#). For more information on using GitHub with GitHub Actions, see [Understanding GitHub Actions](#).

Azure DevOps

You can do the following deployment activities in an Azure DevOps deployment.

- Use a YAML pipeline to automatically trigger PR approvals or run on demand.
- Manage service connections for different environments by using Azure DevOps groups.
- On your critical environments, set up deployment approvals by using the service connection feature and Azure DevOps groups to assign specific user permissions in

your team.

GitHub

You can do the following deployment activities in a GitHub deployment.

- Use GitHub to create PRs or deployment activities.
- Manage service principal credentials by using GitHub Secrets.
- Integrate deployment approval through the workflow that's associated with GitHub.

Automatic deployment with Microsoft Sentinel infrastructure

You can deploy one or more Microsoft Sentinel environments, depending on your enterprise architecture:

- Organizations that need multiple instances on their production environment can set up different subscriptions on the same tenant for each geographical location.
- A centralized instance on the production environment provides access to one or more organizations on the same tenant.
- Groups that need multiple environments like production, preproduction, integration, and so on can create and destroy them as needed.

Physical versus logical environment definitions

You have two choices in setting up your environment definitions, physical or logical. Both have different options and advantages:

- Physical definition - The elements of the Microsoft Sentinel architecture are defined with the following options for infrastructure as code (IaC):
 - Bicep templates
 - Azure Resource Manager templates (ARM templates)
 - Terraform
- Logical definition - This acts as an abstraction layer for setting up different teams in the group and defining their environments. The definition is set in the deployment pipeline and workflows as input for the build environment by using the physical infrastructure layer.

Consider these points when you define your logical environments:

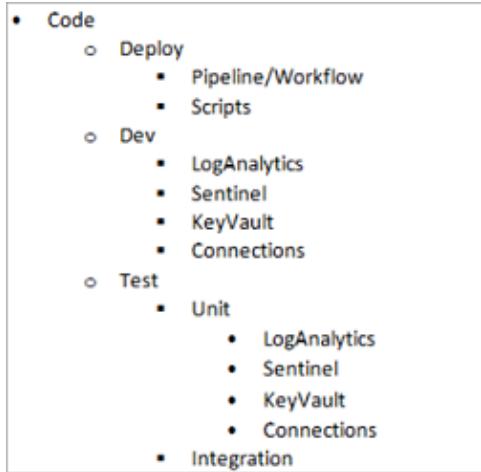
- Naming conventions
- Environment identifications
- Connectors and configurations

Code repository

Given the environment approaches that are shown in the previous section, consider the following GitHub code repository organization.

Physical definition - Based on IaC options, think about an approach that uses individual module definitions that are linked in the main deployment definition.

The following example shows how your code might be organized.



Restrict access to this repository to the team that defines the architecture at the physical level, ensuring a homogeneous definition in the enterprise architecture.

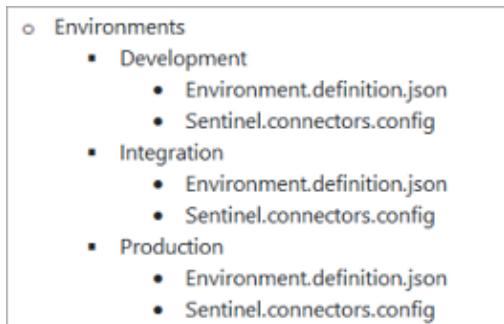
You can adapt the branching and merging strategy to the deployment strategy for each organization. If your team needs to start with the definition, see [Adopt a Git branching strategy](#).

For more information on ARM templates, see [Using linked and nested templates when deploying Azure resources](#).

For more information on setting up Bicep environments, see [Install Bicep tools](#). For more information on GitHub, see [GitHub flow](#).

Logical definitions define a company's environments. The Git repository gathers the different definitions for a company.

The following example shows how your code might be organized.



The repository reflects the PR actions that are made by different teams. Multiple environments are defined by different teams and approved by the company's owners or

approvers.

The privilege level for running an environment deployment is Level 2. This level ensures that the resource group and the resources are created for the environment with the necessary security and privacy. This level also sets the user permissions on allowed actions in the production environments, production and preproduction.

Organizations that want environments on demand for testing and development and the ability to then destroy the environments after finishing their testing, can implement an Azure DevOps pipeline or GitHub actions. They can set scheduled triggers to destroy the environments as needed by using Azure DevOps events or GitHub actions.

Microsoft Sentinel connectors automatic configuration

Microsoft Sentinel connectors are an essential part of the solution that supports connecting with different elements in the enterprise architecture landscape, like Microsoft Entra ID, Microsoft 365, Microsoft Defender, threat intelligence platform solutions, and so on.

When you define an environment, you can use the connectors configuration to set up environments with homogeneous configurations.

Enabling connectors as part of the DevOps model must be supported by the service principal level model. This focus ensures the right level of permissions as shown in the following table.

[Expand table](#)

Connector scenario	Privilege access model level	Azure least privilege	Requires workflow approval
Microsoft Entra ID	Level 0	global admin or security admin	Recommended
Microsoft Entra ID Protection	Level 0	global admin or security admin	Recommended
Microsoft Defender for Identity	Level 0	global Admin or security admin	Recommended
Microsoft Office 365	Level 0	global admin or security admin	Recommended
Microsoft Cloud App Security	Level 0	global admin or security admin	Recommended

Connector scenario	Privilege access model level	Azure least privilege	Requires workflow approval
Microsoft Defender XDR	Level 0	global admin or security admin	Recommended
Microsoft Defender for IOT	Level 2	Contributor	Recommended
Microsoft Defender for Cloud	Level 2	Security Reader	Optional
Azure Activity	Level 2	Subscription Reader	Optional
Threat Intelligence Platforms	Level 0	global admin or security admin	Recommended
Security Events	Level 4	None	Optional
Syslog	Level 4	None	Optional
DNS (preview)	Level 4	None	Optional
Windows Firewall	Level 4	None	Optional
Windows Security Events via AMA	Level 4	None	Optional

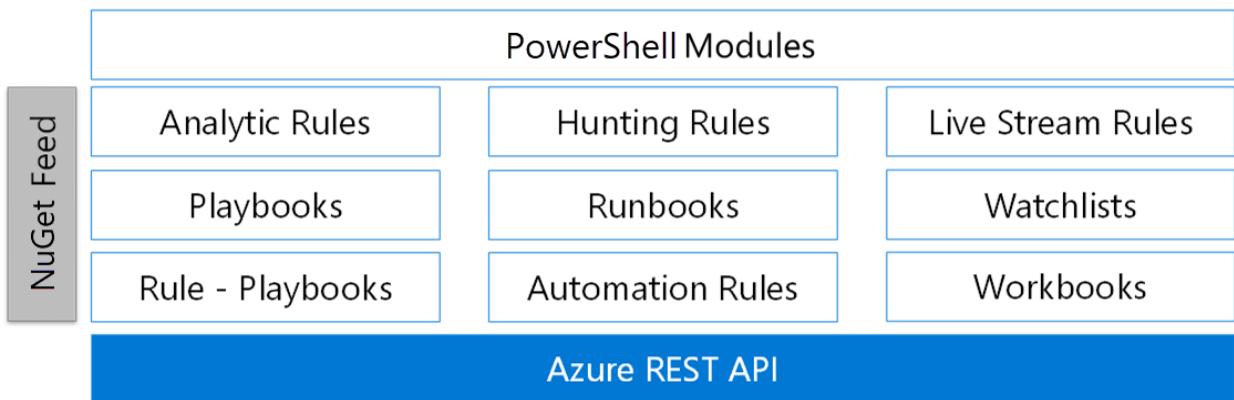
Microsoft Sentinel artifacts deployment

In the implementation of Microsoft Sentinel artifacts, DevOps gains greater relevance, because each company creates multiple artifacts for preventing and remediating attacks.

Implementing the artifacts can be the responsibility of one team or multiple teams. Automatic build and artifacts deployment is often the most common process requirement and determines the approach and conditions for your agents and runners.

Deploying and managing Microsoft Sentinel artifacts requires using the Microsoft Sentinel REST API. For more information, see [Microsoft Sentinel REST API](#). The following diagram shows an Azure DevOps pipeline on an Azure REST API stack.

DevOps Pipelines



You can also implement your repository by using PowerShell.

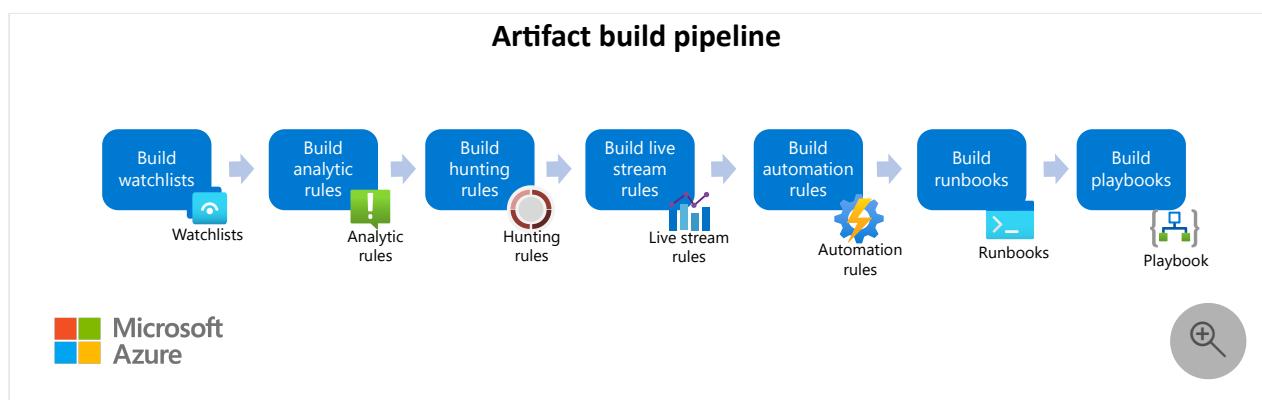
If your team uses MITRE, consider classifying the different artifacts and specifying the tactics and techniques for each one. Be sure you include a corresponding metadata file for each artifact type.

For example, if you're creating a new playbook by using an Azure ARM template and the file name is *Playbook.arm.json*, you add a JSON file named *Playbook.arm.mitre.json*. The metadata for this file then includes the CSV, JSON, or YAML formats that correspond to the MITRE tactics or techniques that you're using.

By following this practice, your team can evaluate your MITRE coverage based on the jobs that are done during setup for the different artifact types that you use.

Build artifacts

The objective of your build process is to ensure that you generate the highest quality artifacts. The following diagram shows some of the build process actions that you can take.



Download a [Visio file](#) of this architecture.

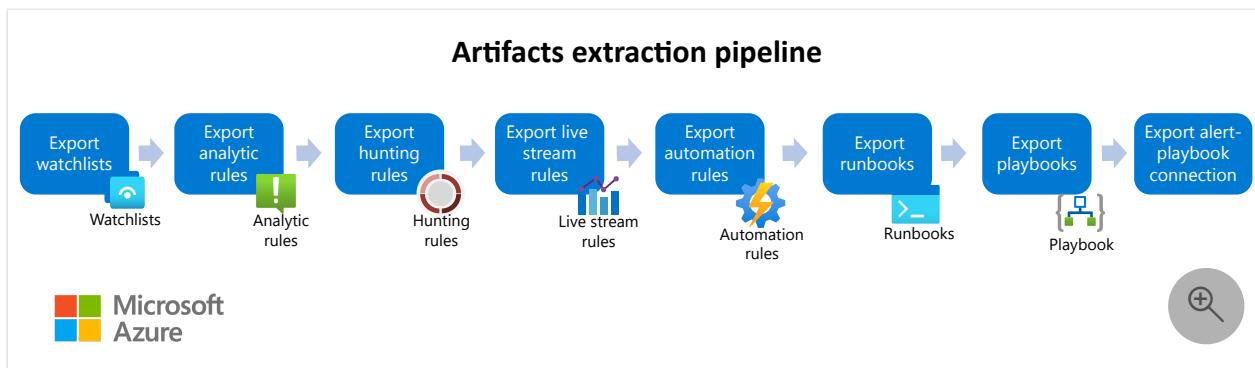
- You can base your artifact definition on a descriptive schema in JSON or YAML format and then validate the schema to avoid syntax errors.
 - Validate your ARM templates by using [ARM template test toolkit](#).

- Validate your YAML and JSON files for custom models by using PowerShell.
- Validate your watchlist settings and be sure that the classless inter-domain routing (CIDR) records that you define follow the correct schema, for example, 10.1.0.0/16.
- Use keyword query language (KQL) queries, which you can validate at the level of the syntax, for analytic rules, hunting rules, and live stream rules, which you can validate at the level of the syntax.
- Make the [KQL local validation](#) one option.
- Integrate the [KQL inline validation](#) tool in the DevOps pipeline.
- If you're implementing logic that's based on PowerShell for Azure Automation, you can include syntax validation and unit testing by using the following elements:
 - [Pester](#)
 - [PSScriptAnalyzer](#)
- Generate the MITRE manifest metadata report based on the metadata files that are included with the artifacts.

Export artifacts

Usually, multiple teams work over several Microsoft Sentinel instances to generate necessary artifacts and validate them. With the goal of reusing existing artifacts, your company can set up automatic processes for getting the artifact definitions from existing environments. Automation can also supply information on any artifacts that are created by different development teams during setup.

The following diagram shows an example artifact extraction process.



[Download a Visio file](#) of this architecture.

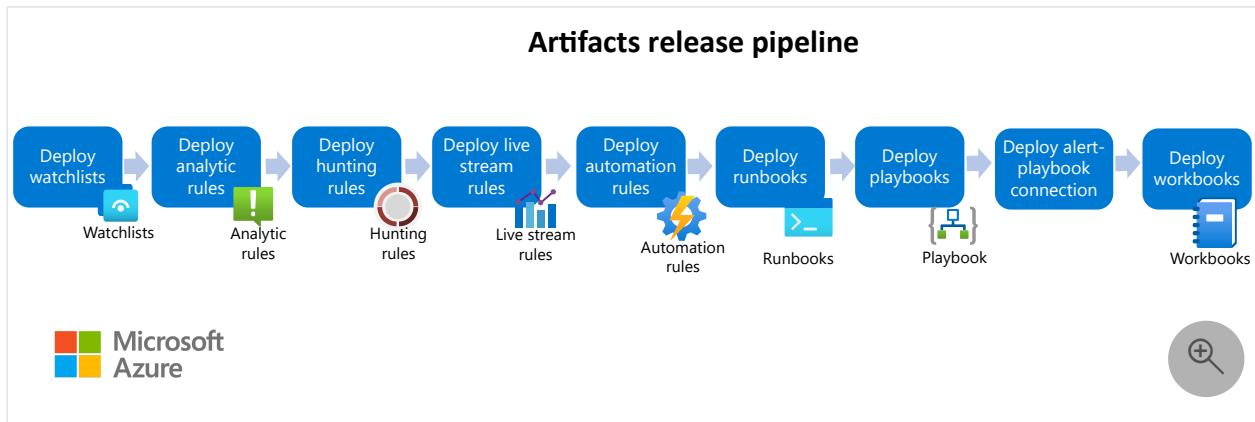
Deploy artifacts

The objectives of your deployment process are to:

- Reduce time to market.
- Increase performance across the multiple teams that are involved with setting up and managing your solution.
- Set up integration testing to evaluate the health of the environment.

Development teams use the process to ensure they can deploy, test, and validate artifact use cases that are under development. The architecture and SOC teams validate the pipeline quality on QA environments and work with the integration tests for attack scenarios. On the test cases, a team usually combines different artifacts as analytic rules, remediation playbooks, watchlists, and so on. A part of each use case includes simulating attacks where the entire chain is evaluated from ingestion, detection, and remediation.

The following diagram shows the deployment process sequence that ensures your artifacts are deployed in the right order.



[Download a Visio file](#) of this architecture.

Managing Sentinel artifacts as code offers you flexible ways to maintain your operations and automate the deployment in a CI/CD DevOps pipeline.

Microsoft solutions provide automation workflows for the following artifacts.

[Expand table](#)

Artifact	Automation workflows
Watchlists	<p>Code review Schema validation</p> <p>Deployment Create, update, delete watchlists and items</p>
Analytics rules fusion Microsoft Security ML behavioral analytics Anomaly Scheduled	<p>Code review KQL Syntax validation Schema validation Pester</p> <p>Deployment Create, Enable, Update, Delete, Export Alert templates support</p>

Artifact	Automation workflows
Automation rules	<p>Code review</p> <p>Schema validation</p> <p>Deployment</p> <p>Create, enable, update, delete, export</p>
Connectors	<p>Code review</p> <p>Schema validation</p> <p>Deployment</p> <p>Actions: enable, delete (disable), update</p>
Hunting rules	<p>Code review</p> <p>KQL Syntax validation</p> <p>Schema validation</p> <p>Pester</p> <p>Deployment</p> <p>Actions: create, enable, update, delete, export</p>
Playbooks	<p>Code review</p> <p>TTK</p> <p>Deployment</p> <p>Actions: create, enable, update, delete, export</p>
Workbooks	<p>Deployment</p> <p>Actions: create, update, delete</p>
Runbooks	<p>Code review</p> <p>PowerShell syntax validation</p> <p>Pester</p> <p>Deployment</p> <p>Actions: create, enable, update, delete, export</p>

Depending on the automation language you choose, some automation capabilities might not be supported. The following diagram shows which automation capabilities are supported by language.

Components	API	PowerShell	ARM	Terraform
Onboarding	✓ **	✓ **	✓ **	✓ **
Connectors	✓	✓	✓	✓
Analytics Rules	✓	✓	✓	✓
Hunting Queries	✓	✓	✓	✓
Workbooks	✓ * **	✓ * **	✓	✗
Playbooks	✓	✓	✓	✓
Watchlists	✓	✗	✓ *	✗
KQL Functions	✓	✓	✓	✓
Automation Rules	✓ *	✗	✓ *	✗

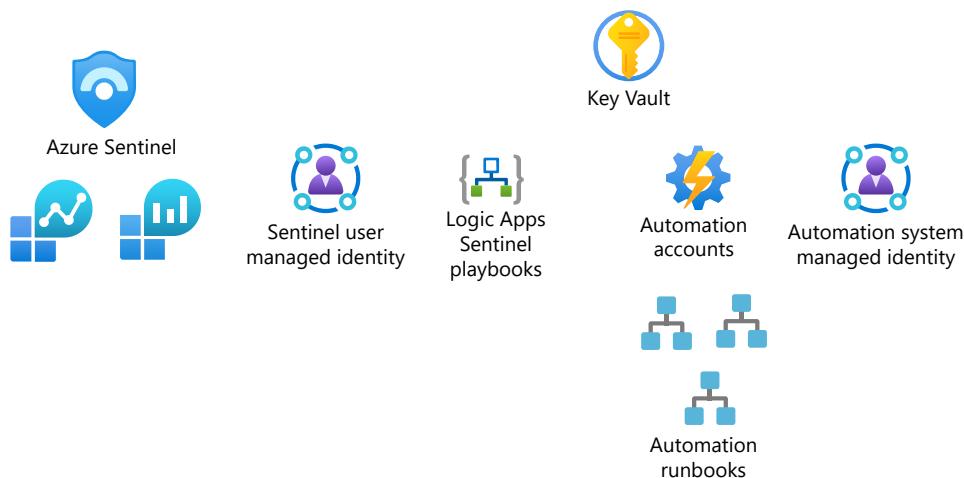
* Features in development that are not yet documented

** Automation methods that are supported by [Microsoft Operational Insights](#) or [Microsoft Insights Resource Provider APIs](#)

Azure Automation

The following diagram shows the components of simplifying Microsoft Sentinel access with managed service identity.

Architecture overview (Azure Sentinel)



[Download a Visio file](#) of this architecture.

If you need to grant access to other resources, use managed identity, which ensures a unique identity for all critical operations.

Use Azure Automation for setting up playbooks. Use PowerShell scripts for the following complex tasks and automation features:

- Integrating with third-party solutions, where different levels of credentials are required and based on Microsoft Entra ID or custom credentials:
 - Microsoft Entra user credentials
 - Microsoft Entra service principal credentials
 - Certificate authentication
 - Custom credentials
 - Managed identity
- Implementing a solution that reuses organizational scripts, or solutions that require the use of PowerShell commands to avoid complex translation to playbooks:
 - PowerShell-based solutions
 - Python-based solutions
- Implementing solutions in hybrid scenarios, where remediation actions can affect your cloud and on-premises resources.

Microsoft Sentinel repositories

Experienced DevOps teams might consider managing Microsoft Sentinel in infrastructure as code (IaC) with CI/CD pipelines that are built in Azure DevOps. Product groups understand the challenges that customers and partners face in building Azure DevOps security architecture, so the following two initiatives can help:

- Documenting the reference architecture
- Developing a new solution, announced at Ignite 2021, that's called "Microsoft Sentinel Repositories"

To support choosing the solution that fits your team's needs, the following table compares the functional and technical criteria.

[\[+\] Expand table](#)

Use case and features	Azure DevOps and GitHub custom approach	Microsoft Sentinel repositories
We want to quickly start deploying Microsoft Sentinel artifacts without spending time defining	Not recommended	Recommended

Use case and features	Azure DevOps and GitHub custom approach	Microsoft Sentinel repositories
Azure DevOps architecture components, such as agents, pipelines, Git, dashboards, a wiki, service principals, RBACs, auditing, and so on.		
We have small teams and low skill sets to manage the CI/CD pipelines.	Not recommended	Recommended
We want to control and manage all security aspects of the CI/CD pipelines.	Supported	Not supported
We need to integrate gates and branching for supervising integration, before allowing developers to trigger deployment pipelines, such as source control, coding review, rollback, workflow approval, and so on.	Supported	Partially supported
We have a customized Git or repository structure.	Supported	Supported
We don't use Resource Manager or Bicep IaC languages to build artifacts.	Supported	Not supported
We want to centrally manage the deployment of artifacts to multiple Microsoft Sentinel workspaces in a single Microsoft Entra tenant.	Supported	Supported
We want to integrate or extend CI/CD pipelines across multiple Microsoft Entra tenants.	Supported	Supported
We have playbooks with different parameterization depending on subscription, location, environment, and so on.	Supported	TBD, guidelines to be documented
We want to integrate different artifacts on the same repository to compose use cases.	Supported	Supported
We want the ability to bulk remove artifacts.	Supported	Not Supported

Availability, performance, and scalability

When choosing the architecture for the Azure DevOps agents in your company for Microsoft Sentinel scenarios, consider the following needs:

- The production environment might require a dedicated agents pool for operations over the Microsoft Sentinel environment.
- Non-production environments might share the agent pool with a large number of instances for handling the different demands from the teams, in particular, for CI/CD practices.
 - Attack simulation scenarios are a special case where dedicated agents can be required. Consider whether a dedicated pool is necessary for your testing needs.
- Organizations that work on hybrid networking scenarios might consider integrating the agents inside the network.

Organizations can create their own images for agents based on containers. For more information, see [Run a self-hosted agent in Docker](#).

Microsoft Sentinel cross-tenant management with Azure DevOps

As a global SOC or MSSP, you might have to manage multiple tenants. Azure Lighthouse supports scaled operations across several Microsoft Entra tenants at the same time, making your management tasks more efficient. For more information, see [Azure Lighthouse Overview](#).

Cross-tenant management is especially effective in the following scenarios:

- Manage Microsoft Sentinel resources [in customer tenants](#).
- [Track attacks and view security alerts across multiple tenants](#) ↗.
- [View incidents](#) across multiple Microsoft Sentinel workspaces that are spread across tenants.

Methods to onboard customers

You have two options to onboard customers, a managed service offer and an ARM template.

Manual onboarding using an ARM template

If you don't want to publish an offer to Azure Marketplace as a service provider, or you don't meet all the requirements, you can onboard customers manually by using ARM templates.

This is the most likely option in an enterprise scenario, where the same enterprise has multiple tenants.

The following table compares the onboarding methods.

 Expand table

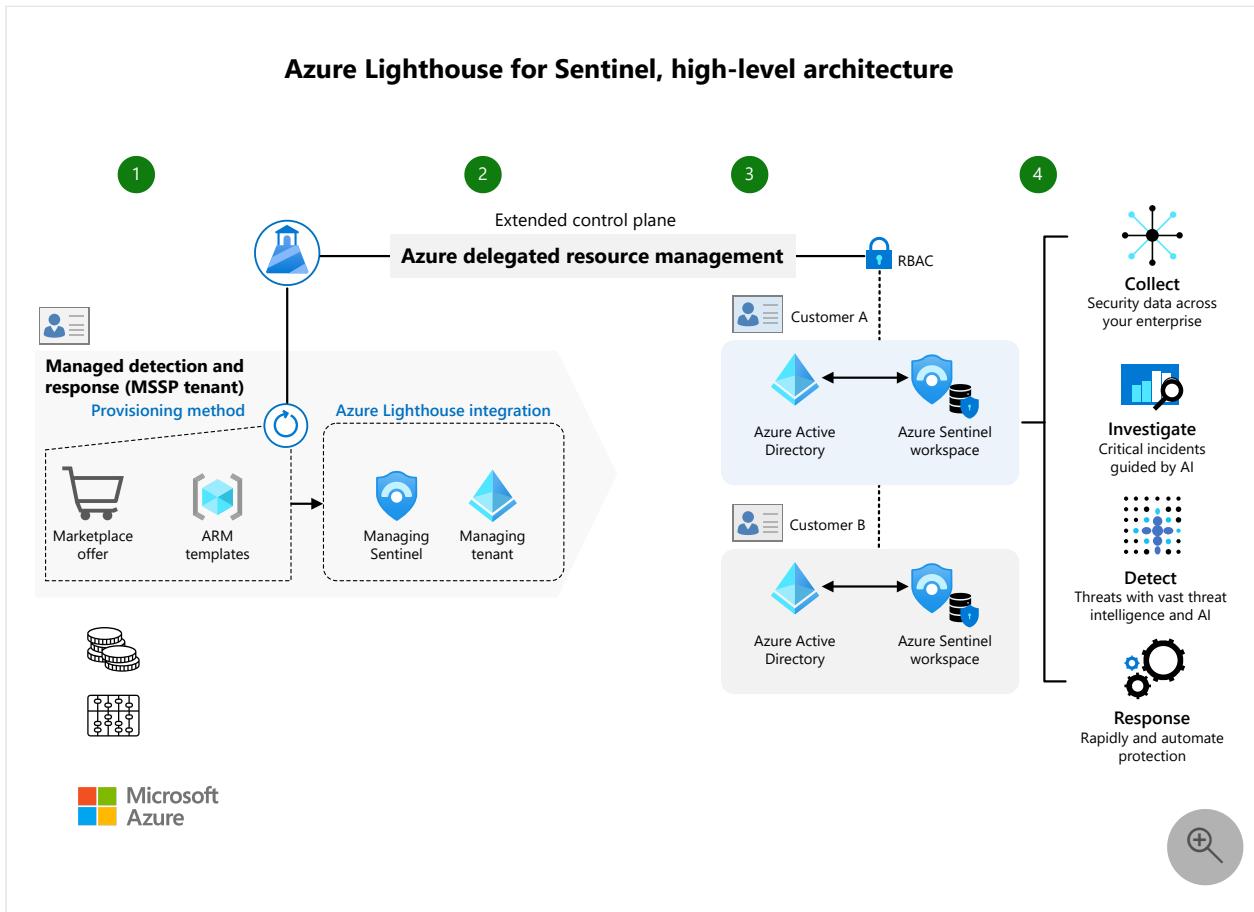
Consideration	Managed service offer	ARM templates
Requires a Partner Center account	Yes	No
Requires Silver or Gold cloud platform competency level or Azure Expert Managed Services Provider (MSP) status	Yes	No
Available to new customers through Azure Marketplace	Yes	No
Can limit offer to specific customers	Yes (only with private offers, which can't be used with subscriptions that are established through a reseller of the CSP program)	Yes
Requires customer acceptance in Azure portal	Yes	No
Can use automation to onboard multiple subscriptions, resource groups, or customers	No	Yes
Provides immediate access to new built-in roles and Azure Lighthouse features	Not always (generally available after some delay)	Yes

For more information on publishing managed service offers, see [Publish a managed service offer to Azure Marketplace](#).

For more information on how to create an ARM template, see [Create and deploy ARM templates](#).

The following diagram shows the high-level architecture integration between an MSSP tenant and a customer's resource provider tenants with Azure Lighthouse and Microsoft Sentinel.

Azure Lighthouse for Sentinel, high-level architecture



Download a [Visio file](#) of this architecture.

1. An MSP offering is integrated through an ARM template or an Azure Marketplace service offering.
2. Azure delegated resource management checks that the request is from a partner tenant and calls a managed service resource provider.
3. The managed service resource provider uses RBAC to control the MSP's access.
4. The MSP completes SecOps actions on a customer resource.
5. The billing process treats expenses as customer charges. Split billing is possible if customer entities have separate workspaces in the same Microsoft Entra tenant.
6. The security and sovereignty of the data is dependent on the customer's tenant boundary.

Identity across multiple tenants

To manage Microsoft Sentinel with Azure DevOps, evaluate the following design decisions for the components.

[Expand table](#)

Use case	Pros
Global identity for managing DevOps actions, single service principal	<p>This case applies to global deployment processes, which involve all tenants.</p> <p>Using unified identity facilitates the access for the different tenants but could make the process of managing approval actions for specific tenants complex.</p> <p>The protection mechanism and authorization model for this kind of identity is also very important, to avoid non-authorized usage that's due to the related global impact.</p>
Dedicated identity for managing DevOps actions, multiple service principals	<p>This case applies when deployment processes are dedicated for each tenant or tenant actions require approval.</p> <p>In this case, the recommendation for protecting and authorizing this identity usage is the same as in the global case, even when the impact is reduced.</p>

Code repository organization

[Expand table](#)

Use case	Pros
Unified repository with a single version of code for all tenants	<p>This case facilitates having unified versions for the code in the repository.</p> <p>In this case, a unified version of the code managing a specific version for tenants could require support over branches for each case.</p>
Unified repository with specific code folders by tenant	This case complements the single-repository case. Here, a folder structure can split dedicated artifacts by tenant.
Dedicated repository by tenant	<p>This approach provides isolation when managing code artifacts. It makes the evolution easier between tenants with different teams or requirements.</p> <p>Consolidating changes requires establishing a process between repositories, which might require effort to maintain.</p>

Build and deployment processes

[Expand table](#)

Use case	Pros
Single build process for all tenants	When all tenants work with the same version of the artifacts, this is the most straightforward option for implementing the generation and package.
Build process by Tenant	A different version of the code is deployed to each tenant.
Global deployment process for all Tenants	New deployments and updates to deployments apply to all tenants. The steps of the deployment and update processes are used for all tenants.
Global deployment process tenant by tenant	New deployments and updates to deployments apply to one or more tenants.
Dedicated deployment process by tenant	The deployment process is adapted for each tenant.

Cost optimization

Cost optimization is about reducing unnecessary expenses and improving operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

The cost of the solution depends on the following factors:

- The volume of data that your company feeds into the Microsoft Sentinel Log Analytics workspace monthly
- The commitment tier or billing method that you choose, like pay as you go (PAYG)
- The retention rate of the data policies at a table or global level

For more information, see [Azure data type retention](#).

To calculate pricing, see the [Microsoft Sentinel pricing calculator](#). For more information on the advanced pricing considerations and examples, see [Plan costs for Microsoft Sentinel](#).

You can incur additional costs if you extend your solution with the following components:

- Playbooks - runtimes for Azure Logic Apps and Azure Functions. For more information, see [Pricing details](#).

- Exporting to external storage like Azure Data Explorer, Event Hubs, or an Azure Storage account.
- A machine learning workspace and the compute that a Jupyter Notebook component uses.

Deploy this scenario

The following section describes the steps for deploying this scenario in the form of a sample use case covering the various DevOps processes.

Build and release the Microsoft Sentinel framework

First, set up the necessary NuGet components in a dedicated repository where different processes can consume the releases that you generate.

If you're working with Azure DevOps, you can create a component feed to host the different NuGet packages from the Microsoft Sentinel framework for PowerShell. For more information, see [Get started with NuGet packages](#).

Create new feed

Feeds host your packages and let you control permissions.

Name

This name will appear in the URL for your feed

Visibility

Members of your Azure Active Directory
Any member of your AAD can view the packages in this feed

Members of mitreaps
Any member of your organization can view the packages in this feed

Specific people
Only users you grant access to can view the packages in this feed

Upstream sources

Include packages from common public sources

For example: nuget.org, npmj.com

Scope

Project: SOC (Recommended)
The feed will be scoped to the SOC project. [Learn more](#).

Organization

[Cancel](#) [Create](#)

If your team chooses a GitHub registry, you can connect it as a NuGet repository, because it's compatible with the feed protocol. For more information, see [Introduction to GitHub packages](#).

When you have an available NuGet repository, the pipeline contains a service connection for NuGet. These screenshots show the configuration for the new service connection that's named Microsoft Sentinel NuGet Framework Connection.

New service connection

X

 Generic

 GitHub

 GitHub Enterprise Server

 Incoming WebHook

 Jenkins

 Jira

 Kubernetes

 Maven

 NuGet

 Other Git

 Python package download

 Python package upload

 SSH

 Service Fabric

 Subversion

 Visual Studio App Center

 npm

[Learn more](#)

Next



After configuring the feed, you can import the pipeline for building the PowerShell framework directly from GitHub in a specific fork. For more information, see [Build GitHub repositories](#). In this case, you create a new pipeline and choose GitHub as the code source.

Another option is to import the Git repository as an Azure DevOps repository that's based on Git. In both cases, to import the pipeline, specify the following path:

```
src/Build/Framework/ADO/Microsoft.Sentinel.Framework.Build.yml
```

Now you can run the pipeline for first time. Then, the framework builds and releases to the NuGet feed.

Define your Microsoft Sentinel environment

When starting with Microsoft Sentinel and using these samples, define the environment in your company, for example, Environment as Code or EaC. You specify the different elements that make up the environment in each case.

The Microsoft Sentinel architecture includes the following elements on Azure:

- Log Analytics Workspace - This workspace forms the foundation of the solution. Security-related information is stored here and the workspace is the engine for Kusto Query Language (KQL).
- Microsoft Sentinel solution over the Log Analytics workspace - This solution extends the functionality of the Log Analytics workspace to include SIEM and SOAR capabilities.
- Key Vault - The key vault keeps the secrets and keys that are used during the remediation processes.
- Automation account - This account is optional and is used for the remediation processes. The remediation process that you use is based on the PowerShell and Python runbooks. The process includes a system-managed identity that works with different resources according to best practices.
- User-managed identity - This feature acts as a Microsoft Sentinel unified identity layer that manages interactions between Microsoft Sentinel playbooks and runbooks.
- Logic App connections - These are connections for Microsoft Sentinel, the key vault, and automation that use the user-managed identity.
- External Logic App connections - These are connections for external resources that are involved in the remediations processes and which are based on the playbooks.
- Azure Event Hubs - This feature is optional and handles integration between Microsoft Sentinel and other solutions, such as Splunk, Azure Databricks and machine learning, and Resilient.
- Storage account - This feature is optional and handles integration between Microsoft Sentinel and other solutions, such as Splunk, Azure Databricks and machine learning, and Resilient.

By using examples from the repository, you can define the environment with JSON files to specify the different logical concepts. The options that are available for defining the environment can be literal or automatic.

In a literal definition, you specify the name and the elements for each resource in the environment as shown in this example.

```
JSON

{
  {
    "SubscriptionId": "<subscription-identifier-associated-with-service-connection>",
    "Name": "<environment-name>",
    "NamingConvention": "<naming-convention-template-for-automatic-cases>",
    "Location": "<environment-location>",
    "ResourceGroup": {
      "Type": "Literal",
      "ResourceGroupName": "<resource-group-name>"
    }
  },
  "Resources": {
    "Sentinel": "
```

```

{
  "Type": "Literal",
  "LogAnalyticsWorkspaceName": "<Log-Analytics-workspace-name>",
  "ManagedIdentityName": "<user-managed-identity-name>",
  "SentinelConnectionName": "<Sentinel-API-connection-name>",
  "KeyVaultName": "<Key-Vault-name>",
  "KeyVaultConnectionName": "<Key-Vault-API-connection-name>"
},
"Automation":
{
  "Type": "Literal",
  "AutomationAccountName": "<automation-account-name>",
  "AutomationAccountConnectionName": "<automation-account-API-connection-name>"
},
"Integration":
{
  "Type": "Literal",
  "EventHubNamespaces": [
    "<Event-Hubs-namespace-1-name>",
    "<Event-Hubs-namespace-2-name>",
    "<Event-Hubs-namespace-3-name>",
    "<Event-Hubs-namespace-4-name>",
    "<Event-Hubs-namespace-5-name>",
    "<Event-Hubs-namespace-6-name>",
    "<Event-Hubs-namespace-7-name>",
    "<Event-Hubs-namespace-8-name>",
    "<Event-Hubs-namespace-9-name>",
    "<Event-Hubs-namespace-10-name>",
  ],
  "StorageAccountName": "<storage-account-name>"
}
}
}

```

In an automatic definition, the element names generate automatically based on naming conventions, as shown in this example.

JSON

```

{
  {
    "SubscriptionId": "<subscription-identifier-associated-with-service-connection>",
    "Name": "<environment-name>",
    "NamingConvention": "<naming-convention-template-for-automatic-cases>",
    "Location": "<environment-location>",
    "ResourceGroup": {
      "Type": "Automatic"
    }
  },
  "Resources":
  {
    "Sentinel":
    {

```

```
        "Type": "Automatic"
    },
    "Automation":
    {
        "Type": "Automatic"
    },
    "Integration":
    {
        "Type": "Automatic",
        "MaxEventHubNamespaces": 5
    }
}
```

You can find samples in the GitHub repository under the Microsoft Sentinel environments path and use the samples as a reference in preparing your use cases.

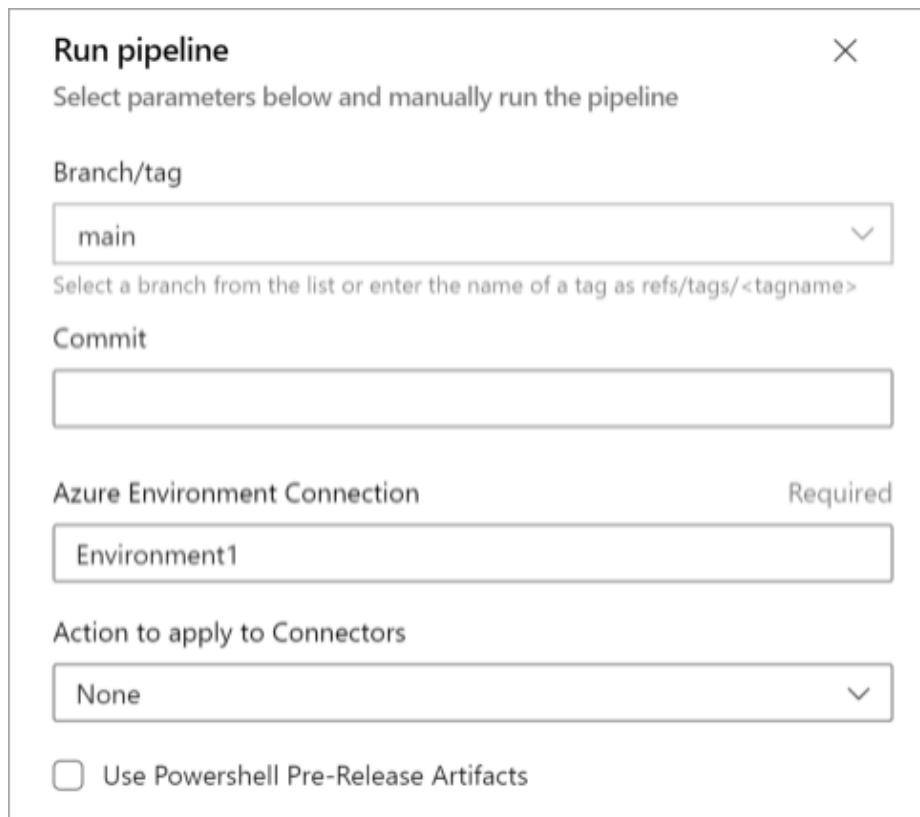
Deploy your Microsoft Sentinel environment

When you have at least one environment defined, you can create the Azure service connection to integrate with Azure DevOps. After you create the service connection, set the linked service principal to the owner role or a similar permissions level over the target subscription.

1. Import the pipeline for creating the new environment as defined in this file.

```
src/Release/Sentinel Deployment/ADO/Microsoft.Sentinel.Environment.Deployment.yml
```

2. Next, enter the name of the service connection that represents the environment.



3. Choose the branch for the environment definition in the repository.
4. Enter the name of the Azure DevOps service connection for your subscription in the **Azure Environment Connection** box.
5. Enter the name of the environment that a service connection can use to resolve multiple environments in the same subscription.
6. Choose the action to apply to the connectors.
7. Select **Use PowerShell Pre-Release Artifacts** if you want to use the prerelease versions of the PowerShell framework components.

The pipeline includes the following steps as part of the deployment flow:

- Deploy NuGet components.
- Connect by using NuGet tools with the artifacts repository.
- Resolve the feed.
- Install the required modules.
- Get the environment definition.
- Validate which resources exist in the destination.
- Add Log Analytics and Microsoft Sentinel if they're not in the workspace.
- If you already have Log Analytics, add Microsoft Sentinel over your instance of Log Analytics.
- Create a managed identity to represent the interaction between Microsoft Sentinel and Logic Apps.
- Create Azure Key Vault and set the role assignment for managing secrets and keys to the managed identity.
- If applicable, create an Azure Automation account and turn on the system-assigned managed identity.
- Set the role assignment over the key vault for the system-assigned managed identity.
- Create the Event Hubs definitions if they don't exist and set whether the definition includes the integration elements.
- Set the role assignment over the key vault for the system-assigned managed identity.
- Create the storage account definitions if they don't exist and set whether the definition includes the integration elements.
- Set the role assignment over the key vault for the system-assigned managed identity.
- Deploy the connector actions.
- Deploy the integration runbook on the Automation account.
- Deploy the Logic Apps connections if they're defined as part of the environment.

Destroy a Microsoft Sentinel environment

When the environment is no longer needed, like in the case of a development or testing environment, you can destroy it as defined in this file.

```
src/Release/Sentinel Deployment/ADO/Microsoft.Sentinel.Environment.Destroy.yml
```

As when you deploy the environment pipeline, you specify the name of the service connection that represents the environment.

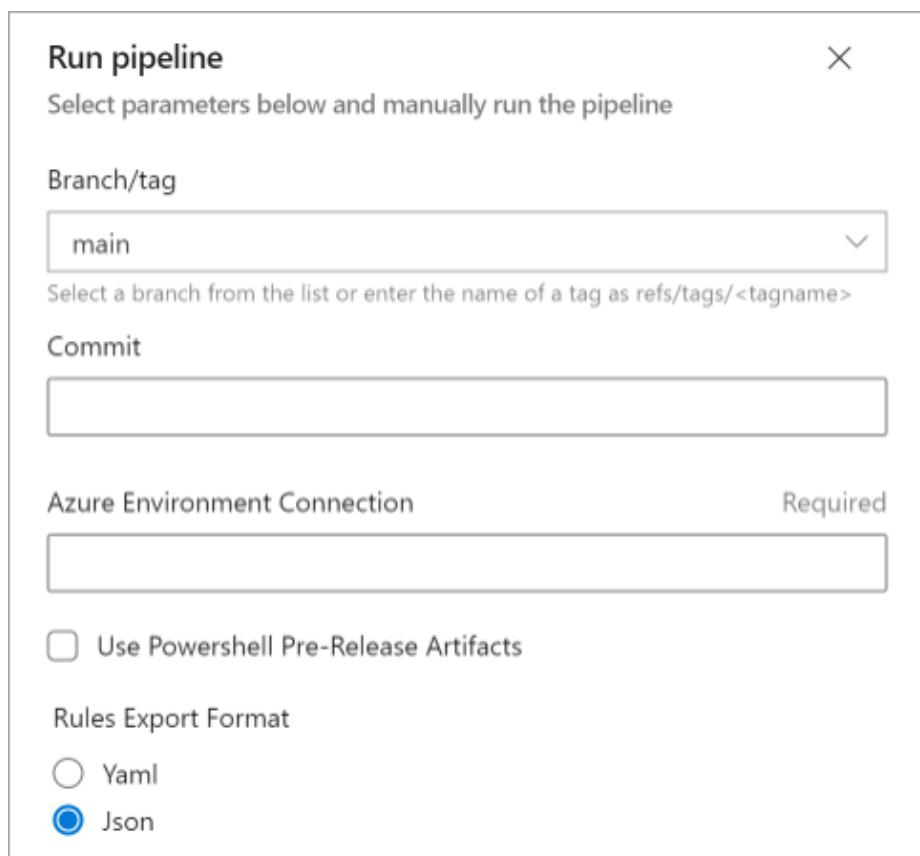
Working with your Microsoft Sentinel environment

After your environment is ready, you can start creating the artifacts for setting up the different use cases.

1. Export the artifacts from the environment that you're working on as defined in this file.

```
src/Release/Artifacts Deployment/ADO/Microsoft.Sentinel.Artifacts.Export.yml
```

2. Choose the branch for the environment definition in the repository.



3. Enter the name of the Azure DevOps service connection for the environment that's being exported in the **Azure Environment Connection** box.
4. Select **Use PowerShell Pre-Release Artifacts** if you want to use the prerelease versions of the PowerShell framework components.
5. Choose the format for the analytic and hunting rules.

The artifacts output file is available in the results. After you have the artifacts, you can include the output file in the Git repository.

Build your artifacts in the Microsoft Sentinel environment

Place your artifacts under the Microsoft Sentinel MITRE use cases path. Set up your folder structure according to the different types of artifacts.

1. Start the build process as defined in this file.

```
src/Build/Artifacts/ADO/Microsoft.Sentinel.Artifacts.Build.yaml
```

2. Choose the branch for the environment definition in the repository.

Diagram of how to choose the branch for building the artifacts.](./media/build-artifacts-pipeline.png)

3. Select **Use PowerShell Pre-Release Artifacts** if you want to use the prerelease versions of the PowerShell framework components.

The pipeline is made up of these steps:

- Deploy NuGet components.
- Connect the NuGet tools to the artifacts repository.
- Resolve the feed.
- Install the required modules.
- Get the [Test toolkit framework](#) for validating the ARM templates.
- Validate the ARM templates.
- Validate the PowerShell Runbooks code and do syntax validation.
- Run the Pester unit tests if applicable.
- Validate the KQL syntax in the hunting and analytic rules.

Deploy your artifacts to the Microsoft Sentinel environment

In deploying your artifacts, you can use the Microsoft Sentinel repositories or the deployment pipeline samples on this repository. For more information, see [Deploy custom content from your repository](#).

Microsoft Sentinel repositories

If you use Microsoft Sentinel repositories, you can set up a release process to include the artifacts in the repository that's connected to each Microsoft Sentinel instance. After the artifacts are committed in the repository, some of the steps are automatically done as part of creating the pipeline and enabling the Microsoft Sentinel repositories.

Also, you can customize the deployment processes that the Microsoft Sentinel repositories do based on practices that are described in this document. One important aspect to consider is the release approval, which you can set up by following these approaches:

- PR approval when committing the artifacts. For more information, see [Create pull requests](#).
- Release pipeline approval when running the deployment. For more information, see [Define approvals and checks](#).

Microsoft Sentinel deployment pipeline samples

By using the Microsoft Sentinel deployment pipeline samples, you can set up a release process.

1. Set up your release process as defined in this file.

```
src/Release/Artifacts Deployment/ADO/Microsoft.Sentinel.Artifacts.Deployment.yml
```

2. Choose the branch for the environment definition in the repository.



3. Enter the name of the Azure DevOps service connection for the environment that's being exported in the **Azure Environment Connection** box.
4. Select **Use PowerShell Pre-Release Artifacts** if you want to use the prerelease versions of the PowerShell framework components.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Kevin Kisoka](#) | Associate Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- To learn about Microsoft Sentinel with DevOps for single-tenant architecture, see [Deploying and managing Microsoft Sentinel as code](#).
- To learn about MSSP multi-tenant architecture, see [Combining Azure Lighthouse with Microsoft Sentinel's DevOps capabilities](#).
- For information on Managed identity with Microsoft Sentinel, see [What's new: Managed identity for Microsoft Sentinel Logic Apps connector](#).
- To learn how to deploy content from a Microsoft Sentinel repository, see [Deploy custom content from your repository](#).
- To learn about Azure DevOps Security considerations, see [Default permissions quick reference](#).
- To learn how to protect an Azure DevOps repository, see [Add protection to a repository resource](#).
- For information on how to manage Azure DevOps service connection security, see [Service connections in Azure Pipelines](#).

Related resources

- [Dev SecOps in GitHub](#)
- [Hybrid security monitoring using Microsoft Defender for cloud and Microsoft Sentinel](#)
- [Design a CI/CD pipeline using Azure DevOps](#)
- [Advanced ARM template functionality](#)

Microsoft Entra IDaaS in security operations

Microsoft Entra ID

Microsoft Sentinel

This architecture shows how security operations center (SOC) teams can incorporate Microsoft Entra identity and access capabilities into an overall integrated and layered *zero-trust* security strategy.

Network security dominated SOC operations when all services and devices were contained on managed networks in organizations. However, [Gartner](#) predicts that through 2022, the market size of cloud services will grow at a rate nearly three times that of overall IT services. As more companies embrace cloud computing, there's a shift toward treating [user identity](#) as the primary security boundary.

Securing identities in the cloud is a high priority.

- Verizon's [2020 data breach investigations report](#) stated that 37% involved use of stolen credentials, and 22% of data breaches involved phishing.
- A 2019 IBM [study of data breach incidents](#) reported that the average global cost of a data breach was \$3.9M, with the US average cost closer to \$8.2M.
- The [Microsoft 2019 security intelligence report](#) reported that phishing attacks increased by a margin of 250% between January and December of 2018.

The [zero trust security model](#) treats all hosts as if they're internet-facing, and considers the entire network to be potentially compromised and hostile. This approach focuses on building strong authentication, authorization, and encryption, while also providing compartmentalized access and better operational agility.

Gartner promotes an [adaptive security architecture](#) that replaces an incident response-based strategy with a *prevent-detect-respond-predict* model. Adaptive security combines access control, behavioral monitoring, usage management, and discovery with continuous monitoring and analysis.

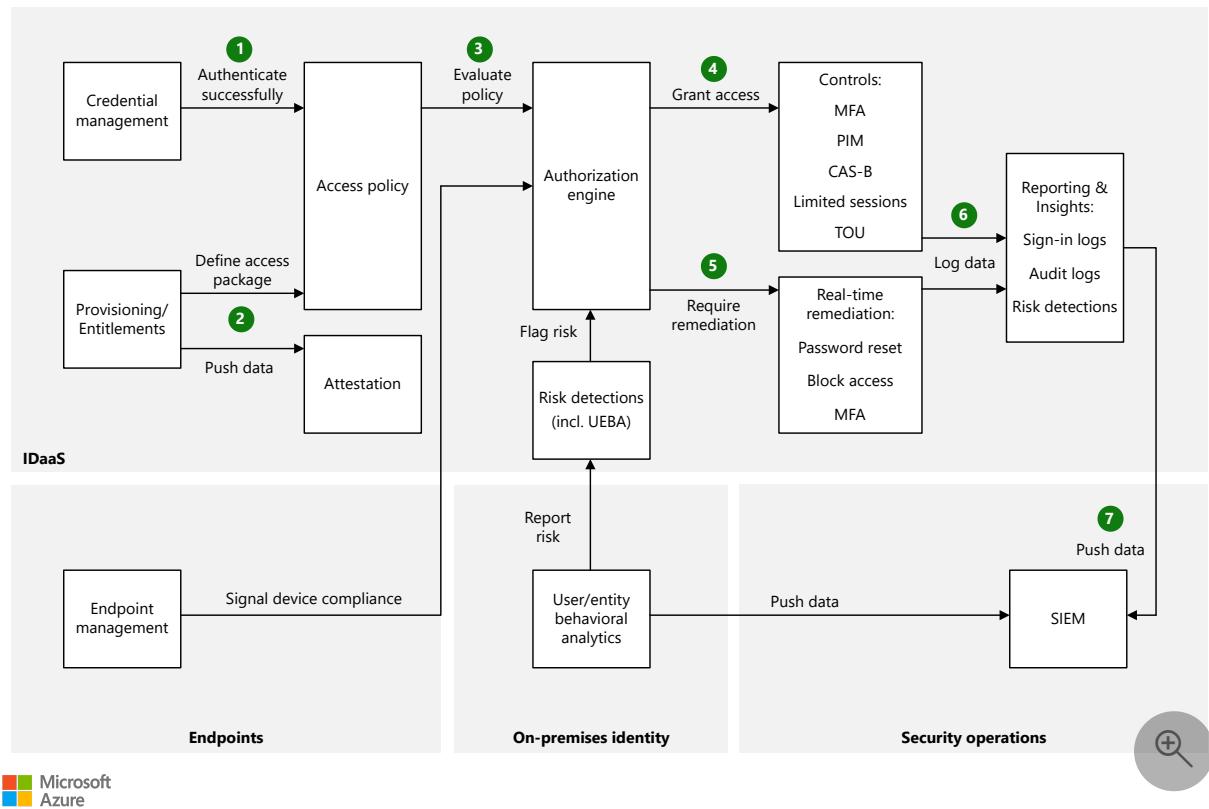
The [Microsoft Cybersecurity Reference Architecture \(MCRA\)](#) describes Microsoft's cybersecurity capabilities and how they integrate with existing security architectures, including cloud and hybrid environments, that use Microsoft Entra ID for *Identity-as-a-Service (IDaaS)*.

This article advances the zero-trust, adaptive security approach to IDaaS, emphasizing components available on the Microsoft Entra platform.

Potential use cases

- Design new security solutions
- Enhance or integrate with existing implementations
- Educate SOC teams

Architecture



[Download a Visio file](#) of this architecture.

Workflow

1. *Credential management* controls authentication.
2. *Provisioning* and *entitlement management* define the access package, assign users to resources, and push data for *attestation*.
3. The *authorization engine* evaluates the *access policy* to determine access. The engine also evaluates *risk detections*, including *user/entity behavioral analytics* (UEBA) data, and checks device compliance for *endpoint management*.
4. If authorized, the user or device gains access per *conditional access policies and controls*.

5. If authorization fails, users can do *real-time remediation* to unblock themselves.
6. All session data is *logged* for analysis and reporting.
7. The SOC team's *security information and event management system (SIEM)* receives all log, risk detection, and UEBA data from cloud and on-premises identities.

Components

The following security processes and components contribute to this Microsoft Entra IDaaS architecture.

Credential management

Credential management includes services, policies, and practices that issue, track, and update access to resources or services. Microsoft Entra credential management includes the following capabilities:

- **Self-service password reset (SSPR)** lets users self-serve and reset their own lost, forgotten, or compromised passwords. SSPR not only reduces helpdesk calls, but provides greater user flexibility and security.
- **Password writeback** syncs passwords changed in the cloud with on-premises directories in real time.
- **Banned passwords** analyzes telemetry data exposing commonly used weak or compromised passwords, and bans their use globally throughout Microsoft Entra ID. You can customize this functionality for your environment, and include a list of **custom passwords** to ban within your own organization.
- **Smart lockout** compares legitimate authentication attempts with brute-force attempts to gain unauthorized access. Under the default smart lockout policy, an account locks out for one minute after 10 failed sign-in attempts. As sign-in attempts continue to fail, the account lockout time increases. You can use policies to adjust the settings for the appropriate mix of security and usability for your organization.
- **Multi-factor authentication (MFA)** requires multiple forms of authentication when users attempt to access protected resources. Most users are familiar with using something they know, like a password, when accessing resources. MFA asks users to also demonstrate something that they have, like access to a trusted device, or something that they are, like a biometric identifier. MFA can use different kinds of **authentication methods** like phone calls, text messages, or **notification through the authenticator app ↗**.

- [Passwordless authentication](#) replaces the password in the authentication workflow with a smartphone or hardware token, biometric identifier, or PIN. Microsoft passwordless authentication can work with Azure resources like [Windows Hello for Business](#), and the [Microsoft Authenticator app](#) on mobile devices. You can also enable passwordless authentication with [FIDO2-compatible security keys](#), which use [WebAuthn](#) and the FIDO Alliance's Client-to-Authenticator (CTAP) protocol.

App provisioning and entitlement

- [Entitlement management](#) is a Microsoft Entra [identity governance](#) feature that enables organizations to manage identity and access lifecycle at scale. Entitlement management automates access request workflows, access assignments, reviews, and expirations.
- [Microsoft Entra provisioning](#) lets you automatically create user identities and roles in applications that users need to access. You can configure [Microsoft Entra provisioning](#) for third-party *software-as-a-service (SaaS)* apps like [SuccessFactors](#), [Workday](#), and [many more](#).
- [Seamless single sign-on \(SSO\)](#) automatically authenticates users to cloud-based applications once they sign into their corporate devices. You can use Microsoft Entra seamless SSO with either [password hash synchronization](#) or [pass-through authentication](#).
- Attestation with [Microsoft Entra access reviews](#) help meet monitoring and auditing requirements. Access reviews let you do things like quickly identify the number of admin users, make sure new employees can access needed resources, or review users' activity to determine whether they still need access.

Conditional access policies and controls

A [conditional access policy](#) is an if-then statement of assignments and access controls. You define the response ("do this") to the reason for triggering your policy ("if this"), enabling the *authorization engine* to make decisions that enforce organizational policies. With [Microsoft Entra Conditional Access](#), you can control how authorized users access your apps. The Microsoft Entra ID [What If tool](#) can help you understand why a conditional access policy was or wasn't applied, or if a policy would apply to a user in a specific circumstance.

[Conditional access controls](#) work in conjunction with conditional access policies to help enforce organizational policy. Microsoft Entra Conditional Access controls let you

implement security based on factors detected at the time of the access request, rather than a one-size fits all approach. By coupling conditional access controls with access conditions, you reduce the need to create additional security controls. As a typical example, you can allow users on a domain-joined device to access resources using SSO, but require MFA for users off-network or using their own devices.

Microsoft Entra ID can use the following conditional access controls with conditional access policies:

- [Azure role-based access control \(Azure RBAC\)](#) lets you configure and assign appropriate roles to users who need to do administrative or specialized tasks with Azure resources. You can use Azure RBAC to create or maintain separate dedicated admin-only accounts, scope access to roles you set up, time limit access, or grant access through approval workflows.
- [Privileged identity management \(PIM\)](#) helps reduce the attack vector for your organization by letting you add additional monitoring and protection to administrative accounts. With [Microsoft Entra PIM](#), you can manage and control access to resources within Azure, Microsoft Entra ID, and other Microsoft 365 services with [just-in-time \(JIT\) access and just-enough-administration \(JEA\)](#). PIM provides a history of administrative activities and a change log, and alerts you when users are added or removed from roles you define.

You can use PIM to [require approval](#) or justification for activating administrative roles. Users can maintain normal privileges most of the time, and request and receive access to roles they need to complete administrative or specialized tasks. When they complete their work and sign out, or the time limit on their access expires, they can reauthenticate with their standard user permissions.

- [Microsoft Defender for Cloud Apps](#) is a *cloud access security broker (CASB)* that analyzes traffic logs to discover and monitor the applications and services in use in your organization. With Defender for Cloud Apps, you can:
 - [Create policies](#) to manage interaction with apps and services
 - Identify applications as [sanctioned or unsanctioned](#)
 - [Control and limit access to data](#)
 - [Apply information protection](#) to guard against information loss

Defender for Cloud Apps can also work with [access policies](#) and [session policies](#) to control user access to SaaS apps. For example, you can:

- [Limit the IP ranges](#) that can access apps
- [Require MFA](#) for app access
- [Allow activities only from within approved apps](#)

- The [access control page in the SharePoint admin center](#) provides several ways to control access to SharePoint and OneDrive content. You can choose to [block access](#), allow [limited, web-only access](#) from unmanaged devices, or [control access based on network location](#).
- You can [scope application permissions to specific Exchange Online mailboxes](#) by using [ApplicationAccessPolicy](#) from the Microsoft Graph API.
- [Terms of Use \(TOU\)](#) provides a way to present information that end users must consent to before gaining access to protected resources. You upload TOU documents to Azure as PDF files, which are then available as controls in conditional access policies. By creating a conditional access policy that requires users to consent to TOU at sign-in, you can easily audit users that accepted the TOU.
- [Endpoint management](#) controls how authorized users can access your cloud apps from a broad range of devices, including mobile and personal devices. You can use conditional access policies to restrict access only to devices that meet certain security and compliance standards. These *managed devices* require a [device identity](#).

Risk detection

Azure Identity Protection includes several policies that can help your organization manage responses to suspicious user actions. *User risk* is the probability that a user identity is compromised. *Sign-in risk* is the probability that a sign-in request isn't coming from the user. Microsoft Entra ID calculates sign-in risk scores based on the probability of the sign-in request originating from the actual user, based on behavioral analytics.

- [Microsoft Entra risk detections](#) use adaptive machine learning algorithms and heuristics to detect suspicious actions related to user accounts. Each detected suspicious action is stored in a record called a *risk detection*. Microsoft Entra ID calculates user and sign-in risk probability using this data, enhanced with Microsoft's internal and external threat intelligence sources and signals.
- You can use the Identity Protection [risk detection APIs](#) in Microsoft Graph to expose information about risky users and sign-ins.
- [Real-time remediation](#) allows users to unblock themselves by using SSPR and MFA to self-remediate some risk detections.

Considerations

Keep these points in mind when you use this solution.

Logging

Microsoft Entra [audit reports](#) provide traceability for Azure activities with audit logs, sign-in logs, and risky sign-in and risky user reports. You can filter and search the log data based on several parameters, including service, category, activity, and status.

You can route Microsoft Entra ID log data to endpoints like:

- Azure Storage accounts
- [Azure Monitor logs](#)
- [Azure event hubs](#)
- SIEM solutions like [Microsoft Sentinel](#), [ArcSight](#), [Splunk](#), [SumoLogic](#), [other external SIEM tools](#), or your own solution.

You can also use the Microsoft Graph [reporting API](#) to retrieve and consume Microsoft Entra ID log data within your own scripts.

On-premises and hybrid considerations

Authentication methods are key to securing your organization's identities in a hybrid scenario. Microsoft provides [specific guidance](#) on choosing a hybrid authentication method with Microsoft Entra ID.

[Microsoft Defender for Identity](#) can use your on-premises Active Directory signals to identify, detect, and investigate advanced threats, compromised identities, and malicious insider actions. Defender for Identity uses UEBA to identify insider threats and flag risk. Even if an identity becomes compromised, Defender for Identity can help identify the compromise based on unusual user behavior.

Defender for Identity is [integrated with Defender for Cloud Apps](#) to extend protection to cloud apps. You can use Defender for Cloud Apps to create [session policies](#) that protect your files on download. For example, you can automatically set view-only permissions on any file downloaded by specific types of users.

You can configure an on-premises application in Microsoft Entra ID to use Defender for Cloud Apps for real-time monitoring. Defender for Cloud Apps uses Conditional Access App Control to monitor and control sessions in real-time based on Conditional Access policies. You can apply these policies to on-premises applications that use Application Proxy in Microsoft Entra ID.

Microsoft Entra [Application Proxy](#) lets users access on-premises web applications from remote clients. With Application Proxy, you can monitor all sign-in activities for your applications in one place.

You can use Defender for Identity with [Microsoft Entra ID Protection](#) to help protect user identities that are synchronized to Azure with [Microsoft Entra Connect](#).

If some of your apps already use an existing [delivery controller or network controller](#) to provide off-network access, you can integrate them with Microsoft Entra ID. Several partners including [Akamai](#), [Citrix](#), [F5 Networks](#), and [Zscaler](#) offer solutions and guidance for integration with Microsoft Entra ID.

Cost optimization

Microsoft Entra pricing ranges from free, for features like SSO and MFA, to Premium P2, for features like PIM and Entitlement Management. For pricing details, see [Microsoft Entra pricing](#).

Next steps

- [Zero Trust security](#)
- [Zero Trust Deployment Guide for Microsoft Entra ID](#)
- [Overview of the security pillar](#)
- [Microsoft Entra demo tenant](#) (requires a Microsoft Partner Network account), or [Enterprise Mobility + Security free trial](#)
- [Microsoft Entra deployment plans](#)

Related resources

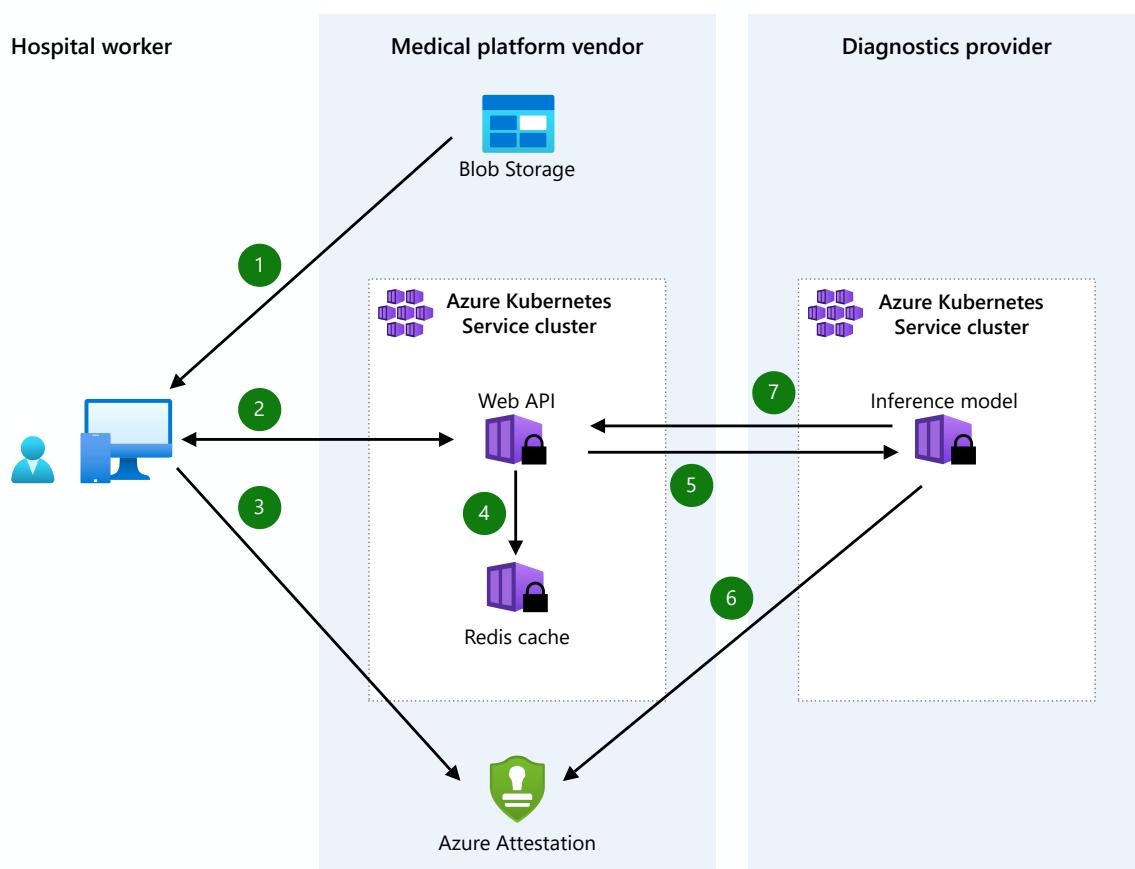
- [Azure IoT reference architecture](#)
- [COVID-19 safe environments with IoT Edge monitoring and alerting](#)
- [Security considerations for highly sensitive IaaS apps in Azure](#)

Confidential computing on a healthcare platform

Azure Kubernetes Service (AKS)

This article presents a solution that Azure confidential computing (ACC) offers for encrypting in-use data.

Architecture



 Microsoft Azure



Download a [Visio file](#) of this architecture.

The diagram outlines the architecture. Throughout the system:

- Network communication is TLS encrypted in transit.
- [Azure Monitor](#) tracks component performance, and [Azure Container Registry](#) (ACR) manages the solution's containers.

Workflow

The solution involves the following steps:

1. A clerk for a local hospital opens a web portal. The entire web app is an [Azure Blob Storage](#) static website.
2. The clerk enters data into the hospital's web portal, which connects to a Python Flask-based web API built by a popular medical platform vendor. A confidential node in the [SCONE](#) confidential computing software protects the patient data. SCONE works within an AKS cluster that has the Software Guard Extensions (SGX) enabled that help run the container in an enclave. The Web API will provide evidence that the sensitive data and app code is encrypted and isolated in a Trusted Execution Environment. This means that no humans, no processes, and no logs have access to the cleartext data or the application code.
3. The hospital's web app client requests that an attestation service (Azure Attestation) validates this evidence, and receives a signed *attestation token* for other apps to verify.
4. If the Web API requires additional components (like a Redis cache), it can pass along the attestation token to verify that the data and app code have so far remained in a safe enclave (see step 6 for verification).
5. The Web API can even consume remote services, such as an ML model hosted by a third-party diagnostics provider. When doing so, it continues to pass along any attestation tokens for evidence that required enclaves are safe. The Web API could also attempt to receive and verify attestation tokens for the diagnostic provider's infrastructure.
6. The remote infrastructure accepts the attestation token from the medical platform's web api and verifies it with a public certificate found in the Azure Attestation service. If the token is verified, there is near certainty that the enclave is safe and neither the data or app code have been opened outside of the enclave.
7. The diagnostics provider, confident that the data has not been exposed, sends it into its own enclave in an Open Neural Network Exchange (ONNX) runtime server. An AI model interprets the medical imagery and returns its diagnosis results back to the medical platform's confidential Web API app. From here, the software can then interact with patient records and/or contact other hospital staff.

Components

- [Azure Blob Storage](#) serves static content like HTML, CSS, JavaScript, and image files directly from a storage container.

- [Azure Attestation](#) is a unified solution that remotely verifies the trustworthiness of a platform. Azure Attestation also remotely verifies the integrity of the binaries that run in the platform. Use Azure Attestation to establish trust with the confidential application.
- [Azure Kubernetes Service](#) simplifies the process of deploying a Kubernetes cluster.
- [Confidential computing nodes](#) are hosted on a specific virtual machine series that can run sensitive workloads on AKS within a hardware-based trusted execution environment (TEE) by allowing user-level code to allocate private regions of memory, known as enclaves. Confidential computing nodes can support confidential containers or enclave-aware containers.
- [SCONE platform](#) is an Azure Partner independent software vendor (ISV) solution from Scontain.
- [Redis](#) is an open-source, in-memory data structure store.
- [Secure Container Environment \(SCONE\)](#) supports the execution of confidential applications in containers that run inside a Kubernetes cluster.
- [Confidential Inferencing ONNX Runtime Server Enclave \(ONNX RT - Enclave\)](#) is a host that restricts the ML hosting party from accessing both the inferencing request and its corresponding response.

Alternatives

- You can use [Fortanix](#) instead of SCONE to deploy confidential containers to use with your containerized application. Fortanix provides the flexibility you need to run and manage the broadest set of applications: existing applications, new enclave-native applications, and pre-packaged applications.
- [Graphene](#) is a lightweight, open-source guest OS. Graphene can run a single Linux application in an isolated environment with benefits comparable to running a complete OS. It has good tooling support for converting existing Docker container applications to Graphene Shielded Containers (GSC).

Scenario details

When organizations collaborate, they share information. But most parties don't want to give other parties access to all parts of the data. Mechanisms exist for safeguarding data at rest and in transit. However, encrypting data in use poses different challenges.

By using confidential computing and containers, the solution provides a way for a provider-hosted application to securely collaborate with a hospital and a third-party diagnostic provider. Azure Kubernetes Service (AKS) hosts confidential computing nodes. Azure Attestation establishes trust with the diagnostic provider. By using these Azure components, the architecture isolates the sensitive data of the hospital patients while the specific shared data is being processed in the cloud. The hospital data is then inaccessible to the diagnostic provider. Through this architecture, the provider-hosted application can also take advantage of advanced analytics. The diagnostic provider makes these analytics available as confidential computing services of machine learning (ML) applications.

Potential use cases

Many industries protect their data by using confidential computing for these purposes:

- Securing financial data
- Protecting patient information
- Running ML processes on sensitive information
- Performing algorithms on encrypted datasets from many sources
- Protecting container data and code integrity

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Azure confidential computing virtual machines (VMs) are available in 2nd-generation D family sizes for general purpose needs. These sizes are known collectively as D-Series v2 or DCsv2 series. This scenario uses Intel SGX-enabled DCs_v2-series virtual machines with Gen2 operating system (OS) images. But you can only deploy certain sizes in certain regions. For more information, see [Quickstart: Deploy an Azure Confidential Computing VM in the Marketplace](#) and [Products available by region](#) ↗.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To explore the cost of running this scenario, use the [Azure pricing calculator](#), which preconfigures all Azure services.

A [sample cost profile](#) is available for the Contoso Medical SaaS Platform, as pictured in the diagram. It includes the following components:

- System node pool and SGX node pool: no disks, all ephemeral
- AKS Load Balancer
- Azure Virtual Network: nominal
- Azure Container Registry
- Storage account for single-page application (SPA)

The profile doesn't include the following components:

- Azure Attestation Service: free
- Azure Monitor Logs: usage based
- SCONE ISV licensing
- Compliance services required for solutions working with sensitive data, including:
 - Microsoft Defender for Cloud and Microsoft Defender for Kubernetes
 - Azure DDoS Protection: Network Protection
 - Azure Firewall
 - Azure Application Gateway and Azure Web Application Firewall
 - Azure Key Vault

Deploy this scenario

Deploying this scenario involves the following high-level steps:

- Deploy the confidential inferencing server on an existing SGX-enabled AKS Cluster. See the [confidential ONNX inference server](#) project on GitHub for information on this step.
- Configure Azure Attestation policies.
- Deploy an SGX-enabled AKS cluster node pool.
- Get access to [curated confidential applications called SconeApps](#). SconeApps are available on a private GitHub repository that's currently only available for commercial customers, through SCONE Standard Edition. Go to the [SCONE website](#) and contact the company directly to get this service level.
- Install and run SCONE services on your AKS cluster.

- Install and test the Flask-based application on your AKS cluster.
- Deploy and access the web client.

These steps focus on the enclave containers. A secured infrastructure would extend beyond this implementation and include compliance requirements, such as added protections required by HIPAA.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Amar Gowda](#) | Principal Product Manager

Next steps

- Learn more about [Azure confidential computing](#)
- [Static website hosting in Blob Storage](#)
- See the [confidential ONNX inference server](#) project on GitHub.
- [Official ONNX runtime website](#)
- [Confidential ONNX inference server \(GitHub sample\)](#)
- [Confidential containers on AKS](#)
- [MobileCoin use case with anonymized blockchain data](#)
- [Sample brain segmentation image](#) for use with the delineation function that invokes the confidential inferencing server.

Related resources

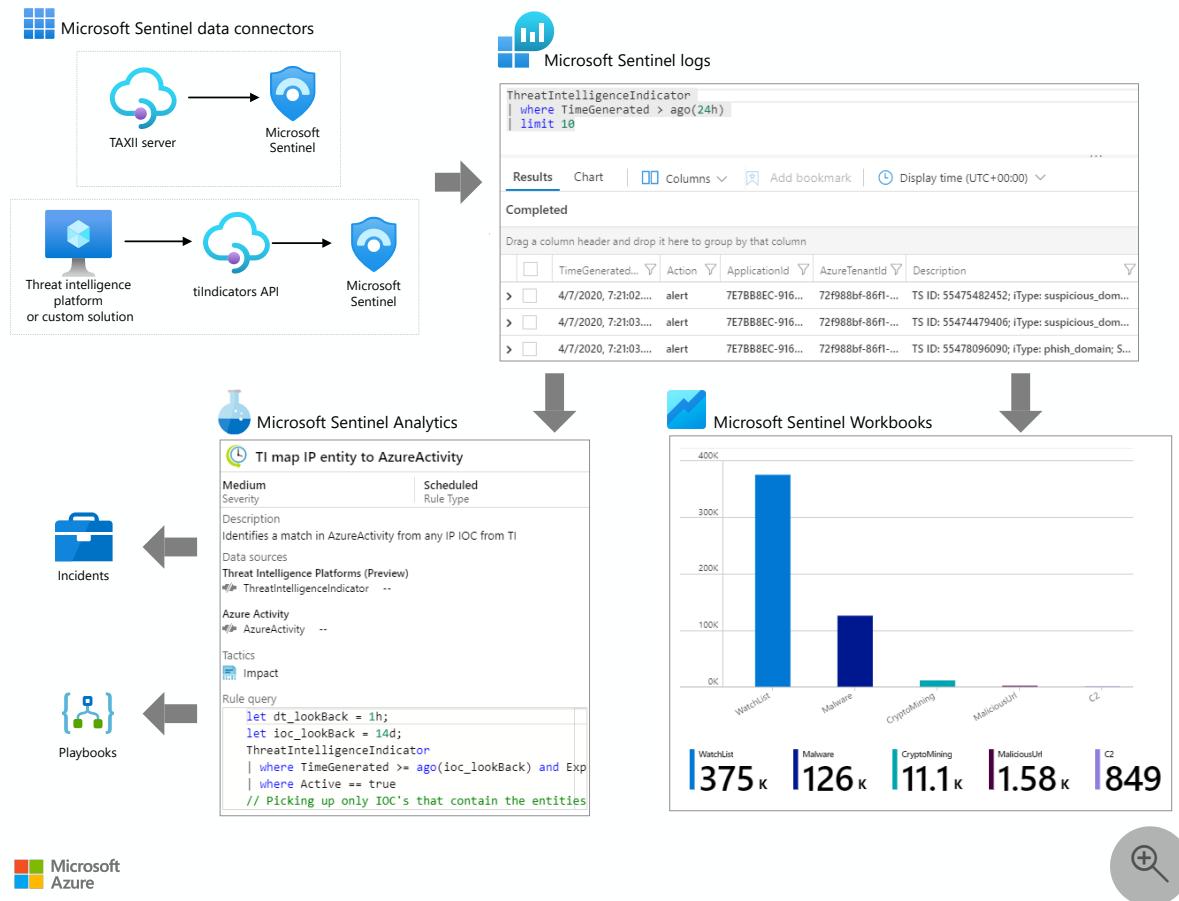
- [Health data consortium on Azure](#)
- [HIPAA and HITRUST compliant health data AI](#)
- [Baseline architecture for an Azure Kubernetes Service \(AKS\) cluster](#)

Threat indicators for cyber threat intelligence in Microsoft Sentinel

Microsoft Entra ID Azure Logic Apps Azure Monitor Microsoft Sentinel

This article describes how a cloud-based *security information and event management (SIEM)* solution like **Microsoft Sentinel** can use *threat indicators* to detect, provide context, and inform responses to existing or potential cyber threats.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

You can use Microsoft Sentinel to:

- Import threat indicators from [Structured Threat Information Expression \(STIX\)](#) and [Trusted Automated Exchange of Intelligence Information \(TAXII\)](#) servers, or any *threat intelligence platform (TIP)* solution.
- View and query threat indicator data.
- Create analytics rules to generate security alerts, incidents, and automated responses from *cyber threat intelligence (CTI)* data.
- Visualize key CTI information in workbooks.

Threat indicator data connectors

Microsoft Sentinel imports threat indicators, like all other event data, by using data connectors. The two Microsoft Sentinel data connectors for threat indicators are:

- Threat Intelligence – TAXII
- Threat Intelligence Platforms

Depending on where your organization gets its threat indicator data, you can use either or both data connectors. Enable the data connectors in each workspace where you want to receive the data.

Threat Intelligence – TAXII data connector

The most widely adopted industry standard for CTI transmission is the [STIX data format](#) and [TAXII protocol](#). Organizations that get threat indicators from current STIX/TAXII version 2.x solutions can import their threat indicators into Microsoft Sentinel by using the Threat Intelligence – TAXII data connector. The built-in Microsoft Sentinel TAXII client imports threat intelligence from TAXII 2.x servers.

For more information on how to import STIX/TAXII threat indicator data into Microsoft Sentinel, see [Import threat indicators with the TAXII data connector](#).

Threat Intelligence Platforms data connector

Many organizations use TIP solutions like MISP, Anomali ThreatStream, ThreatConnect, or Palo Alto Networks MineMeld to aggregate threat indicator feeds from various sources. Organizations use the TIP to curate the data. Then they choose which threat indicators to apply to security solutions like network devices, advanced threat protection solutions, or SIEMs like Microsoft Sentinel. The Threat Intelligence Platforms data connector lets organizations use their integrated TIP solution with Microsoft Sentinel.

The Threat Intelligence Platforms data connector uses the [Microsoft Graph Security](#) [tilndicators API](#). Any organization that has a custom TIP can use this data connector to

use the `tiIndicators` API and send indicators to Microsoft Sentinel and other Microsoft security solutions like [Defender ATP](#).

For more information on how to import TIP data into Microsoft Sentinel, see [Import threat indicators with the Platforms data connector](#).

Threat indicator logs

After you import threat indicators into Microsoft Sentinel by using the Threat Intelligence – TAXII or Threat Intelligence Platforms data connectors, you can view the imported data in the **ThreatIntelligenceIndicator** table in **Logs**, where all Microsoft Sentinel event data is stored. Microsoft Sentinel features like Analytics and Workbooks also use this table.

For more information on how to work with the threat indicators log, see [Work with threat indicators in Microsoft Sentinel](#).

Microsoft Sentinel Analytics

The most crucial use for threat indicators in SIEM solutions is to power analytics that match events with threat indicators to produce security alerts, incidents, and automated responses. Microsoft Sentinel Analytics creates analytics rules that trigger on schedule to generate alerts. You express rule parameters as queries. Then you configure how often the rule runs, what query results generate security alerts and incidents, and any automated responses to the alerts.

You can create new analytics rules from scratch or a set of built-in Microsoft Sentinel rule templates that you can use or modify to meet your needs. The analytics rule templates that match threat indicators with event data are all titled starting with **TI map**. They all work similarly.

The template differences are which type of threat indicators to use, such as domain, email, file hash, IP address, or URL, and which event types to match against. Each template lists the required data sources for the rule to function, so you can see if you have the necessary events already imported in Microsoft Sentinel.

For more information on how to create an analytics rule from a template, see [Create an analytics rule from a template](#).

In Microsoft Sentinel, enabled analytics rules are on the **Active rules** tab of the **Analytics** section. You can edit, enable, disable, duplicate, or delete active rules.

Generated security alerts are in the **SecurityAlert** table in the **Logs** section of Microsoft Sentinel. The security alerts also generate security incidents in the **Incidents** section. Security operations teams can triage and investigate the incidents to determine appropriate responses. For more information, see [Tutorial: Investigate incidents with Microsoft Sentinel](#).

You can also designate automation to trigger when the rules generate security alerts. Automation in Microsoft Sentinel uses playbooks powered by Azure Logic Apps. For more information, see [Tutorial: Set up automated threat responses in Microsoft Sentinel](#).

Microsoft Sentinel Threat Intelligence Workbook

Workbooks provide powerful interactive dashboards that give you insights into all aspects of Microsoft Sentinel. You can use a Microsoft Sentinel workbook to visualize key CTI information. The templates provide a starting point, and you can easily customize them for your business needs. You can create new dashboards that combine many different data sources and visualize your data in unique ways. Microsoft Sentinel workbooks are based on [Azure Monitor workbooks](#), so extensive documentation and templates are available.

For more information on how to view and edit the Microsoft Sentinel Threat Intelligence Workbook, see [View and edit the Threat Intelligence Workbook](#).

Alternatives

- Threat indicators provide useful context in other Microsoft Sentinel experiences like hunting and notebooks. For more information about using CTI in notebooks, see [Jupyter notebooks in Sentinel](#).
- Any organization that has a custom TIP can use the [Microsoft Graph Security](#) `tilndicators` API to send threat indicators to other Microsoft security solutions like [Defender ATP](#).
- Microsoft Sentinel provides many other built-in data connectors to solutions like Microsoft Threat Protection, Microsoft 365 sources, and Microsoft Defender for Cloud Apps. There are also built-in connectors to the broader security ecosystem for non-Microsoft solutions. You can also use common event format, Syslog, or REST API to connect your data sources with Microsoft Sentinel. For more information, see [Connect data sources](#).

Scenario details

Cyber threat intelligence can come from many sources, such as open-source data feeds, threat intelligence sharing communities, paid intelligence feeds, and security investigations within organizations.

CTI can range from written reports on a threat actor's motivations, infrastructure, and techniques to specific observations of IP addresses, domains, and file hashes. CTI provides essential context for unusual activity, so security personnel can act quickly to protect people and assets.

The most utilized CTI in SIEM solutions like Microsoft Sentinel is threat indicator data, sometimes called *Indicators of Compromise (IoCs)*. Threat indicators associate URLs, file hashes, IP addresses, and other data with known threat activity like phishing, botnets, or malware.

This form of threat intelligence is often called *tactical threat intelligence* because security products and automation can use it on a large scale to protect and detect potential threats. Microsoft Sentinel can help detect, respond to, and provide CTI context for malicious cyber activity.

Potential use cases

- Connect to open-source threat indicator data from public servers to identify, analyze, and respond to threat activity.
- Use existing threat intelligence platforms or custom solutions with the Microsoft Graph `tiIndicators` API to connect and control access to threat indicator data.
- Provide CTI context and reporting for security investigators and stakeholders.

Considerations

- The Microsoft Sentinel Threat Intelligence data connectors are currently in public preview. Certain features might not be supported or might have constrained capabilities.
- Microsoft Sentinel uses *Azure role-based access control (Azure RBAC)* to assign the built-in roles Contributor, Reader, and Responder to users, groups, and Azure services. These roles can interact with Azure roles (Owner, Contributor, Reader) and Log Analytics roles (Log Analytics reader, Log Analytics contributor). You can create custom roles and use advanced Azure RBAC on the data you store in Microsoft Sentinel. For more information, see [Permissions in Microsoft Sentinel](#).
- Microsoft Sentinel is free for the first 31 days on any Azure Monitor Log Analytics workspace. Afterward, you can use pay-as-you-go or Capacity Reservations models

for the data you ingest and store. For more information, see [Microsoft Sentinel pricing](#).

Deploy this scenario

The following sections provide steps on how to:

- Enable the [Threat Intelligence – TAXII](#) and [Threat Intelligence Platforms](#) data connectors.
- Create an example Microsoft Sentinel [Analytics rule](#) to generate security alerts and incidents from CTI data.
- View and edit the Microsoft Sentinel [Threat Intelligence Workbook](#).

Import threat indicators with the TAXII data connector

Warning

The following instructions use Limo, Anomali's free STIX/TAXII feed. This feed has reached **end-of-life and is no longer being updated**. The following instructions can't be completed as written. You can substitute this feed with another API-compatible feed you can access.

TAXII 2.x servers advertise API roots, which are URLs that host threat intelligence collections. If you already know the TAXII server **API root** and **Collection ID** you want to work with, you can skip ahead and enable the TAXII connector in Microsoft Sentinel.

If you don't have the API root, you can usually get it from the threat intelligence provider's documentation page, but sometimes the only information available is the discovery endpoint URL. You can find the API root by using the discovery endpoint. The following example uses the discovery endpoint of the [Anomali Limo](#) ThreatStream TAXII 2.0 server.

1. From a browser, go to the ThreatStream TAXII 2.0 server discovery endpoint, <https://limo.anomali.com/taxii>. Sign in by using the username *guest* and password *guest*. After you sign in, you see the following information:

JSON

```
{  
  "api_roots":  
  [  
    "https://limo.anomali.com/api/v1/taxii2/feeds/",  
    "https://limo.anomali.com/api/v1/taxii2/feeds/  
  ]  
}
```

```

        "https://limo.anomali.com/api/v1/taxii2/trusted_circles/",
        "https://limo.anomali.com/api/v1/taxii2/search_filters/"
    ],
    "contact": "info@anomali.com",
    "default": "https://limo.anomali.com/api/v1/taxii2/feeds/",
    "description": "TAXII 2.0 Server (guest)",
    "title": "ThreatStream Taxii 2.0 Server"
}

```

2. To browse collections, enter the API root you got from the previous step into your browser: <https://limo.anomali.com/api/v1/taxii2/feeds/collections/>. You see information like:

JSON

```
{
  "collections": [
    {
      "can_read": true,
      "can_write": false,
      "description": "",
      "id": "107",
      "title": "Phish Tank"
    },
    ...
    {
      "can_read": true,
      "can_write": false,
      "description": "",
      "id": "41",
      "title": "CyberCrime"
    }
  ]
}
```

You now have the information you need to connect Microsoft Sentinel to one or more TAXII server collections provided by Anomali Limo. For example:

[] [Expand table](#)

API root	Collection ID
Phish Tank	107
CyberCrime	41

To enable the Threat Intelligence – TAXII data connector in Microsoft Sentinel:

1. In the [Azure portal](#), search for and select **Microsoft Sentinel**.
2. Select the workspace where you want to import threat indicators from the TAXII service.
3. Select **Data connectors** from the leftmost pane. Search for and select **Threat Intelligence – TAXII (Preview)** and select **Open connector page**.
4. On the **Configuration** page, enter a **Friendly name (for server)** value such as the collection title. Enter the **API root URL** and **Collection ID** you want to import. Enter a username and password if required, and select **Add**.

You can connect your TAXII servers to Microsoft Sentinel using the built-in TAXII connector. For detailed configuration instructions, see the [full documentation](#).

Enter the following information and select **Add** to configure your TAXII server.

Friendly name (for server) *

API root URL *

Collection ID *

Username

Password

You see your connection under a list of configured TAXII 2.0 servers. Repeat the configuration for each collection you want to connect from the same or different TAXII servers.

Import threat indicators with the Platforms data connector

The `tiIndicators` API needs the **Application (client) ID**, **Directory (tenant) ID**, and **client secret** from your TIP or custom solution to connect and send threat indicators to

Microsoft Sentinel. To get this information, register the TIP or solution app in Microsoft Entra ID and grant it the needed permissions.

For more information, see [Connect your threat intelligence platform to Microsoft Sentinel](#).

Create an Analytics rule from a template

This example uses the rule template called **TI map IP entity to AzureActivity**, which compares any IP address-type threat indicators with all your Azure Activity IP address events. Any match generates a security alert and a corresponding incident for investigation by your security operations team.

The example assumes you've used one or both threat intelligence data connectors to import threat indicators and the Azure Activity data connector to import your Azure subscription-level events. You need both data types to use this analytics rule successfully.

1. In the [Azure portal](#), search for and select **Microsoft Sentinel**.
2. Select the workspace where you imported threat indicators with either threat intelligence data connector.
3. On the leftmost pane, select **Analytics**.
4. On the **Rule templates** tab, search for and select the rule **(Preview) TI map IP entity to AzureActivity**. Select **Create rule**.
5. On the first **Analytic rule wizard - Create new rule from template** page, make sure the rule **Status** is set to **Enabled**. Change the rule name or description if you want. Select **Next: Set rule logic**.

Analytic rule wizard - Create new rule from template

(Preview) TI map IP entity to AzureActivity

General Set rule logic Incident settings (Preview) Automated response Review and create

Create an analytic rule that will run on your data to detect threats.

Analytic rule details

Name *

(Preview) TI map IP entity to AzureActivity

Description

Identifies a match in AzureActivity from any IP IOC from TI

Tactics

Impact

Severity

Medium

Status

Enabled

Disabled

[Next : Set rule logic >](#)

The rule logic page contains the query for the rule, entities to map, rule scheduling, and the number of query results that generate a security alert. The template settings run once an hour. They identify any IP address IoCs that match any IP addresses from Azure events. They also generate security alerts for all matches. You can keep these settings or change any of them to meet your needs. When you're finished, select **Next: Incident settings (Preview)**.

6. Under **Incident settings (Preview)**, make sure that **Create incidents from alerts triggered by this analytics rule** is set to **Enabled**. Select **Next: Automated response**.

This step lets you configure automation to trigger when the rule generates a security alert. Automation in Microsoft Sentinel uses playbooks powered by Azure Logic Apps. For more information, see [Tutorial: Set up automated threat responses in Microsoft Sentinel](#). For this example, select **Next: Review**. After you review the settings, select **Create**.

Your rule activates immediately when it's created and then triggers on the regular schedule.

View and edit the Threat Intelligence Workbook

1. In the [Azure portal](#), search for and select **Microsoft Sentinel**.
2. Select the workspace where you've imported threat indicators with either threat intelligence data connector.
3. On the leftmost pane, select **Workbooks**.
4. Search for and select the workbook titled **Threat Intelligence**.
5. Make sure you have the necessary data and connections as shown. Select **Save**.

Threat Intelligence

MICROSOFT

Gain insights into threat indicators, including type and severity of threats, threat activity over time, and correlation with other data sources, including Office 365 and firewalls.

Required data types: ⓘ

- ✓ ThreatIntelligenceIndicator
- ✓ SecurityAlert

Relevant data connectors: ⓘ

- ThreatIntelligence
- ThreatIntelligenceTaxii

Threat Intelligence overview

Indicators imported into Sentinel by indicator type and date

Indicator Type	Count
1832	12.1
5.43	

Indicators imported into Sentinel by indicator provider and date

Indicator Provider	Count
2	

Active indicators by indicator type

Indicator Type	Count
226	1.9
1.32	

Active indicators by indicator provider

Indicator Provider	Count
230	

View template **Save**

In the pop-up window, select a location and then select **OK**. This step saves the workbook so that you can modify it and save your changes.

6. Select **View saved workbook** to open the workbook and see the default charts the template provides.

To edit the workbook, select **Edit**. You can select **Edit** next to any chart to edit the query and settings for that chart.

To add a new chart that shows threat indicators by threat type:

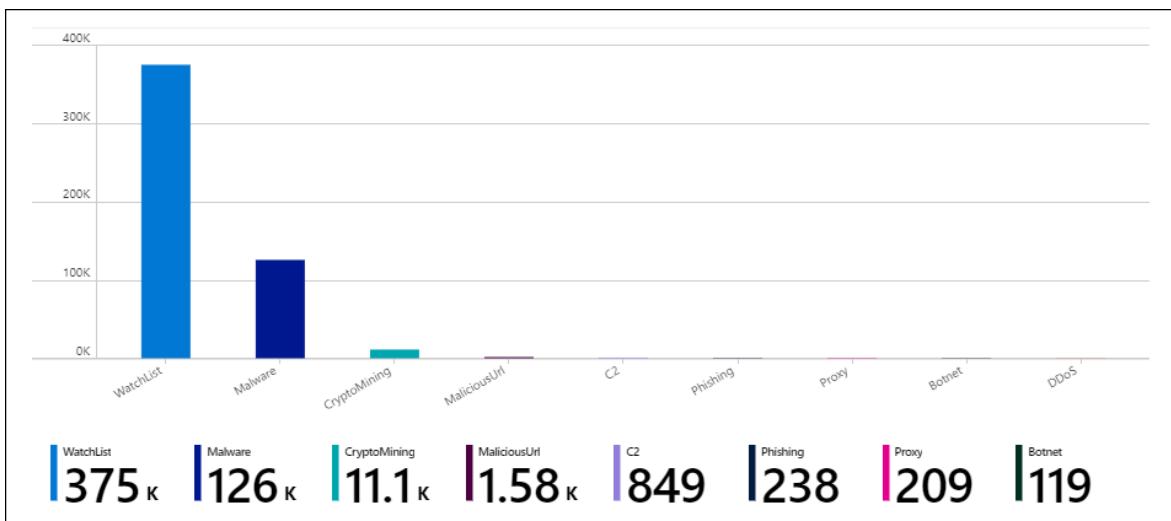
1. Select **Edit**. Scroll to the bottom of the page and select **Add > Add query**.

2. Under **Log Analytics workspace Logs Query**, enter the following query:

```
ThreatIntelligenceIndicator  
| summarize count() by ThreatType
```

3. Select **Bar chart** in the **Visualization** dropdown and select **Done editing**.

4. At the top of the page, select **Done editing**. Select the **Save** icon to save your new chart and workbook.



Next steps

Go to the [Microsoft Sentinel repo](#) on GitHub to see contributions by the community at large and by Microsoft. Here you find new ideas, templates, and conversations about all the feature areas of Microsoft Sentinel.

Microsoft Sentinel workbooks are based on Azure Monitor workbooks, so extensive documentation and templates are available. A great place to start is [Create interactive](#)

reports with Azure Monitor workbooks. There's collection of community driven [Azure Monitor Workbook Templates](#) on GitHub to download.

To learn more about the featured technologies, see:

- [What is Microsoft Sentinel?](#)
- [Quickstart: On-board Microsoft Sentinel](#)
- [Microsoft Graph Security Indicators API](#)
- [Tutorial: Investigate incidents with Microsoft Sentinel](#)
- [Tutorial: Set up automated threat responses in Microsoft Sentinel](#)

Related resources

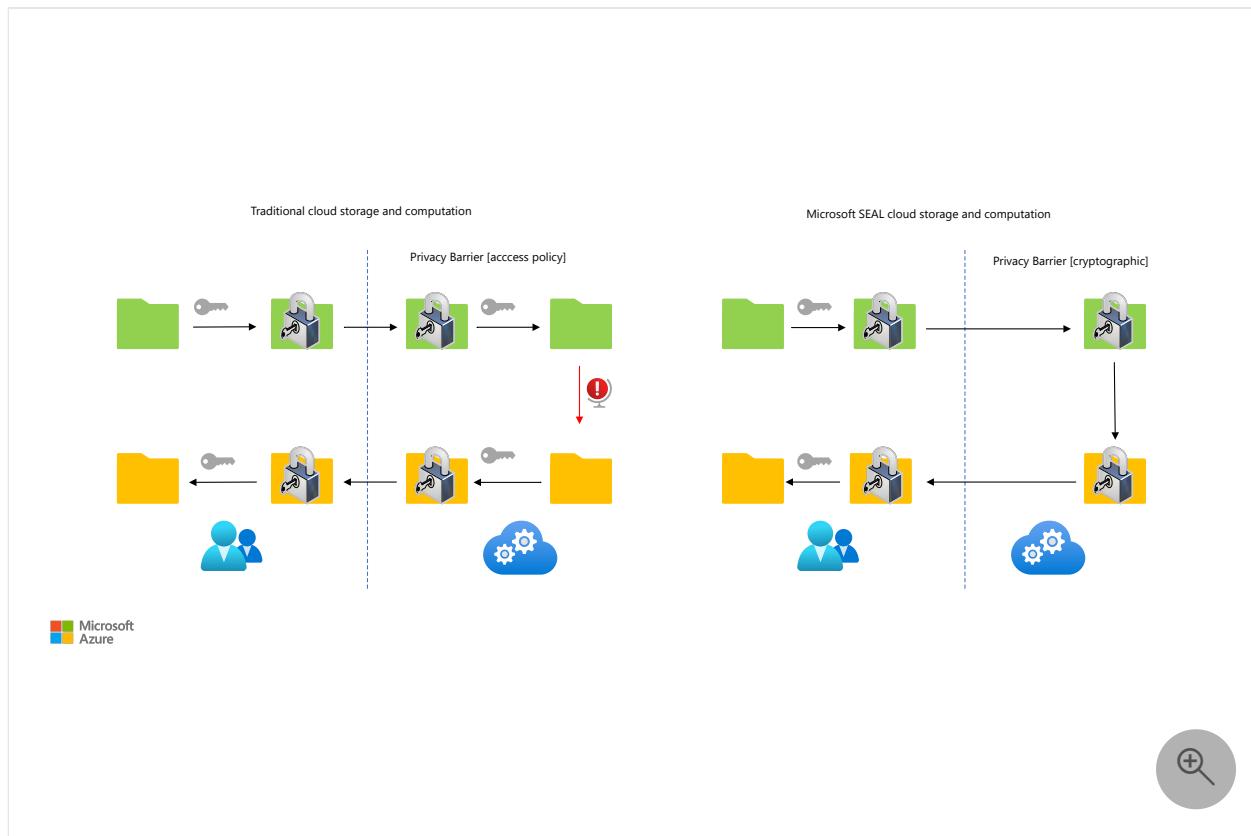
- [Automate Microsoft Sentinel integration with Azure DevOps](#)
- [Monitor hybrid security by using Microsoft Defender for Cloud and Microsoft Sentinel](#)
- [Azure security solutions for AWS](#)
- [Monitor hybrid security by using Microsoft Defender for Cloud and Microsoft Sentinel](#)

Homomorphic encryption with SEAL

.NET

This article discusses how and when to use homomorphic encryption and how to implement homomorphic encryption with the open-source [Microsoft Simple Encrypted Arithmetic Library \(SEAL\)](#).

Architecture



[Download a Visio file](#) of this architecture.

Workflow

Traditional encryption schemes consist of three functionalities: key generation, encryption, and decryption. *Symmetric-key* encryption schemes use the same secret key for both encryption and decryption. It enables efficient encryption of large amounts of data for secure, outsourced cloud storage. *Public-key* encryption schemes use a public key for encryption and a separate, secret key for decryption. Anyone who knows the public key can encrypt data, but only someone who knows the secret key can decrypt

and read the data. Public-key encryption enables secure online communication, but is typically less efficient than symmetric-key encryption.

You can use traditional encryption for secure storage and communication, but outsourced computation has required the removal of encryption layers. Cloud services that provide outsourced computation must implement access policies to prevent unauthorized access to the data and keys. Data privacy relies on the access control policies that are imposed by the cloud provider and trusted by the customer.

With [Microsoft SEAL](#) homomorphic encryption, cloud providers never have unencrypted access to the data they store and compute on. Computations can be performed directly on encrypted data. The results of such encrypted computations remain encrypted, and can be decrypted only by the data owner by using the secret key. Most homomorphic encryption uses public-key encryption schemes, although the public-key functionality may not always be needed.

Scenario details

Companies often send, receive, and store their cloud data in encrypted form. But to take advantage of cloud computing, companies must provide either unencrypted data or the keys to decrypt it. This practice puts company data at increased risk. *Homomorphic encryption* allows computation directly on encrypted data, making it easier to apply the potential of the cloud for privacy-critical data.

Potential use cases

- Lightweight computations like addition and multiplication on privacy-critical data and parts of programs.
- Outsourced cloud computing, where a single owner owns all the data and has sole access to the decryption keys.

Considerations

- Only some computations are possible on encrypted data. Microsoft SEAL homomorphic encryption library allows additions and multiplications on encrypted integers or real numbers. Encrypted comparison, sorting, or regular expressions aren't often feasible to evaluate on encrypted data using this technology. So only specific privacy-critical cloud computations on parts of programs can be implemented using Microsoft SEAL.

- Microsoft SEAL comes with two homomorphic encryption schemes with different properties. The *BFV scheme* allows modular arithmetic to be done on encrypted integers. The *CKKS scheme* allows additions and multiplications on encrypted real or complex numbers, but yields only approximate results. CKKS is the best choice when summing up encrypted real numbers, evaluating machine learning models on encrypted data, or computing distances of encrypted locations. For applications where exact values are necessary, the BFV scheme is the only choice.
- Homomorphic encryption isn't efficient. Because homomorphic encryption comes with a large performance overhead, computations that are already costly to do on unencrypted data probably aren't feasible on encrypted data.
- Data encrypted with homomorphic encryption is many times larger than unencrypted data, so it may not make sense to encrypt entire large databases, for example, with this technology. Instead, scenarios where strict privacy requirements prohibit unencrypted cloud computation, but the computations themselves are fairly lightweight, are meaningful use cases.
- Typically, homomorphic encryption schemes have a single secret key, which is held by the data owner. Homomorphic encryption isn't reasonable for scenarios where multiple different private data owners want to engage in collaborative computation.
- It isn't always easy or straightforward to translate an unencrypted computation into a computation on encrypted data. Even if new users can program and run a computation using Microsoft SEAL, there can be a great difference between efficient and inefficient implementation. It can be hard to know how to improve performance.
- While the homomorphic encryption primitive itself is secure, it doesn't guarantee that the apps and protocols that use it are secure.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Jose Contreras](#) | Principal Software Engineering Manager

Next steps

To learn more about homomorphic encryption and the Microsoft SEAL library, see [Microsoft SEAL](#) from Microsoft Research, and the [SEAL code project](#) on GitHub.

See the following resources about security in Azure:

- [Introduction to Azure security](#)
- [Azure security best practices](#)
- [Microsoft cloud security benchmark](#)
- [Overview of the security pillar](#)
- [Security in the Microsoft Cloud Adoption Framework](#)

Related resources

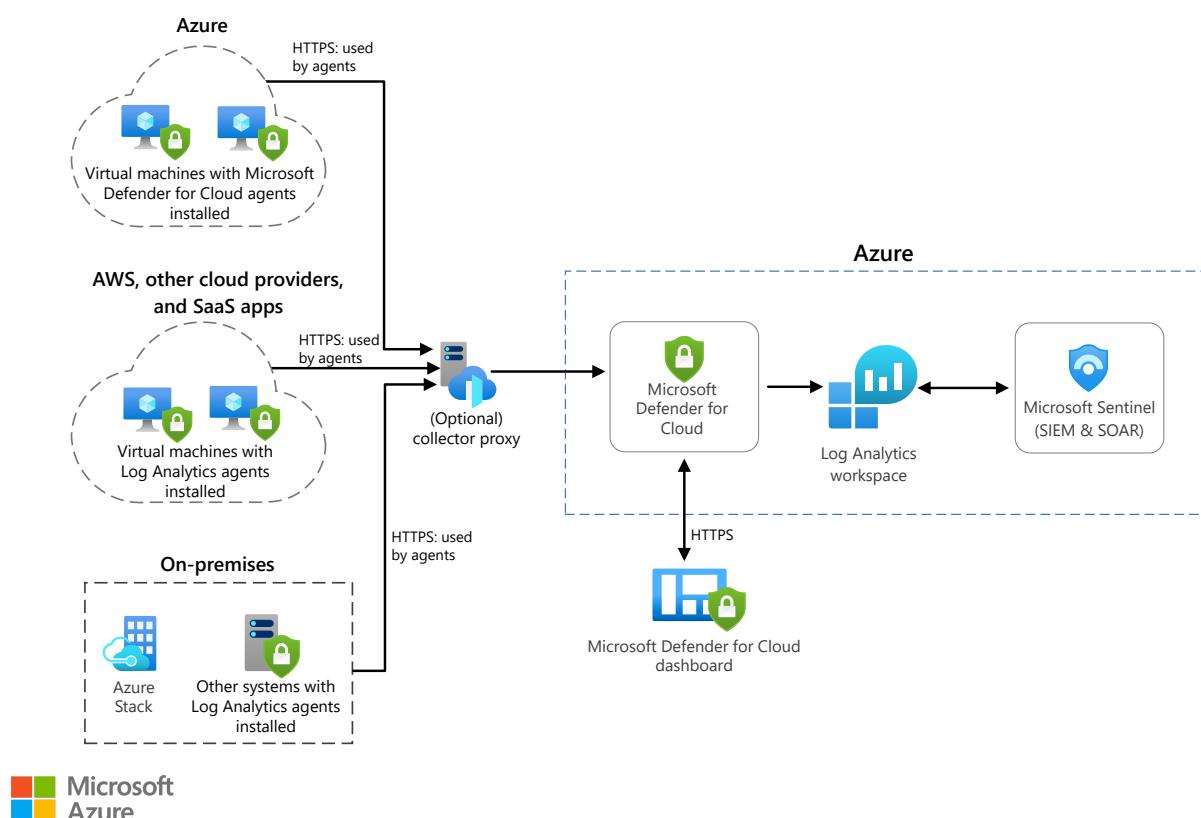
- [Centralized app configuration and security](#)
- [Confidential computing on a healthcare platform](#)
- [Secure research environment for regulated data](#)

Monitor hybrid security using Microsoft Defender for Cloud and Microsoft Sentinel

Azure Log Analytics Azure Monitor Microsoft Defender for Cloud Microsoft Sentinel Azure Stack

This reference architecture illustrates how to use Microsoft Defender for Cloud and Microsoft Sentinel to monitor the security configuration and telemetry of on-premises, Azure, and Azure Stack workloads.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

- **Microsoft Defender for Cloud.** This is an advanced, unified security-management platform that Microsoft offers to all Azure subscribers. Defender for Cloud is segmented as a cloud security posture management (CSPM) and cloud workload

protection platform (CWPP). CWPP is defined by workload-centric security protection solutions, which are typically agent-based. Microsoft Defender for Cloud provides threat protection for Azure workloads, both on-premises and in other clouds, including Windows and Linux virtual machines (VMs), containers, databases, and Internet of Things (IoT). When activated, the Log Analytics agent deploys automatically into Azure Virtual Machines. For on-premises Windows and Linux servers and VMs, you can manually deploy the agent, use your organization's deployment tool, such as Microsoft Endpoint Protection Manager, or utilize scripted deployment methods. Defender for Cloud begins assessing the security state of all your VMs, networks, applications, and data.

- **Microsoft Sentinel.** Is a cloud-native Security Information and Event Management (SIEM) and security orchestration automated response (SOAR) solution that uses advanced AI and security analytics to help you detect, hunt, prevent, and respond to threats across your enterprise.
- **Azure Stack.** Is a portfolio of products that extend Azure services and capabilities to your environment of choice, including the datacenter, edge locations, and remote offices. Azure Stack implementations typically utilize racks of four to sixteen servers that are built by trusted hardware partners and delivered to your datacenter.
- **Azure Monitor.** Collects monitoring telemetry from a variety of on-premises and Azure sources. Management tools, such as those in Microsoft Defender for Cloud and Azure Automation, also push log data to Azure Monitor.
- **Log Analytics workspace.** Azure Monitor stores log data in a Log Analytics workspace, which is a container that includes data and configuration information.
- **Log Analytics agent.** The Log Analytics agent collects monitoring data from the guest operating system and VM workloads in Azure, from other cloud providers, and from on-premises. The Log Analytics Agent supports Proxy configuration and, typically in this scenario, a Microsoft Operations Management Suite (OMS) Gateway acts as proxy.
- **On-premises network.** This is the firewall configured to support HTTPS egress from defined systems.
- **On-premises Windows and Linux systems.** Systems with the Log Analytics Agent installed.
- **Azure Windows and Linux VMs.** Systems on which the Microsoft Defender for Cloud monitoring agent is installed.

Components

- [Microsoft Defender for Cloud ↗](#)
- [Microsoft Sentinel ↗](#)

- [Azure Stack](#)
- [Azure Monitor](#)

Scenario details

Potential use cases

Typical uses for this architecture include:

- Best practices for integrating on-premises security and telemetry monitoring with Azure-based workloads
- Integrating Microsoft Defender for Cloud with Azure Stack
- Integrating Microsoft Defender for Cloud with Microsoft Sentinel

Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a specific requirement that overrides them.

Microsoft Defender for Cloud upgrade

This reference architecture uses **Microsoft Defender for Cloud** to monitor on-premises systems, Azure VMs, Azure Monitor resources, and even VMs hosted by other cloud providers. Details about Microsoft Defender for Cloud pricing can be found [here](#).

Customized Log Analytics Workspace

Microsoft Sentinel needs access to a Log Analytics workspace. In this scenario, you can't use the default Defender for Cloud Log Analytics workspace with Microsoft Sentinel. Instead, you create a customized workspace. Data retention for a customized workspace is based on the workspace pricing tier, and you can find pricing models for Monitor Logs [here](#).

Note

Microsoft Sentinel can run on workspaces in any general availability (GA) region of Log Analytics except the China and Germany (Sovereign) regions. Data that Microsoft Sentinel generates, such as incidents, bookmarks, and alert rules, which may contain some customer data sourced from these workspaces, is saved either in

Europe (for Europe-based workspaces), in Australia (for Australia-based workspaces), or in the East US (for workspaces located in any other region).

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

A **security policy** defines the set of controls that are recommended for resources within a specified subscription. In Microsoft Defender for Cloud, you define policies for your Azure subscriptions according to the security requirements of your company and the type of applications or data sensitivity for each subscription.

The security policies that you enable in Microsoft Defender for Cloud drive security recommendations and monitoring. To learn more about security policies, refer to [Strengthen your security policy with Microsoft Defender for Cloud](#). You can assign security policies in Microsoft Defender for Cloud only at the management or subscription group levels.

Note

Part one of the reference architecture details how to enable Microsoft Defender for Cloud to monitor Azure resources, on-premises systems, and Azure Stack systems.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

As previously described, costs beyond your Azure subscription can include:

1. Microsoft Defender for Cloud costs. For more information, refer to [Defender for Cloud pricing](#).

2. Azure Monitor workspace offers granularity of billing. For more information, refer to [Manage Usage and Costs with Azure Monitor Logs](#).
3. Microsoft Sentinel is a paid service. For more information, refer to [Microsoft Sentinel pricing](#).

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

Microsoft Defender for Cloud roles

Defender for Cloud assesses the configuration of your resources to identify security issues and vulnerabilities, and displays information related to a resource when you are assigned the role of owner, contributor, or reader for the subscription or resource group to which a resource belongs.

In addition to these roles, there are two specific Defender for Cloud roles:

- **Security Reader.** A user that belongs to this role has read only rights to Defender for Cloud. The user can observe recommendations, alerts, a security policy, and security states, but can't make changes.
- **Security Admin.** A user that belongs to this role has the same rights as the Security Reader, and also can update security policies, and dismiss alerts and recommendations. Typically, these are users that manage the workload.
- The security roles, **Security Reader** and **Security Admin**, have access only in Defender for Cloud. The security roles don't have access to other Azure service areas, such as storage, web, mobile, or IoT.

Microsoft Sentinel subscription

- To enable Microsoft Sentinel, you need contributor permissions to the subscription in which the Microsoft Sentinel workspace resides.
- To use Microsoft Sentinel, you need contributor or reader permissions on the resource group to which the workspace belongs.
- Microsoft Sentinel is a paid service. For more information, refer to [Microsoft Sentinel pricing](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale in an efficient manner to meet the demands that users place on it. For more information, see [Performance efficiency pillar overview](#).

The Log Analytics Agent for Windows and Linux is designed to have very minimal impact on the performance of VMs or physical systems.

Microsoft Defender for Cloud operational process won't interfere with your normal operational procedures. Instead, it passively monitors your deployments and provides recommendations based on the security policies you enable.

Deploy this scenario

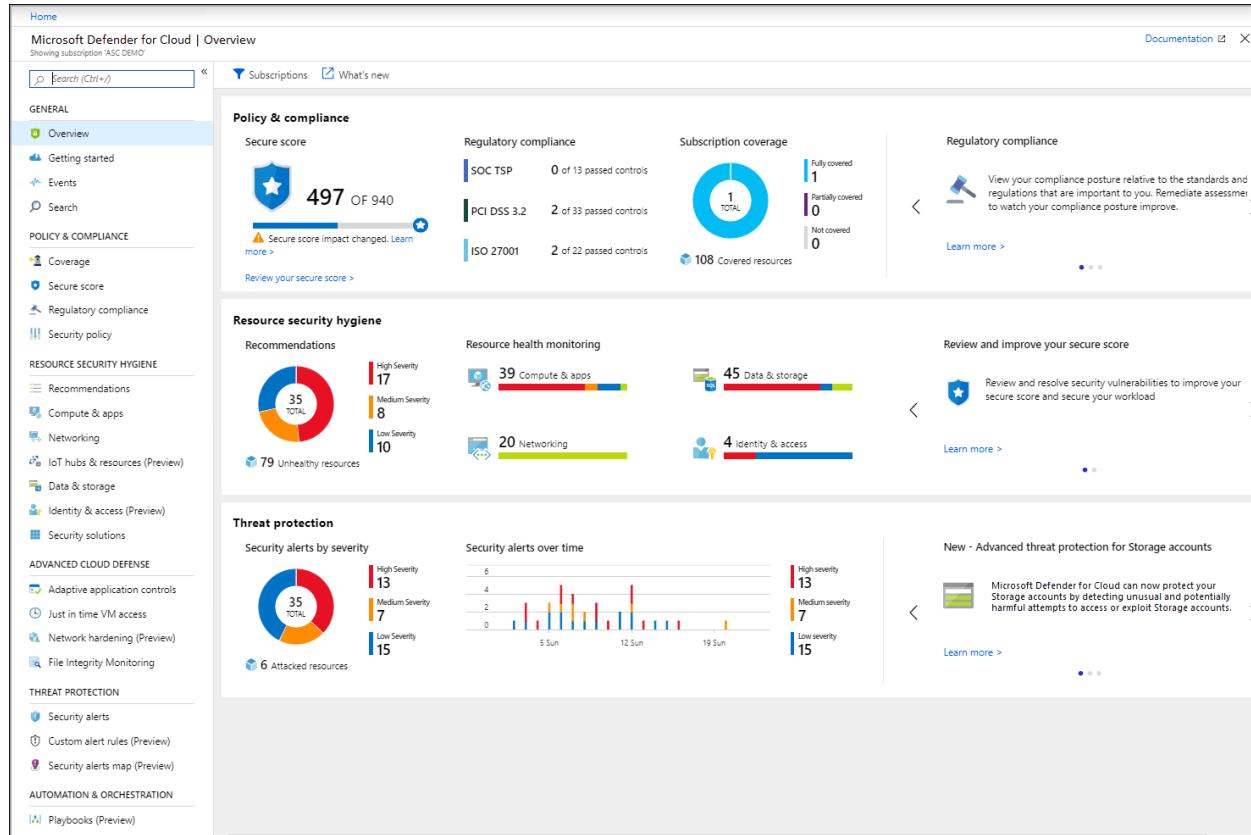
Create a Log Analytics workspace in the Azure portal

1. Sign into the Azure portal as a user with Security Admin privileges.
2. In the Azure portal, select **All services**. In the list of resources, enter **Log Analytics**. As you begin entering, the list filters based on your input. Select **Log Analytics workspaces**.
3. Select **Add** on the Log Analytics page.
4. Provide a name for the new Log Analytics workspace, such as **Defender for Cloud-SentinelWorkspace**. This name must be globally unique across all Azure Monitor subscriptions.
5. Select a subscription by selecting from the drop-down list if the default selection is not appropriate.
6. For **Resource Group**, choose to use an existing resource group or create a new one.
7. For **Location**, select an available geolocation.
8. Select **OK** to complete the configuration.

2 items		<input type="checkbox"/> Show hidden types	
<input type="checkbox"/>	NAME	TYPE	LOCATION
<input type="checkbox"/>	myAKSCluster	Kubernetes service	East US
<input type="checkbox"/>	ASC-SentinelWorkspace	Log Analytics	East US

Enable Defender for Cloud

While you're still signed into the Azure portal as a user with Security Admin privileges, select **Defender for Cloud** in the panel. **Defender for Cloud - Overview** opens:



The screenshot shows the Microsoft Defender for Cloud - Overview page. The left sidebar contains a navigation menu with sections like Home, Overview, Getting started, Events, Search, Policy & Compliance, Resource Security Hygiene, Advanced Cloud Defense, Threat Protection, and Automation & Orchestration. The main content area is divided into several sections: **Policy & compliance** (Secure score: 497 of 940, Regulatory compliance: SOC TSP 0/13, PCI DSS 3.2 2/33, ISO 27001 2/22, Subscription coverage: 108 resources, Fully covered 1, Partially covered 0, Not covered 0), **Resource security hygiene** (Recommendations: 35 total, 79 unhealthy resources, 17 High Severity, 8 Medium Severity, 10 Low Severity), **Threat protection** (Security alerts by severity: 35 total, 6 attacked resources, 13 High severity, 7 Medium severity, 15 Low severity), and **Regulatory compliance** (View your compliance posture relative to the standards and regulations that are important to you. Remediate assessments to watch your compliance posture improve). There are also sections for **Review and improve your secure score** and **New - Advanced threat protection for Storage accounts**.

Defender for Cloud automatically enables the Free tier for any of the Azure subscriptions not previously onboarded by you or another subscription user.

Upgrade Microsoft Defender for Cloud

1. On the Defender for Cloud main menu, select **Getting Started**.
2. Select the **Upgrade Now** button. Defender for Cloud lists your subscriptions and workspaces that are eligible for use.
3. You can select eligible workspaces and subscriptions to start your trial. Select the previously created workspace, **ASC-SentinelWorkspace**, from the drop-down menu.
4. In the Defender for Cloud main menu, select **Start trial**.
5. The **Install Agents** dialog box should display.
6. Select the **Install Agents** button. The **Defender for Cloud - Coverage** blade displays and you should observe your selected subscription.

Security Center - Coverage

Not covered Basic coverage Standard coverage

Looking good! The subscriptions below are fully protected.

3 SUBSCRIPTIONS

NAME	MY ROLE	OWNER	RESOURCES	ID
Contoso IT - demo	Reader	Display Owners	465	<Subscription ID>
QA-RomeCore-OMSTest2-Prod	Contributor	Display Owners	110	<Subscription ID>
ASC DEMO	Contributor	Display Owners	104	<Subscription ID>

You've now enabled automatic provisioning and Defender for Cloud will install the Log Analytics Agent for Windows (**HealthService.exe**) and the **omsagent** for Linux on all supported Azure VMs and any new ones that you create. You can turn off this policy and manually manage it, although we strongly recommend automatic provisioning.

To learn more about the specific Defender for Cloud features available in Windows and Linux, refer to [Feature coverage for machines](#).

Enable Microsoft Defender for Cloud monitoring of on-premises Windows computers

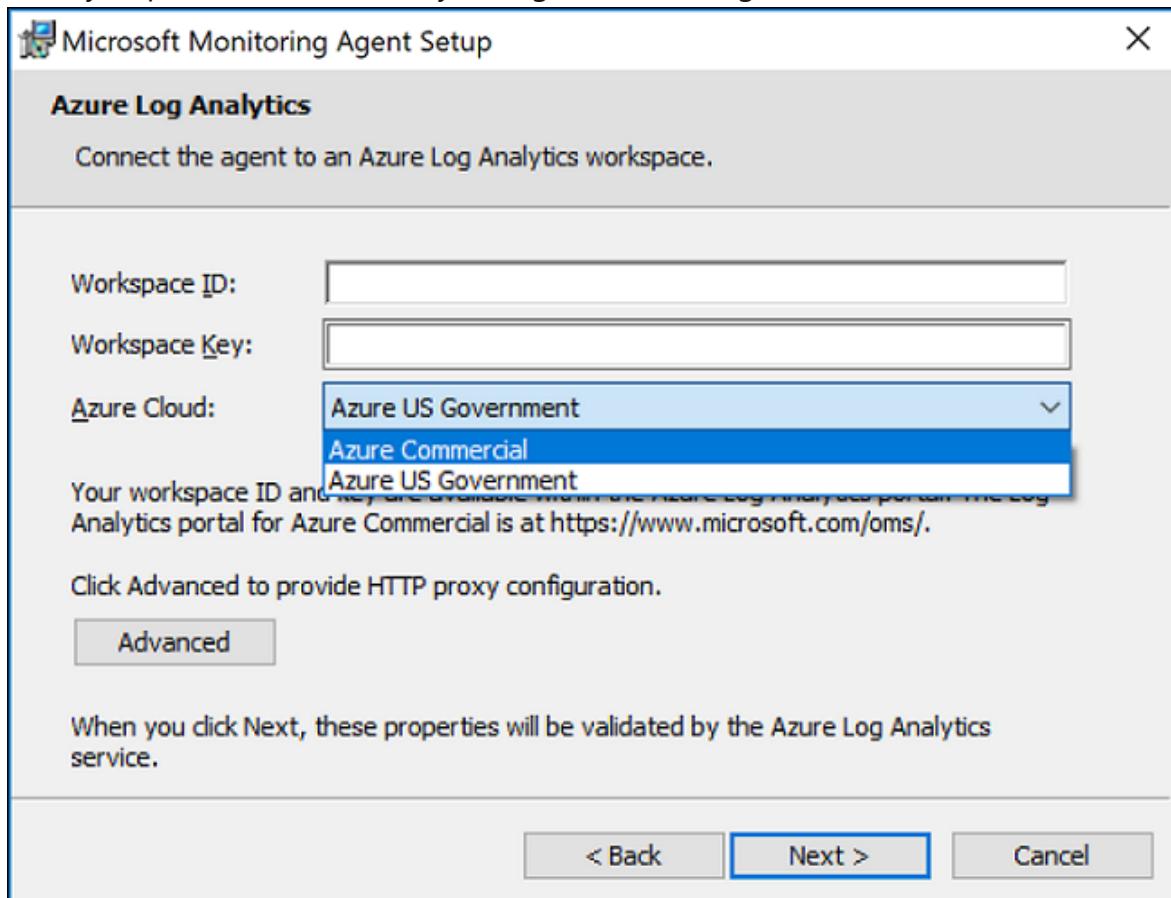
1. In the Azure portal on the **Defender for Cloud - Overview** blade, select the **Get Started** tab.
2. Select **Configure** under **Add new non-Azure computers**. A list of your Log Analytics workspaces displays, and should include the **Defender for Cloud-SentinelWorkspace**.
3. Select this workspace. The **Direct Agent** blade opens with a link for downloading a Windows agent and keys for your workspace identification (ID) to use when you configure the agent.
4. Select the **Download Windows Agent** link applicable to your computer processor type to download the setup file.
5. To the right of **Workspace ID**, select **Copy**, and then paste the ID into Notepad.
6. To the right of **Primary Key**, select **Copy**, and then paste the key into Notepad.

Install the Windows agent

To install the agent on the targeted computers, follow these steps.

1. Copy the file to the target computer and then **Run Setup**.
2. On the **Welcome** page, select **Next**.
3. On the **License Terms** page, read the license and then select **I Agree**.
4. On the **Destination Folder** page, change or keep the default installation folder and then select **Next**.

5. On the **Agent Setup Options** page, choose to connect the agent to Azure Log Analytics and then select **Next**.
6. On the **Azure Log Analytics** page, paste the **Workspace ID** and **Workspace Key (Primary Key)** that you copied into Notepad in the previous procedure.
7. If the computer should report to a Log Analytics workspace in Azure Government cloud, select **Azure US Government** from the **Azure Cloud** drop-down list. If the computer needs to communicate through a proxy server to the Log Analytics service, select **Advanced**, and then provide the proxy server's URL and port number.
8. After you provide the necessary configuration settings, select **Next**.



9. On the **Ready to Install** page, review your choices and then select **Install**.
10. On the **Configuration completed successfully** page, select **Finish**.

When complete, the Log Analytics agent appears in Windows Control Panel, and you can review your configuration and verify that the agent is connected.

For further information about installing and configuring the agent, refer to [Install Log Analytics agent on Windows computers](#).

The Log Analytics Agent service collects event and performance data, executes tasks, and other workflows defined in a management pack. Defender for Cloud extends its cloud workload protection platforms by integrating with **Microsoft Defender for Servers**. Together, they provide comprehensive endpoint detection and response (EDR) capabilities.

For more information about Microsoft Defender for Servers, refer to [Onboard servers to the Microsoft Defender for Servers service](#).

Enable Microsoft Defender for Cloud monitoring of on-premises Linux computers

1. Return to the **Getting Started** tab as previously described.
2. Select **Configure** under **Add new non-Azure computers**. A list of your Log Analytics workspaces displays. The list should include the **Defender for Cloud-SentinelWorkspace** that you created.
3. On the **Direct Agent** blade under **DOWNLOAD AND ONBOARD AGENT FOR LINUX**, select **copy** to copy the **wget** command.
4. Open Notepad and then paste this command. Save this file to a location that you can access from your Linux computer.

Note

On Unix and Linux operating systems, **wget** is a tool for non-interactive file downloading from the web. It supports HTTPS, FTPs, and proxies.

The Linux agent uses the Linux Audit Daemon framework. Defender for Cloud integrates functionalities from this framework within the Log Analytics agent, which enables audit records to be collected, enriched, and aggregated into events by using the Log Analytics Agent for Linux. Defender for Cloud continuously adds new analytics that use Linux signals to detect malicious behaviors on cloud and on-premises Linux machines.

For a list of the Linux alerts, refer to the [Reference table of alerts](#).

Install the Linux agent

To install the agent on the targeted Linux computers, follow these steps:

1. On your Linux computer, open the file that you previously saved. Select and copy the entire content, open a terminal console, and then paste the command.
2. Once the installation finishes, you can validate that the **omsagent** is installed by running the **pgrep** command. The command will return the **omsagent** process identifier (PID). You can find the logs for the agent at:
`/var/opt/microsoft/omsagent/"workspace id"/log/`.

It can take up to 30 minutes for the new Linux computer to display in Defender for Cloud.

Enable Microsoft Defender for Cloud monitoring of Azure Stack VMs

After you onboard your Azure subscription, you can enable Defender for Cloud to protect your VMs running on Azure Stack by adding the **Azure Monitor, Update and Configuration Management** VM extension from the Azure Stack marketplace. To do this:

1. Return to the **Getting Started** tab as previously described.
2. Select **Configure** under **Add new non-Azure computers**. A list of your Log Analytics workspaces displays, and it should include the **Defender for Cloud-SentinelWorkspace** that you created.
3. On the **Direct Agent** blade there is a link for downloading the agent and keys for your workspace ID to use during agent configuration. You don't need to download the agent manually. It'll be installed as a VM extension in the following steps.
4. To the right of **Workspace ID**, select **Copy**, and then paste the ID into Notepad.
5. To the right of **Primary Key**, select **Copy**, and then paste the key into Notepad.

Enable Defender for Cloud monitoring of Azure Stack VMs

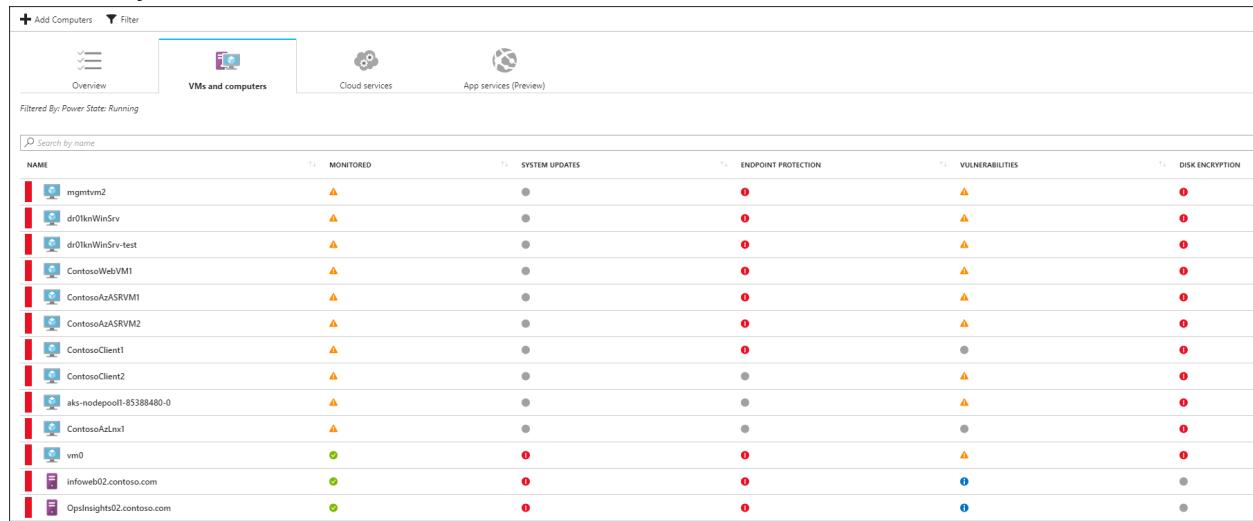
Microsoft Defender for Cloud uses the **Azure Monitor, Update and Configuration Management** VM extension bundled with Azure Stack. To enable the **Azure Monitor, Update and Configuration Management** extension, follow these steps:

1. In a new browser tab, sign into your **Azure Stack** portal.
2. Refer to the **Virtual machines** page, and then select the virtual machine that you want to protect with Defender for Cloud.
3. Select **Extensions**. The list of VM extensions installed on this VM displays.
4. Select the **Add** tab. The **New Resource** menu blade opens and displays the list of available VM extensions.
5. Select the **Azure Monitor, Update and Configuration Management** extension and then select **Create**. The **Install extension** configuration blade opens.
6. On the **Install extension** configuration blade, paste the **Workspace ID** and **Workspace Key (Primary Key)** that you copied into Notepad in the previous procedure.
7. When you finish providing the necessary configuration settings, select **OK**.
8. Once the extension installation completes, its status will display as **Provisioning Succeeded**. It might take up to one hour for the VM to appear in the Defender for Cloud portal.

For more information about installing and configuring the agent for Windows, refer to [Install the agent using setup wizard](#).

For troubleshooting issues for the Linux agent, refer to [How to troubleshoot issues with the Log Analytics agent for Linux](#).

Now you can monitor your Azure VMs and non-Azure computers in one place. **Azure Compute** provides you with an overview of all VMs and computers along with recommendations. Each column represents one set of recommendations, and the color represents the VMs or computers and the current security state for that recommendation. Defender for Cloud also provides any detections for these computers in security alerts.



NAME	MONITORED	SYSTEM UPDATES	ENDPOINT PROTECTION	VULNERABILITIES	DISK ENCRYPTION
mngtvm2	▲	●	●	▲	●
dr01knWinSrv	▲	●	●	▲	●
dr01knWinSrv-test	▲	●	●	▲	●
ContosoWebVM1	▲	●	●	▲	●
ContosoAzASRVM1	▲	●	●	▲	●
ContosoAzASRVM2	▲	●	●	▲	●
ContosoClient1	▲	●	●	●	●
ContosoClient2	▲	●	●	▲	●
aks-nodepool1-85388480-0	▲	●	●	▲	●
ContosoAzLn1	▲	●	●	●	●
vm0	●	●	●	▲	●
infoweb02.contoso.com	●	●	●	●	●
OpsInsights02.contoso.com	●	●	●	●	●

There are two types of icons represented on the **Compute** blade:



Note

Part two of the reference architecture will connect alerts from Microsoft Defender for Cloud and stream them into Microsoft Sentinel.

The role of Microsoft Sentinel is to ingest data from different data sources and perform data correlation across these data sources. Microsoft Sentinel leverages machine learning and AI to make threat hunting, alert detection, and threat responses smarter.

To onboard Microsoft Sentinel, you need to enable it, and then connect your data sources. Microsoft Sentinel comes with a number of connectors for Microsoft solutions, which are available out of the box and provide real-time integration, including Microsoft

Defender for Cloud, Microsoft Threat Protection solutions, Microsoft 365 sources (including Office 365), Microsoft Entra ID, Microsoft Defender for Servers, Microsoft Defender for Cloud Apps, and more. Additionally, there are built-in connectors to the broader security ecosystem for non-Microsoft solutions. You can also use Common Event Format, syslog, or the Representational State Transfer API to connect your data sources with Microsoft Sentinel.

Requirements for integrating Microsoft Sentinel with Microsoft Defender for Cloud

1. A Microsoft Azure Subscription
2. A Log Analytics workspace that isn't the default workspace created when you enable Microsoft Defender for Cloud.
3. Microsoft Defender for Cloud.

All three requirements should be in place if you worked through the previous section.

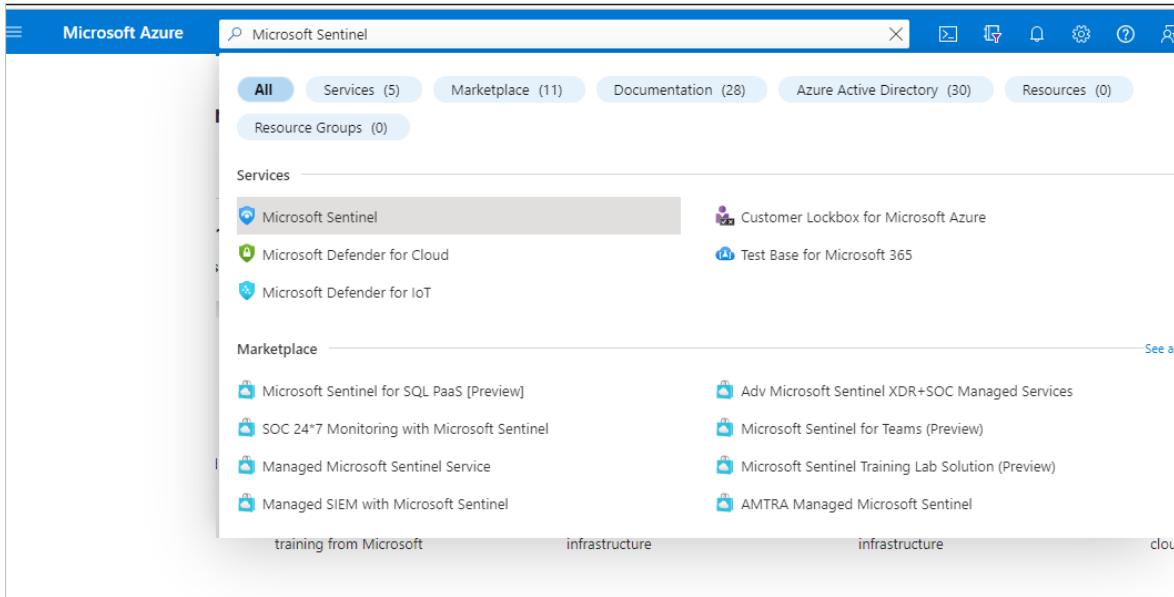
Global prerequisites

- To enable Microsoft Sentinel, you need contributor permissions to the subscription in which the Microsoft Sentinel workspace resides.
- To use Microsoft Sentinel, you need contributor or reader permissions on the resource group to which the workspace belongs.
- You might need additional permissions to connect specific data sources. You don't need additional permissions to connect to Defender for Cloud.
- Microsoft Sentinel is a paid service. For more information, refer to [Microsoft Sentinel pricing](#).

Enable Microsoft Sentinel

1. Sign into the Azure portal with a user that has contributor rights for **Defender for Cloud-Sentinelworkspace**.

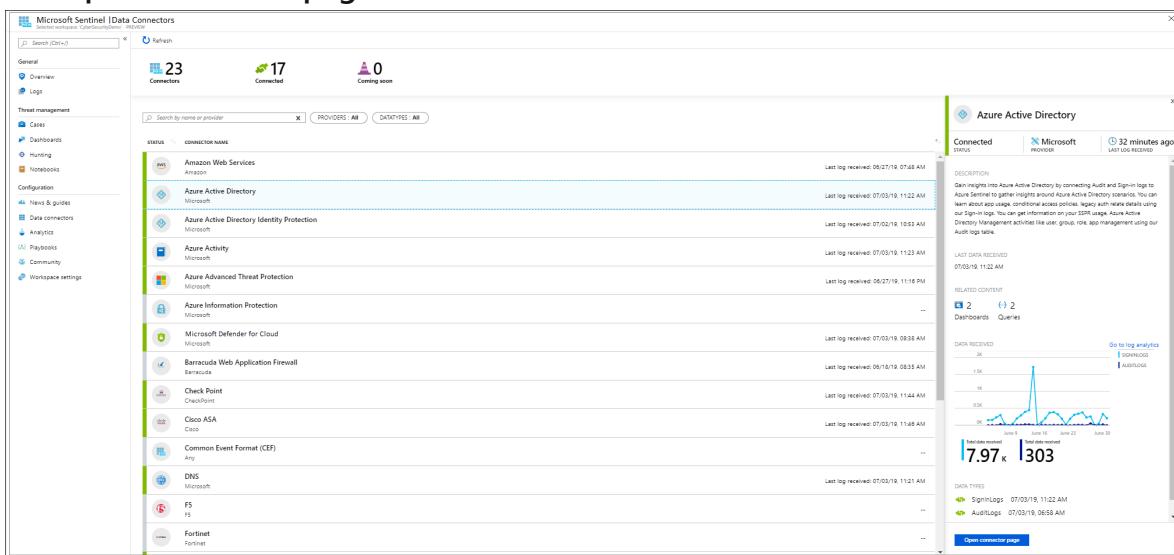
2. Search for and select Microsoft Sentinel.



The screenshot shows the Microsoft Azure search interface with the search term 'Microsoft Sentinel' entered. The results are filtered under the 'All' tab. The first result, 'Microsoft Sentinel', is highlighted with a gray box. Other results include 'Customer Lockbox for Microsoft Azure', 'Test Base for Microsoft 365', 'Microsoft Defender for Cloud', 'Microsoft Defender for IoT', 'Microsoft Sentinel for SQL PaaS [Preview]', 'SOC 24x7 Monitoring with Microsoft Sentinel', 'Managed Microsoft Sentinel Service', 'Managed SIEM with Microsoft Sentinel', 'Adv Microsoft Sentinel XDR+SOC Managed Services', 'Microsoft Sentinel for Teams (Preview)', 'Microsoft Sentinel Training Lab Solution (Preview)', and 'AMTRA Managed Microsoft Sentinel'. The results are categorized into 'training from Microsoft', 'infrastructure', 'infrastructure', and 'cloud'.

3. Select Add.

4. On the **Microsoft Sentinel** blade, select **Defender for Cloud-Sentinelworkspace**.
5. In Microsoft Sentinel, select **Data connectors** from the navigation menu.
6. From the data connectors gallery, select **Microsoft Defender for Cloud**, and select the **Open connector page** button.



The screenshot shows the Microsoft Sentinel Data Connectors blade for Microsoft Defender for Cloud. The left sidebar includes 'General', 'Threat management', 'Configuration', and 'Logs'. The main area shows 23 connectors, with 17 connected and 0 coming soon. A search bar and filters for 'PROVIDERS: ALL' and 'DATATYPES: ALL' are available. A detailed view of the 'Azure Active Directory' connector is shown on the right, with a 'DESCRIPTION' section, a 'LAST DATA RECEIVED' timestamp, and a 'RELATED CONTENT' section showing 2 dashboards and 2 queries. A graph at the bottom shows 'DATA RECEIVED' over time, with a peak of 7.97 k and 1303 entries.

7. Under **Configuration**, select **Connect** next to those subscriptions for which you want alerts to stream into Microsoft Sentinel. The **Connect** button will be available only if you have the required permissions and the Defender for Cloud subscription.
8. You should now observe the **Connection Status** as **Connecting**. After connecting, it will switch to **Connected**.
9. After confirming the connectivity, you can close Defender for Cloud **Data Connector** settings and refresh the page to observe alerts in Microsoft Sentinel. It might take some time for the logs to start syncing with Microsoft Sentinel. After you connect, you'll observe a data summary in the Data received graph and the connectivity status of the data types.
10. You can select whether you want the alerts from Microsoft Defender for Cloud to automatically generate incidents in Microsoft Sentinel. Under **Create incidents**,

select **Enabled** to turn on the default analytics rule that automatically creates incidents from alerts. You can then edit this rule under **Analytics**, in the **Active rules** tab.

11. To use the relevant schema in Log Analytics for the Microsoft Defender for Cloud alerts, search for **SecurityAlert**.

One advantage of using Microsoft Sentinel as your SIEM is that it provides data correlation across multiple sources, which enables you to have an end-to-end visibility of your organization's security-related events.

Note

To learn how to increase visibility in your data and identify potential threats, refer to [Azure playbooks on TechNet Gallery](#), which has a collection of resources including a lab in which you can simulate attacks. You should not use this lab in a production environment.

To learn more about Microsoft Sentinel, refer to the following articles:

- [Quickstart](#): Get started with Microsoft Sentinel
- [Tutorial](#): Detect threats out-of-the-box

Next steps

Azure Monitor

- [Azure Monitor](#)

Microsoft Defender for Cloud

- [Microsoft Defender for Cloud](#)
- [Microsoft Defender for Cloud Smart Alert Correlation](#)
- [Microsoft Defender for Cloud Connect Data](#)
- [Microsoft Defender for Cloud Coverage](#)
- [Microsoft Defender for Cloud Endpoint Protection](#)
- [Microsoft Defender for Cloud FAQ](#)
- [Microsoft Defender for Cloud Planning](#)
- [Microsoft Defender for Cloud Secure Score](#)
- [Microsoft Defender for Cloud Security Alerts](#)
- [Microsoft Defender for Cloud Security Policies](#)

- Microsoft Defender for Cloud Security Recommendations
- Microsoft Defender for Cloud Supported Platforms
- Microsoft Defender for Cloud Threat Protection
- Microsoft Defender for Cloud Tutorial

Microsoft Sentinel

- Microsoft Sentinel
- Microsoft Sentinel Analytics
- Microsoft Sentinel Attack Detection
- Microsoft Sentinel Connect Windows Firewall
- Microsoft Sentinel Connect Windows Security Events
- Microsoft Sentinel Data Sources
- Microsoft Sentinel Hunting
- Microsoft Sentinel Investigate
- Microsoft Sentinel Monitor
- Microsoft Sentinel Overview
- Microsoft Sentinel Permissions
- Microsoft Sentinel Quickstart

Azure Stack

- Azure Stack
- Azure Stack Automate Onboarding PowerShell
- Azure Stack Hub

Related resources

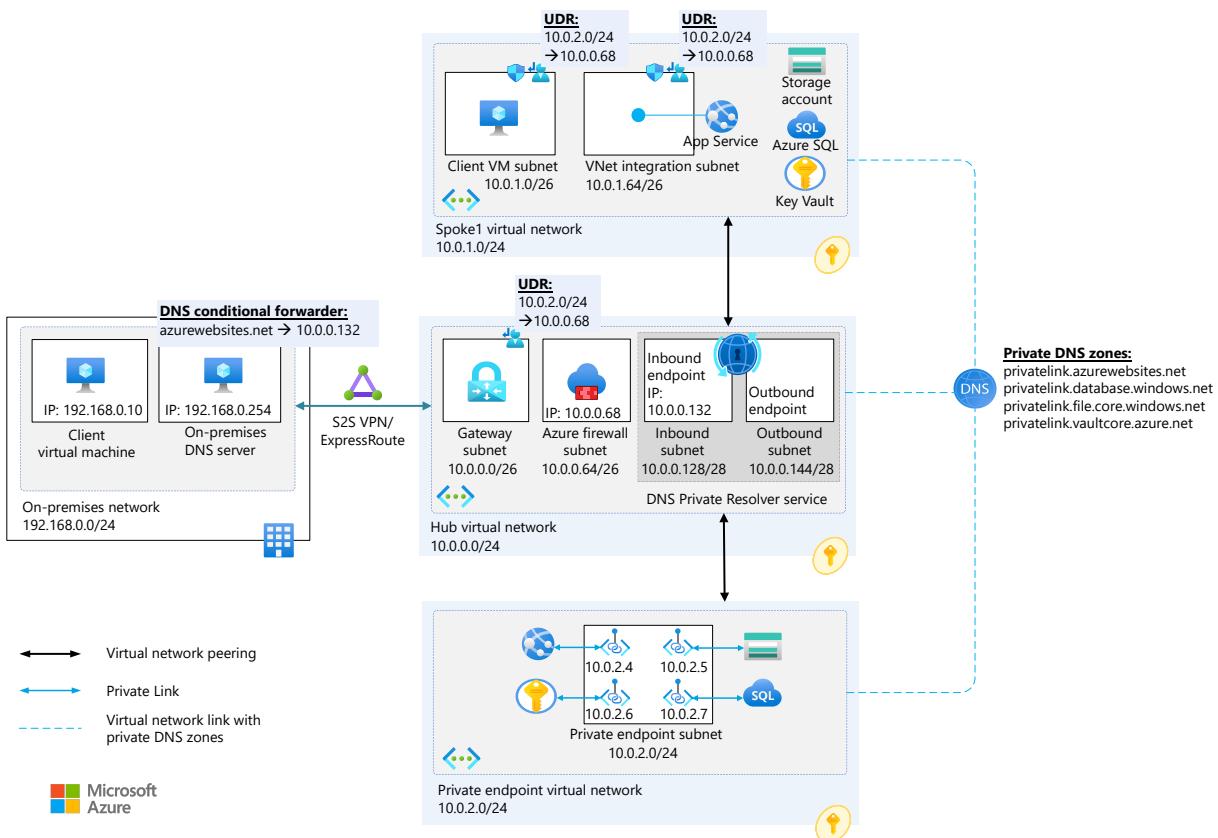
- Implement a secure hybrid network
- Enhanced-security hybrid messaging infrastructure — web access
- Centralized app configuration and security
- Automate Sentinel integration with Azure DevOps

Improved-security access to multitenant web apps from an on-premises network

Azure App Service Azure Virtual Network Azure Private Link Azure Key Vault Azure Storage Accounts

This article shows how to set up improved-security private connectivity to a multitenant web app or function app from an on-premises network or from within an Azure virtual network. It also shows how to set up improved-security connectivity between the app and other Azure PaaS services over Azure Private Link, without using the public internet.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

- By using Azure App Service [regional virtual network integration](#), the web app connects to Azure services through delegated subnet *VNet Integration Subnet* in an Azure virtual network.

- The *VNet Integration Subnet* and *Private Endpoint Subnet* networks are separate virtual networks in different subscriptions. Both networks are peered with *Hub Virtual Network* as part of a hub-and-spoke network configuration. For regional virtual network integration, the peered virtual networks must be in the same Azure region.
- [Azure Private Link](#) service sets up a [private endpoint](#) for the PaaS services, web apps, Azure SQL database, Azure storage account, and Azure key vault in *Private Endpoint Virtual Network*.

In this example, this virtual network is dedicated for the deployment of private endpoints only. No other resources, like virtual machines (VMs), will be deployed in this virtual network. The future demand to add private endpoints was taken into account when the subnet size was selected.

- The on-premises network and Azure virtual networks can be connected via [Site-to-Site \(S2S\) VPN](#) or [Azure ExpressRoute private peering](#). Users in the on-premises network access the app privately and with improved security over the private network only.

In this example, the on-premises network and Azure virtual networks are connected via ExpressRoute private peering.

- For an on-premises network that already has a Domain Name System (DNS) solution in place, the on-premises DNS solution is configured to forward DNS traffic to an Azure private DNS record (for example, `azurewebsites.net`) via a [conditional forwarder](#) that forwards the request to the DNS Private Resolver service's inbound endpoint that's deployed in Azure. DNS Private Resolver queries Azure DNS and receives information about the Azure Private DNS virtual network link. Then the resolution is done by the [private DNS zone linked to the virtual network](#).

Private DNS zones are also deployed in the same subscription as *Private Endpoint Virtual Network*.

In this example, a DNS forwarder machine at IP address 192.168.0.254 in the on-premises network forwards all DNS resolution requests to the hostname `azurewebsites.net` to the DNS Private Resolver service's inbound endpoint in Azure at address 10.0.0.132. Then the requests are resolved by the Azure-provided DNS service, which has IP address 168.63.129.16, via the Azure Private DNS zone that's linked to the virtual network.

An outbound endpoint is required to enable conditional forwarding name resolution from Azure to on-premises, other cloud providers, or external DNS servers, using a DNS forwarding ruleset.

Configuring a DNS forwarding ruleset isn't required for this scenario.

This app service configuration should be present:

[\[+\] Expand table](#)

Key	Value
WEBSITE_DNS_SERVER	168.63.129.16

- Virtual networks are linked to all the Azure private DNS zones.
 - The virtual network that has private endpoints is automatically linked to the private DNS zones. You need to link the other virtual networks separately.
- The web app communicates with the private endpoints of the PaaS services in *Private Endpoint Virtual Network* via Azure Firewall.
- On Azure Firewall, the [application rules](#) are configured to allow communication between *VNet Integration Subnet* and the private endpoints of PaaS resources. The target fully qualified domain names (FQDNs) are:
 - *.azurewebsites.net
 - *.database.windows.net
 - *.core.windows.net
 - *.vaultcore.azure.net
- Firewall and virtual network configuration for Azure SQL, Azure Storage Account, and Azure Key Vault allows traffic only from *VNet Integration Subnet*. The configuration doesn't allow communication with any other virtual network or with the public internet.

Components

- [Azure App Service](#) hosts web applications and function apps, allowing autoscale and high availability without requiring you to manage infrastructure.
- [Azure SQL Database](#) is a general-purpose relational-database managed service that supports relational data, spatial data, JSON, and XML.

- [Azure Storage account](#) provides a unique namespace for Azure Storage data that's accessible from anywhere in the world over HTTP or HTTPS. It contains all Azure Storage data objects: blobs, file shares, queues, tables, and disks.
- [Azure Key Vault](#) is a service for securely storing and accessing API keys, passwords, certificates, cryptographic keys, or any other secrets used by cloud apps and services.
- [Azure Virtual Network](#) is the fundamental building block for private networks in Azure. Azure resources like VMs can securely communicate with each other, the internet, and on-premises networks through virtual networks.
- [Azure Private Link](#) provides a private endpoint in a virtual network for connectivity to Azure PaaS services like Azure Storage and SQL Database, or to customer or partner services.
- [Azure ExpressRoute](#) private peering extends on-premises networks into the Microsoft cloud over a private connection. You could also establish Site-to-Site VPN between on-premises and the Azure network instead of using Azure ExpressRoute.
- [Azure Firewall](#) is a managed, cloud-based network security service that helps protect Azure Virtual Network resources.
- [Private DNS Zone](#) provides a reliable and secure DNS service for managing and resolving domain names in the virtual network.
- [DNS Private Resolver](#) enables the querying of [Azure DNS](#) private zones from an on-premises environment, and vice-versa, without deploying VM-based DNS servers.

Alternatives

For private connectivity, an alternative approach is to use [App Service Environment](#) to host the web application in an isolated environment. For the database, you can natively deploy [Azure SQL Managed Instance](#) in a virtual network, so you don't need VNet Integration or private endpoints. These offerings are typically more expensive because they provide single-tenant isolated deployment and other features.

If you have an App Service Environment but aren't using SQL Managed Instance, you can still use a private endpoint for private connectivity to an Azure SQL database. If you already have SQL Managed Instance but are using multitenant App Service, you can still use regional VNet Integration to connect to the SQL Managed Instance private address.

For some other Azure services, like Key Vault or Storage, there's no alternative to using private endpoints for highly secure and private connections from Web Apps.

Potential use cases

- Access a multitenant web app or function app privately with improved security over its [private endpoint](#) from an on-premises network or from within Azure virtual networks.
- Connect from a web app or function app to Azure platform as a service (PaaS) offerings:
 - Another web app
 - SQL Database
 - Azure Storage
 - Key Vault
 - Any other service that supports Azure private endpoints for inbound connectivity

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

Using Private Endpoint for your web app enables you to:

- Help secure your web app by configuring the private endpoint, eliminating public exposure.
- Connect with improved security to Web Apps from on-premises networks that connect to the virtual network by using a VPN or ExpressRoute private peering. Inbound connections to the web app are allowed from the on-premises network or from within the Azure virtual network only.
- Avoid any data exfiltration from your virtual network.

You can further improve the security of the inbound connection to the web app by fronting the app with a service like [Azure Application Gateway](#) or [Azure Front Door](#), optionally with [Azure Web Application Firewall](#). When you enable Private Endpoint for your web app, the [access restrictions](#) configuration of the web app isn't evaluated.

This scenario also improves security of the outbound connection from an App Service web app to a downstream dependency like a database, Storage, or Key Vault.

You can configure application routing to route either all traffic or only private traffic (also known as [RFC1918](#) traffic) into your virtual network. You configure this behavior by using the **Route All** setting. If **Route All** is disabled, the web app routes only private traffic into your virtual network. To block traffic to public addresses, enable the **Route All** setting to the virtual network. You can also use a [network security group](#) to block outbound traffic to resources in your virtual network or the internet. When **Route All** isn't enabled, NSGs are applied only to RFC1918 traffic.

In this example, the web app doesn't need to communicate with any service that isn't in the virtual network, so **Route All** is enabled.

An important security consideration in this scenario is the configuration of the firewall for PaaS resources.

SQL Database firewall options

Without using private connectivity, you can add [firewall rules](#) that allow inbound traffic from specified IP address ranges only. Another approach is to [allow Azure services](#) to access the server. This approach locks down the firewall to allow only traffic from within Azure. But this traffic includes all Azure regions and other customers.

You can also add a more restrictive firewall rule to allow only your app's [outbound IP addresses](#) to access the database. But because App Service is a multitenant service, these IP addresses are shared with and allow traffic from other customers on the same [deployment stamp](#), which uses the same outbound IP addresses.

Using private connectivity through the virtual network provides these firewall options to help prevent others from accessing the database:

- Create a [virtual network rule](#) that allows traffic only from the regional subnet delegated by VNet Integration, *VNet Integration Subnet* in this example. The delegated subnet must have a [service endpoint](#) configured for *Microsoft.Sql* so the database can identify traffic from that subnet.
- Configure the firewall to [deny public network access](#). Doing so turns off all other firewall rules and makes the database accessible only through its private endpoint.

The option of denying public network access is the most secure configuration. But if you use this option, database access is possible only via the virtual network that hosts the private endpoint. To connect to the database, anything other than the web app must have direct connectivity to the virtual network.

For example, deployments or urgent manual connections from SQL Server Management Studio (SSMS) on local machines can't reach the database except through VPN or ExpressRoute connectivity into the virtual network. You could also remotely connect to a VM in the virtual network and use SSMS from there. For exceptional situations, you could temporarily allow public network access and reduce risk by using other configuration options.

Storage Account and Key Vault firewall options

Storage accounts and key vaults have a public endpoint that's accessible from the internet. You can also create [private endpoints for your storage account](#) and [key vault](#). Doing so assigns these services a private IP address from your virtual network and helps to secure all traffic between your virtual network and the respective service over a private link.

When you create a private endpoint, *VNet Integration Subnet* can access the service privately and with improved security over a private link. But the storage account and key vault are still accessible from other Azure virtual networks. To block access from any other virtual network, create the service endpoint for this delegated subnet.

Availability

Private Link support for App Service, Azure SQL Database, Azure Storage, and Azure Key Vault is available in all public regions. To check availability in other regions, see [Azure Private Link availability](#)

Private Link introduces another component and availability consideration into the architecture. The Private Link service has a [high-availability SLA](#). You need to take this SLA into account when you calculate the composite SLA of the entire solution.

Scalability

For information about integrating Azure Private Link for PaaS services with Azure Private DNS zones in hub-and-spoke network architectures, see [Private Link and DNS integration at scale](#).

Global peering

Any service in any Azure region that can connect through the virtual network can reach the PaaS services' private endpoints, for example, through [virtual network peering](#) in

hub-and-spoke topologies. However, for App Service regional VNet Integration, the peered virtual networks must be located in the same Azure region.

Lack of global peering support means you can't use this solution for cross-region connectivity from App Service to a database or other private endpoint in another Azure region. For example, this solution wouldn't work for a multiregional deployment to support a partial failover, in which the web app remains active in one region but must connect to a failed-over database in another region, or vice versa. But other solutions exist for this situation.

If you need to connect Web Apps to a virtual network in another region, you can set up gateway-required VNet Integration. The limitation is that gateway-required VNet Integration can't be used with a virtual network connected with Azure ExpressRoute.

Logging and monitoring

Azure Private Link is integrated with [Azure Monitor](#), which allows you to see if data is flowing.

You can also use the connection troubleshoot service in Azure [Network Watcher](#) to trace the connectivity from a VM in a virtual network to the FQDN of the Private Endpoint resource.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

There's no added cost for App Service regional VNet Integration in supported pricing tiers in Basic, Standard, Premium v2, Premium v3, Isolated v2 App Service, and Azure Functions Premium plans.

Private endpoint is available for Windows web apps and Linux web apps, containerized or not, hosted on Basic, Standard, Premium v2, Premium v3, and Isolated v2 App Service plans, and also for function apps deployed to a Premium plan.

The Azure Private Link service that enables the private endpoints for PaaS services has an associated cost that's based on an hourly fee plus a premium on bandwidth. See the [Private Link pricing](#) page for details. Connections from a client virtual network to the Azure Firewall in the hub virtual network incur charges. You aren't charged for connections from Azure Firewall in the hub virtual network to private endpoints in a peered virtual network.

Azure Private DNS zone costs are based on the number of DNS zones hosted in Azure and the number of received DNS queries.

To explore the cost of running this scenario, see the [Azure pricing calculator estimate](#). All the services described in this article are preconfigured with reasonable default values for a small-scale application. To see how the pricing would change for your use case, change the appropriate variables to match your expected usage.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributor.

Principal author:

- [Ankit Singhal](#) | Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- See step-by-step guidance on how to [integrate Azure Functions with an Azure virtual network by using private endpoints](#).
- See the steps to configure [Azure Firewall application rules to inspect traffic destined to private endpoints in various network topologies](#).
- For more information on inbound and outbound scenarios for App Service, and which features to use in which case, see the [App Service networking features overview](#).
- For more information about private endpoints for Azure Web Apps, see [Using Private Endpoints for Azure Web Apps](#).
- For more information about integrating multitenant web apps with Azure Virtual Network, see [Integrate your app with an Azure virtual network](#).
- The FQDN of some of the PaaS services might resolve automatically to a public IP address. For information about overriding the DNS configuration to connect to the private endpoint, see [Azure Private Endpoint DNS configuration](#).

Related resources

- [Web app private connectivity to Azure SQL Database](#)
- [Tutorial: Integrate Azure Functions with an Azure virtual network by using private endpoints](#)

- Tutorial: Establish Azure Functions private site access
- Use Key Vault references for App Service and Azure Functions

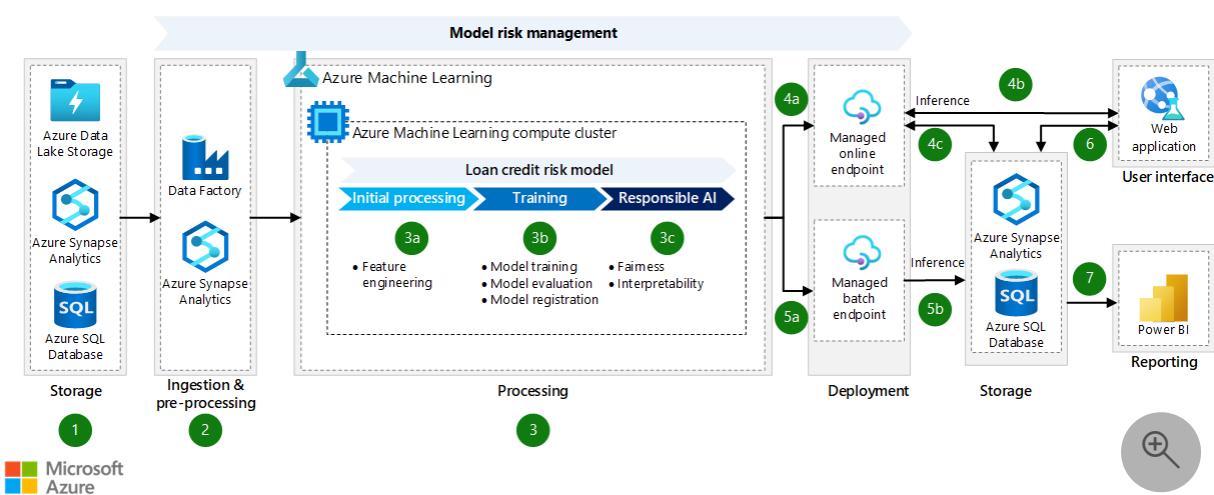
Model loan credit risk and default probability

Azure Machine Learning Azure Synapse Analytics Azure App Service Azure Data Lake Storage Power BI

This article describes an architecture that uses Azure Machine Learning to predict the delinquency and default probabilities of loan applicants. The model's predictions are based on the fiscal behavior of the applicant. The model uses a huge set of data points to classify applicants and provide an eligibility score for each applicant.

Apache®, [Spark](#), and the flame logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. No endorsement by the Apache Software Foundation is implied by the use of these marks.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The following dataflow corresponds to the preceding diagram:

1. Storage: Data is stored in a database like an Azure Synapse Analytics pool if it's structured. Older SQL databases can be integrated into the system. Semi-structured and unstructured data can be loaded into a data lake.
2. Ingestion and pre-processing: Azure Synapse Analytics processing pipelines and ETL processing can connect to data stored in Azure or third-party sources via built-

in connectors. Azure Synapse Analytics supports multiple analysis methodologies that use SQL, Spark, Azure Data Explorer, and Power BI. You can also use existing Azure Data Factory orchestration for the data pipelines.

3. Processing: Azure Machine Learning is used to develop and manage the machine learning models.

a. Initial processing: During this stage, raw data is processed to create a curated dataset that will train a machine learning model. Typical operations include data type formatting, imputation of missing values, feature engineering, feature selection, and dimensionality reduction.

b. Training: During the training stage, Azure Machine Learning uses the processed dataset to train the credit risk model and select the best model.

- Model training: You can use a range of machine learning models, including classical machine learning and deep learning models. You can use hyperparameter tuning to optimize model performance.
- Model evaluation: Azure Machine Learning assesses the performance of each trained model so you can select the best one for deployment.
- Model registration: You register the model that performs best in Azure Machine Learning. This step makes the model available for deployment.

c. Responsible AI: Responsible AI is an approach to developing, assessing, and deploying AI systems in a safe, trustworthy, and ethical way. Because this model infers an approval or denial decision for a loan request, you need to implement the principles of Responsible AI.

- *Fairness metrics* assess the effect of unfair behavior and enable mitigation strategies. Sensitive features and attributes are identified in the dataset and in cohorts (subsets) of the data. For more information, see [Model performance and fairness](#).
- *Interpretability* is a measure of how well you can understand the behavior of a machine learning model. This component of Responsible AI generates human-understandable descriptions of the model's predictions. For more information, see [Model interpretability](#).

4. Real-time machine learning deployment: You need to use real-time model inference when the request needs to be reviewed immediately for approval.

a. Managed machine learning online endpoint. For real-time scoring, you need to choose an appropriate compute target.

- b. Online requests for loans use real-time scoring based on input from the applicant form or loan application.
 - c. The decision and the input used for model scoring are stored in persistent storage and can be retrieved for future reference.
5. Batch machine learning deployment: For offline loan processing, the model is scheduled to be triggered at regular intervals.
- a. Managed batch endpoint. Batch inference is scheduled and the result dataset is created. Decisions are based on the creditworthiness of the applicant.
 - b. The result set of scoring from batch processing is persisted in the database or Azure Synapse Analytics data warehouse.
6. Interface to data about applicant activity: The details input by the applicant, the internal credit profile, and the model's decision are all staged and stored in appropriate data services. These details are used in the decision engine for future scoring, so they're documented.
- Storage: All details of credit processing are retained in persistent storage.
 - User interface: The approval or denial decision is presented to the applicant.
7. Reporting: Real-time insights about the number of applications processed and approve or deny outcomes are continuously presented to managers and leadership. Examples of reporting include near real-time reports of amounts approved, the loan portfolio created, and model performance.

Components

- [Azure Blob Storage](#) provides scalable object storage for unstructured data. It's optimized for storing files like binary files, activity logs, and files that don't adhere to a specific format.
- [Azure Data Lake Storage](#) is the storage foundation for creating cost-effective data lakes on Azure. It provides blob storage with a hierarchical folder structure and enhanced performance, management, and security. It services multiple petabytes of information while sustaining hundreds of gigabits of throughput.
- [Azure Synapse Analytics](#) is an analytics service that brings together the best of SQL and Spark technologies and a unified user experience for Azure Synapse Data Explorer and pipelines. It integrates with Power BI, Azure Cosmos DB, and Azure Machine Learning. The service supports both dedicated and serverless resource models and the ability to switch between those models.
- [Azure SQL Database](#) is an always up-to-date, fully managed relational database that's built for the cloud.

- [Azure Machine Learning](#) is a cloud service for managing machine learning project lifecycles. It provides an integrated environment for data exploration, model building and management, and deployment and supports code-first and low-code/no-code approaches to machine learning.
- [Power BI](#) is a visualization tool that provides easy integration with Azure resources.
- [Azure App Service](#) enables you to build and host web apps, mobile back ends, and RESTful APIs without managing infrastructure. Supported languages include .NET, .NET Core, Java, Ruby, Node.js, PHP, and Python.

Alternatives

You can use [Azure Databricks](#) to develop, deploy, and manage machine learning models and analytics workloads. The service provides a unified environment for model development.

Scenario details

Organizations in the financial industry need to predict the credit risk of individuals or businesses that request credit. This model evaluates the delinquency and default probabilities of loan applicants.

Credit risk prediction involves deep analysis of population behavior and classification of the customer base into segments based on fiscal responsibility. Other variables include market factors and economic conditions, which have a significant influence on results.

Challenges. Input data includes tens of millions of customer profiles and data about customer credit behavior and spending habits that's based on billions of records from disparate systems, like internal customer activity systems. The third-party data about economic conditions and the country/region's market analysis can come from monthly or quarterly snapshots that require the loading and maintenance of hundreds of GBs of files. Credit bureau information about the applicant or semi-structured rows of customer data, and cross joins between these datasets and quality checks to validate the integrity of the data, are needed.

The data usually consists of wide-column tables of customer information from credit bureaus together with market analysis. The customer activity consists of records with dynamic layout that might not be structured. Data is also available in free-form text from the customer service notes and applicant-interaction forms.

Processing these large volumes of data and ensuring the results are current requires streamlined processing. You need a low-latency storage and retrieval process. The data

infrastructure should be able to scale to support disparate data sources and provide the ability to manage and secure the data perimeter. The machine learning platform needs to support the complex analysis of the many models that are trained, tested, and validated across many population segments.

Data sensitivity and privacy. The data processing for this model involves personal data and demographic details. You need to avoid the profiling of populations. Direct visibility to all personal data must be restricted. Examples of personal data include account numbers, credit card details, social security numbers, names, addresses, and postal codes.

Credit card and bank account numbers must always be obfuscated. Certain data elements need to be masked and always encrypted, providing no access to the underlying information, but available for analysis.

Data needs to be encrypted at rest, in transit, and during processing via secure enclaves. Access to data items is logged in a monitoring solution. The production system needs to be set up with appropriate CI/CD pipelines with approvals that trigger model deployments and processes. Audit of the logs and workflow should provide the interactions with the data for any compliance needs.

Processing. This model requires high computational power for analysis, contextualizing, and model training and deployment. Model scoring is validated against random samples to ensure that credit decisions don't include any race, gender, ethnic, or geographic-location bias. The decision model needs to be documented and archived for future reference. Every factor that's involved in the decision outcomes is stored.

Data processing requires high CPU usage. It includes SQL processing of structured data in DB and JSON format, Spark processing of the data frames, or big data analytics on terabytes of information in various document formats. Data ELT/ETL jobs are scheduled or triggered at regular intervals or in real time, depending on the value of most recent data.

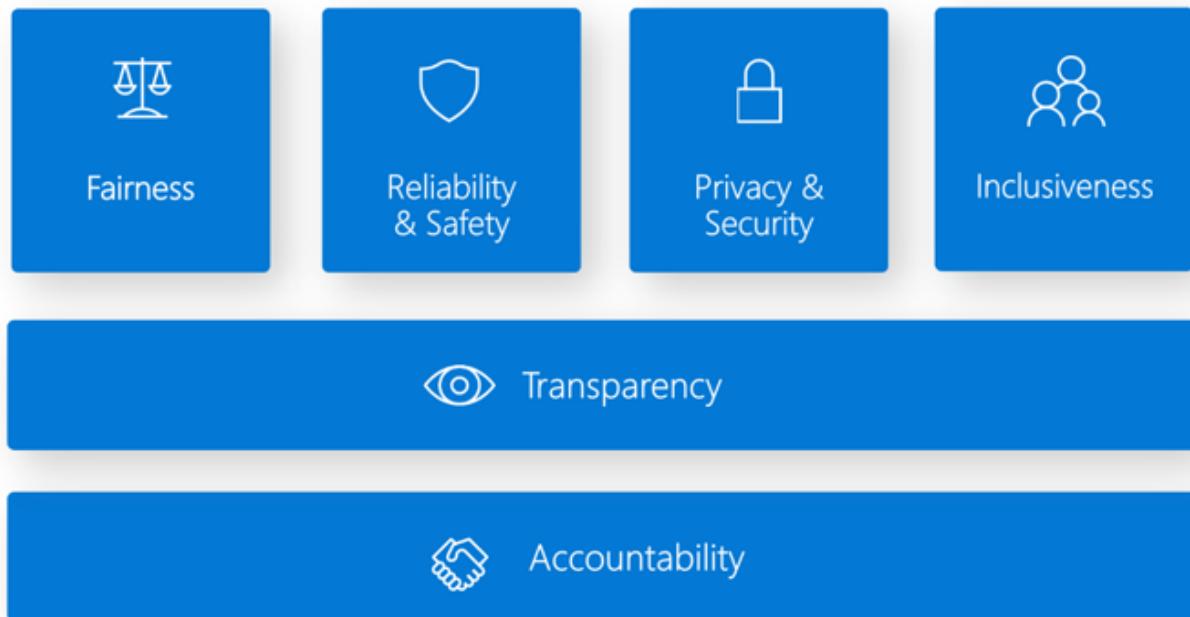
Compliance and regulatory framework. Every detail of loan processing needs to be documented, including the submitted application, the features used in model scoring, and the result set of the model. Model training information, data used for training, and training results should be registered for future reference and audit and compliance requests.

Batch versus real-time scoring. Certain tasks are proactive and can be processed as batch jobs, like pre-approved balance transfers. Some requests, like online credit line increases, require real-time approval.

Real-time access to the status of online loan requests must be available to the applicant. The loan-issuing financial institution continuously monitors the performance of the credit model and needs insight into metrics like loan-approval status, number of approved loans, dollar amounts issued, and the quality of new loan originations.

Responsible AI

The [Responsible AI dashboard](#) provides a single interface for multiple tools that can help you implement Responsible AI. The Responsible AI Standard is based on six principles:



Fairness and inclusiveness in Azure Machine Learning. This component of the Responsible AI dashboard helps you evaluate unfair behaviors by avoiding harms of allocation and harms of quality-of-service. You can use it to assess fairness across sensitive groups defined in terms of gender, age, ethnicity, and other characteristics. During assessment, fairness is quantified via disparity metrics. You should implement the mitigation algorithms in the [Fairlearn](#) open-source package, which use parity constraints.

Reliability and safety in Azure Machine Learning. The error analysis component of Responsible AI can help you:

- Gain a deep understanding of how failure is distributed for a model.
- Identify cohorts of data that have a higher error rate than the overall benchmark.

Transparency in Azure Machine Learning. A crucial part of transparency is understanding how features affect the machine learning model.

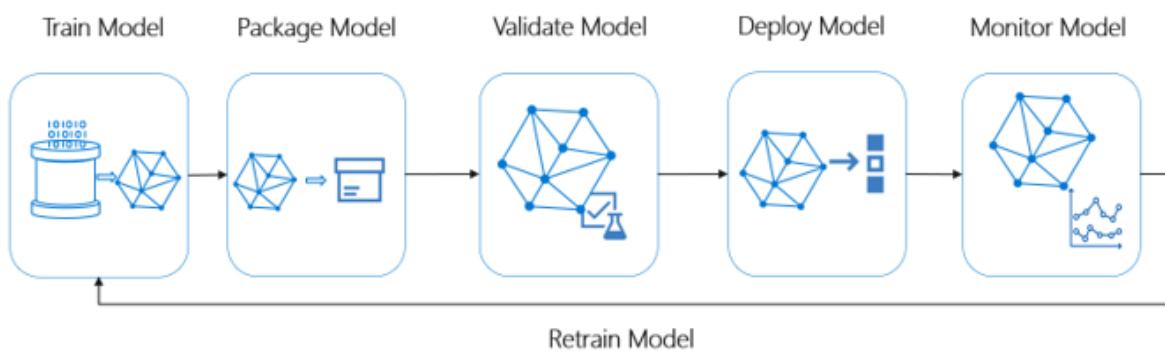
- *Model interpretability* helps you understand what influences the behavior of the model. It generates human-understandable descriptions of the model's predictions. This understanding helps to ensure that you can trust the model and helps you debug and improve it. [InterpretML](#) can help you understand the structure of glass-box models or the relationship among features in black-box deep neural network models.
- *Counterfactual what-if* can help you understand and debug a machine learning model in terms of how it reacts to feature changes and perturbations.

Privacy and security in Azure Machine Learning. Machine learning administrators need to create a secure configuration to develop and manage the deployment of models. [Security and governance features](#) can help you comply with your organization's security policies. Other tools can help you assess and secure your models.

Accountability in Azure Machine Learning. Machine learning operations (MLOps) is based on DevOps principles and practices that increase the efficiency of AI workflows. Azure Machine Learning can help you implement MLOps capabilities:

- Register, package, and deploy models
- Get notifications and alerts to changes in models
- Capture governance data for the end-to-end lifecycle
- Monitor applications for operational problems

This diagram illustrates the MLOps capabilities of Azure Machine Learning:



Potential use cases

You can apply this solution to the following scenarios:

- Finance: Get financial analysis of customers or cross-sales analysis of customers for targeted marketing campaigns.
- Healthcare: Use patient information as input to suggest treatment offerings.
- Hospitality: Create a customer profile to suggest offerings for hotels, flights, cruise packages, and memberships.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that you can use to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

Azure solutions provide defense in depth and a Zero Trust approach.

Consider implementing the following security features in this architecture:

- Deploy dedicated Azure services into virtual networks
- Azure SQL Database security capabilities
- Secure the credentials in data factory by using Key Vault
- Enterprise security and governance for Azure Machine Learning
- Azure security baseline for Synapse Analytics Workspace

Cost optimization

Cost optimization is about reducing unnecessary expenses and improving operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To estimate the cost of implementing this solution, use the [Azure pricing calculator](#).

Also consider these resources:

- [Plan and manage costs for Azure Synapse Analytics](#)
- [Plan and manage costs for Azure Machine Learning](#)

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

Machine learning solutions need to be scalable and standardized for easier management and maintenance. Ensure that your solution supports ongoing inference with retraining cycles and automated redeployments of models.

For more information, see [Azure MLOps \(v2\) solution accelerator](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

- For more information about designing scalable solutions, see [Performance efficiency checklist](#).
- For information about regulated industries, see [Scale AI and machine learning initiatives in regulated industries](#).
- Manage your Azure Synapse Analytics environment with [SQL](#), [Spark](#), or [serverless SQL](#) pools.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Charitha Basani](#) | Senior Cloud Solution Architect

Other contributor:

- [Mick Alberts](#) | Technical Writer

To see non-public LinkedIn profiles, sign into LinkedIn.

Next steps

- [Azure security baseline for Azure Machine Learning](#)
- [Azure Synapse Analytics](#)
- [Deploy machine learning models to Azure](#)
- [What is Responsible AI?](#)

Related resources

- [MLOps framework](#)
- [Responsible AI](#)
- [Responsible and trusted AI](#)

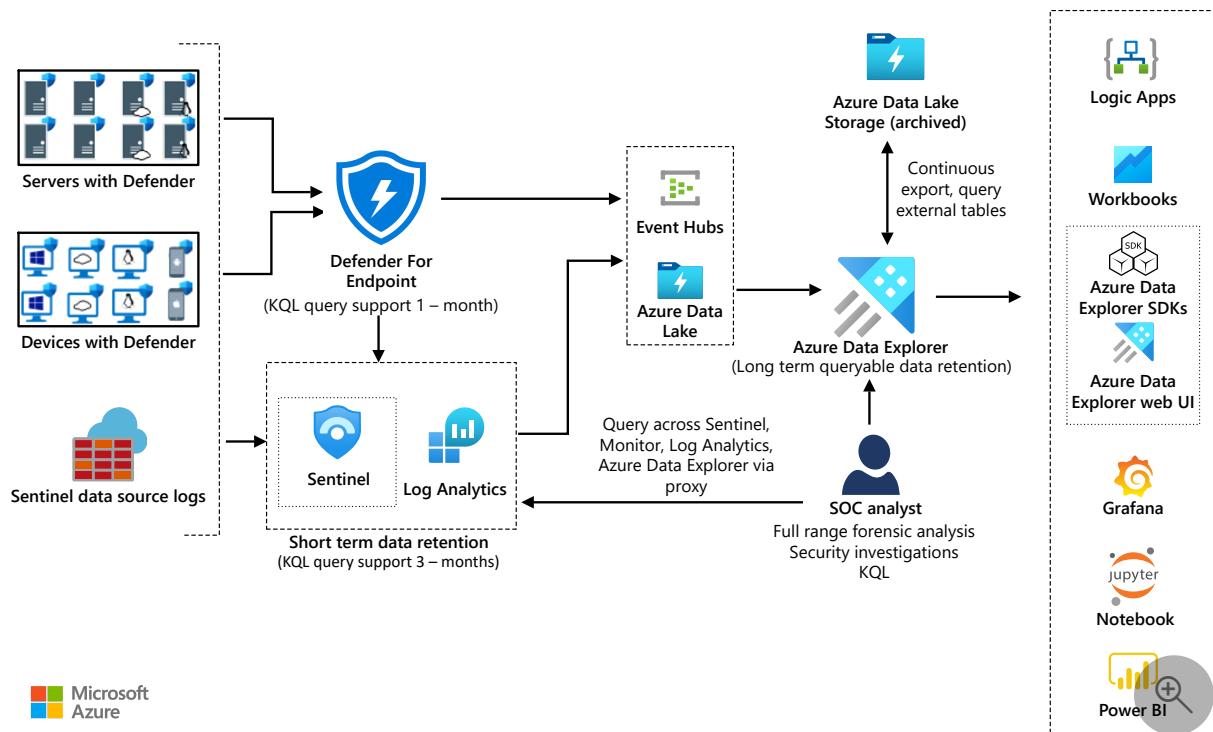
Long-term security log retention with Azure Data Explorer

Azure Data Explorer Azure Data Lake Storage Azure Event Hubs Azure Log Analytics Microsoft Sentinel

This solution stores security logs in Azure Data Explorer on a long-term basis. This solution minimizes costs and provides easy access when you need to query the data.

Grafana [↗](#) and *Jupyter Notebooks* [↗](#) are trademarks of their respective companies. No endorsement is implied by the use of these marks.

Architecture



Download a [Visio file](#) [↗](#) of this architecture.

Dataflow

1. For SIEM and SOAR, an enterprise uses Sentinel and Defender for Endpoint.
2. Defender for Endpoint uses native functionality to export data to Azure Event Hubs and Azure Data Lake. Sentinel ingests Defender for Endpoint data to monitor devices.

3. Sentinel uses Log Analytics as a data platform for exporting data to Event Hubs and Azure Data Lake.
4. Azure Data Explorer uses connectors for [Event Hubs](#), [Azure Blob Storage](#), and [Azure Data Lake Storage](#) to ingest data with low-latency and high throughput. This process uses [Azure Event Grid](#), which triggers the Azure Data Explorer ingestion pipeline.
5. If needed, Azure Data Explorer continuously exports security logs to Azure Storage. These logs are in compressed, partitioned Parquet format and are ready to be queried.
6. To follow regulatory requirements, Azure Data Explorer exports pre-aggregated data to Data Lake Storage for archiving.
7. Log Analytics and Sentinel support cross-service queries with Azure Data Explorer. SOC analysts use this capability to run full-range investigations on security data.
8. Azure Data Explorer provides native capabilities for processing, aggregating, and analyzing data.
9. Various tools provide near real-time analytics dashboards that quickly deliver insights:
 - [Azure Data Explorer dashboards](#)
 - [Power BI](#)
 - [Grafana](#)

Components

- [Defender for Endpoint](#)  protects organizations from threats across devices, identities, apps, email, data, and cloud workloads.
- [Sentinel](#) is a cloud-native SIEM and SOAR solution. It uses advanced AI and security analytics to detect, hunt, prevent, and respond to threats across enterprises.
- [Monitor](#)  is a software as a service (SaaS) solution that collects and analyzes data on environments and Azure resources. This data includes app telemetry, such as performance metrics and activity logs. Monitor also offers alerting functionality.
- [Log Analytics](#) is a Monitor service that you can use to query and inspect Monitor log data. Log Analytics also provides features for charting and statistically analyzing query results.

- [Event Hubs](#) is a fully managed, real-time data ingestion service that's straightforward and scalable.
- [Data Lake Storage](#) is a scalable storage repository that holds a large amount of data in the data's native, raw format. This data lake is built on top of [Blob Storage](#) and provides functionality for storing and processing data.
- [Azure Data Explorer](#) is a fast, fully managed, and highly scalable data analytics platform. You can use this cloud service for real-time analysis on large volumes of data. Azure Data Explorer is optimized for interactive, ad-hoc queries. It can handle diverse data streams from applications, websites, IoT devices, and other sources.
- [Azure Data Explorer dashboards](#) natively import data from Azure Data Explorer Web UI queries. These optimized dashboards provide a way to display and explore query results.

Alternatives

- Instead of using Azure Data Explorer for long-term storage of security logs, you can use Storage. This approach simplifies the architecture and can help control the cost. A disadvantage is the need to rehydrate the logs for security audits and interactive investigative queries. With Azure Data Explorer, you can move data from the cold partition to the hot partition by changing a policy. This functionality speeds up data exploration.
- Another option with this solution is to send all data, regardless of its security value, to Sentinel and Azure Data Explorer at the same time. Some duplication results, but the cost savings can be significant. Because Azure Data Explorer provides long-term storage, you can reduce your Sentinel retention costs with this approach.
- Log Analytics doesn't currently support exporting custom log tables. In this scenario, you can use Azure Logic Apps to export data from Log Analytics workspaces. For more information, see [Archive data from Log Analytics workspace to Azure Storage using Logic Apps](#).

Scenario details

Security logs are useful for identifying threats and tracing unauthorized attempts to access data. Security attacks can begin well before they're discovered. As a result, having access to long-term security logs is important. Querying long-term logs is critical for identifying the impact of threats and investigating the spread of illicit access attempts.

This article outlines a solution for long-term retention of security logs. At the core of the architecture is Azure Data Explorer. This service provides storage for security data at minimal cost but keeps that data in a format that you can query. Other main components include:

- Microsoft Defender for Endpoint and Microsoft Sentinel, for these capabilities:
 - Comprehensive endpoint security
 - Security information and event management (SIEM)
 - Security orchestration automated response (SOAR)
- Log Analytics, for short-term storage of Sentinel security logs.

Potential use cases

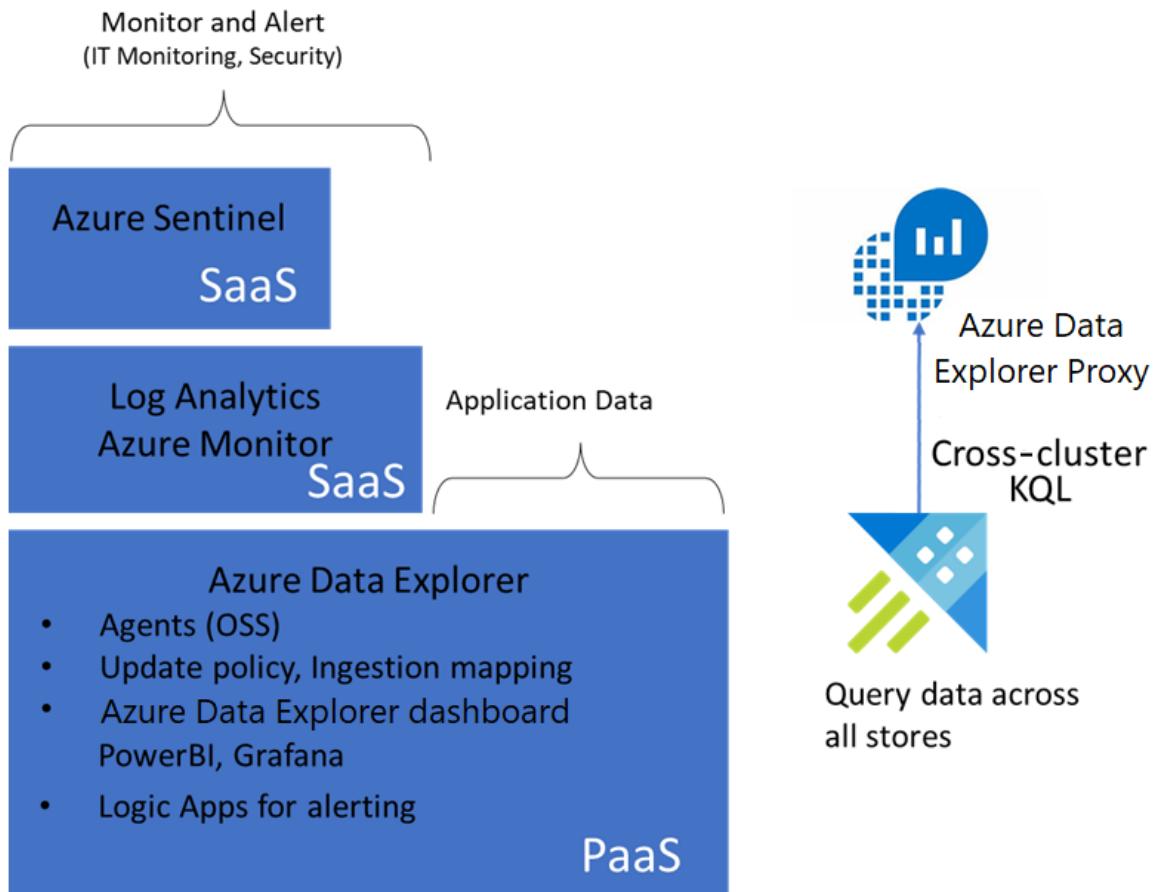
This solution applies to various scenarios. Specifically, security operations center (SOC) analysts can use this solution for:

- Full-scale investigations.
- Forensic analysis.
- Threat hunting.
- Security audits.

A customer testifies to the usefulness of the solution: "We deployed an Azure Data Explorer cluster almost a year and a half ago. In the last Solarigate data breach, we used an Azure Data Explorer cluster for forensic analysis. A Microsoft Dart team also used an Azure Data Explorer cluster to complete the investigation. Long-term security data retention is critical for full-scale data investigations."

Monitoring stack

The following diagram shows the Azure monitoring stack:



- Sentinel uses a Log Analytics workspace to store security logs and provide SIEM and SOAR solutions.
- Monitor tracks the status of IT assets and sends alerts when needed.
- Azure Data Explorer provides an underlying data platform that stores security logs for Log Analytics workspaces, Monitor, and Sentinel.

Main features

The solution's main features offer many benefits, as the following sections explain.

Long-term queryable data store

Azure Data Explorer indexes data during the storage process, making the data available for queries. When you need to focus on running audits and investigations, there's no need to process the data. Querying the data is straightforward.

Full-scale forensic analysis

Azure Data Explorer, Log Analytics, and Sentinel support cross-service queries. As a result, in a single query, you can reference data that's stored in any of these services.

SOC analysts can use the Kusto query language (KQL) to run full-range investigations. You can also use Azure Data Explorer queries in Sentinel for hunting purposes. For more information, see [What's New: Sentinel Hunting supports ADX cross-resource queries](#).

On-demand data caching

Azure Data Explorer supports [window-based hot caching](#). This functionality provides a way to move data from a selected period into the hot cache. Then you can run fast queries on the data, making investigations more efficient. You might need to add compute nodes to the hot cache for this purpose. After the investigation is complete, you can change the hot cache policy to move the data into the cold partition. You can also restore the cluster to its original size.

Continuous exporting to archive data

To follow regulatory requirements, some enterprises need to store security logs for an unlimited amount of time. Azure Data Explorer supports continuous exporting of data. You can use this capability to build an archival tier by storing security logs in Storage.

Proven query language

The Kusto query language is native to Azure Data Explorer. This language is also available in Log Analytics workspaces and Sentinel threat-hunting environments. This availability significantly reduces the learning curve for SOC analysts. Queries that you run on Sentinel also work on data that you store in Azure Data Explorer clusters.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Keep the following points in mind when you implement this solution.

Scalability

Consider these scalability issues:

Data export method

If you need to export a large amount of data from Log Analytics, you might reach Event Hubs capacity limits. To avoid this situation:

- Export data from Log Analytics into Blob Storage.
- Use Azure Data Factory workloads to periodically export the data into Azure Data Explorer.

By using this method, you can copy data from Data Factory only when the data nears its retention limit in Sentinel or Log Analytics. As a result, you avoid duplicating the data.

For more information, see [Export data from Log Analytics into Azure Data Explorer](#).

Query usage and audit preparedness

Generally, you keep data in the cold cache in your Azure Data Explorer cluster. This approach minimizes your cluster cost and is sufficient for most queries that involve data from previous months. But when you query large data ranges, you might need to scale out the cluster and load the data into the hot cache.

You can use the hot window feature of the hot cache policy for this purpose. You can also use this feature when you audit long-term data. When you use the hot window, you might need to scale your cluster up or out to make room for more data in the hot cache. After you've finished querying the large data range, change the hot cache policy to reduce your computing cost.

By turning on the optimized autoscale feature in your Azure Data Explorer cluster, you can optimize your cluster size based on the caching policy. For more information on querying cold data in Azure Data Explorer, see [Query cold data with hot windows](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

If you need to store security data for a long time or for an unlimited period, export the logs to Storage. Azure Data Explorer supports continuous exporting of data. By using this functionality, you can export data to Storage in compressed, partitioned Parquet format. You can then seamlessly query that data. For more information, see [Continuous data export overview](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

The Azure Data Explorer cluster cost is primarily based on the computing power that's used to store data in the hot cache. Queries on hot cache data offer better performance over cold cache queries. This solution stores most of the data in the cold cache, minimizing the computing cost.

To explore the cost of running this solution in your environment, use the [Azure pricing calculator](#).

Deploy this scenario

To automate deployment, use this [PowerShell script](#). This script creates these components:

- The target table
- The raw table
- The table mapping that defines how Event Hubs records land in the raw table
- Retention and update policies
- Event Hubs namespaces
- Data export rules in the Log Analytics workspace
- The data connection between Event Hubs and the Azure Data Explorer raw data table

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Deepak Agrawal](#) | Product Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Integrate Azure Data Explorer for long-term log retention](#)
- [Move Your Microsoft Sentinel Logs to Long-Term Storage with Ease](#)
- [Cross-resource query Azure Data Explorer by using Azure Monitor](#)

- HOW TO: Configure Microsoft Sentinel data export for long-term storage [↗](#)
- Using Azure Data Explorer for long-term retention of Microsoft Sentinel logs [↗](#)
- What's New: Microsoft Sentinel Hunting supports ADX cross-resource queries [↗](#)
- How to stream Microsoft Defender ATP hunting logs in Azure Data Explorer [↗](#)
- Blog Series: Limitless Advanced Hunting with Azure Data Explorer (ADX) [↗](#)

Related resources

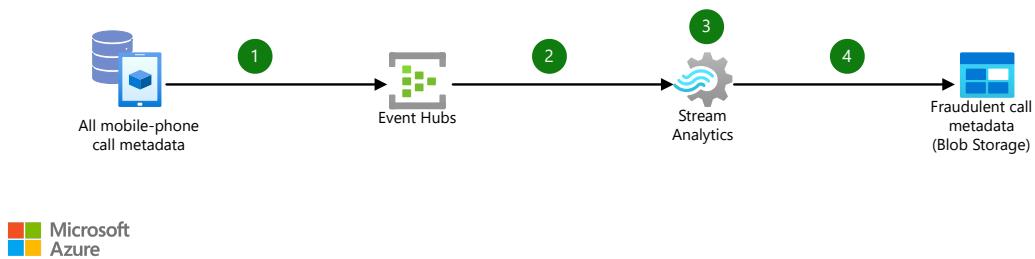
- [Azure Data Explorer monitoring](#)
- [Azure Data Explorer interactive analytics](#)
- [Big data analytics with Azure Data Explorer](#)

Real-time fraud detection

Azure Blob Storage Azure Event Hubs Azure Stream Analytics

This example scenario is relevant to organizations that need to analyze data in real time to detect fraudulent transactions or other anomalous activity. Also, see [Detect mobile bank fraud](#).

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

This scenario covers the back-end components of a real-time analytics pipeline. Data flows through the scenario as follows:

1. Mobile phone call metadata is sent from the source system to an Azure Event Hubs instance.
2. A Stream Analytics job is started. It receives data via the event hub source.
3. The Stream Analytics job runs a predefined query to transform the input stream and analyze it based on a fraudulent-transaction algorithm. This query uses a tumbling window to segment the stream into distinct temporal units.
4. The Stream Analytics job writes the transformed stream representing detected fraudulent calls to an output sink in Azure Blob storage.

Components

- [Azure Event Hubs](#) is a real-time streaming platform and event ingestion service, capable of receiving and processing millions of events per second. Event Hubs can process and store events, data, or telemetry that's produced by distributed

software and devices. In this scenario, Event Hubs receives all phone call metadata to be analyzed for fraudulent activity.

- [Azure Stream Analytics](#) is an event-processing engine that can analyze high volumes of data that streams from devices and other data sources. It also supports extracting information from data streams to identify patterns and relationships. These patterns can trigger other downstream actions. In this scenario, Stream Analytics transforms the input stream from Event Hubs to identify fraudulent calls.
- [Blob storage](#) is used in this scenario to store the results of the Stream Analytics job.

Alternatives

Many technology choices are available for real-time message ingestion, data storage, stream processing, storage of analytical data, and analytics and reporting.

Algorithms for fraud detection that are more complex can be produced by various machine learning services in Azure. For an overview of these options, see [Technology choices for machine learning](#).

For scenarios that are built by using Machine Learning Server, see [Fraud detection using Machine Learning Server](#). For other solution templates that use Machine Learning Server, see [Data science scenarios and solution templates](#).

Scenario details

Potential applications include identifying fraudulent credit card activity or fraudulent mobile phone calls. Traditional online analytical systems might take hours to transform and analyze the data to identify anomalous activity.

By using fully managed Azure services such as Event Hubs and Stream Analytics, companies can eliminate the need to manage individual servers, while reducing costs and using Microsoft's expertise in cloud-scale data ingestion and real-time analytics. This scenario specifically addresses the detection of fraudulent activity. If you have other needs for data analytics, you should review the list of available [Azure Analytics services](#).

This sample represents one part of a broader data processing architecture and strategy. Other options for this aspect of an overall architecture are discussed later in this article.

Potential use cases

Other relevant use cases include:

- Detecting fraudulent mobile-phone calls in telecommunications scenarios.
- Identifying fraudulent credit card transactions for banking institutions.
- Identifying fraudulent purchases in retail or e-commerce scenarios.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Availability

Azure Monitor provides unified user interfaces for monitoring across various Azure services. For more information, see [Monitoring in Microsoft Azure](#). Event Hubs and Stream Analytics are both integrated with Azure Monitor.

Scalability

The components of this scenario are designed for hyperscale ingestion and massively parallel real-time analytics. Azure Event Hubs is highly scalable, capable of receiving and processing millions of events per second with low latency. Event Hubs can [automatically scale up](#) the number of throughput units to meet usage needs. Azure Stream Analytics is capable of analyzing high volumes of streaming data from many sources. You can scale up Stream Analytics by increasing the number of [streaming units](#) allocated to execute your streaming job.

For general guidance on designing scalable solutions, see the [performance efficiency checklist](#) in the Azure Architecture Center.

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

Azure Event Hubs secures data through an [authentication and security model](#) that's based on a combination of Shared Access Signature (SAS) tokens and event publishers. An event publisher defines a virtual endpoint for an event hub. The publisher can only be used to send messages to an event hub. It's not possible to receive messages from a publisher.

For general guidance on designing secure solutions, see the [Azure Security Documentation](#).

Resiliency

For general guidance on designing resilient solutions, see [Designing reliable Azure applications](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To explore the cost of running this scenario, all the services are pre-configured in the cost calculator. To see how the pricing changes for your use case, change the appropriate variables to match your expected data volume.

We have provided three sample cost profiles that are based on the amount of traffic you expect to get:

- [Small ↗](#): process one million events through one standard streaming unit per month.
- [Medium ↗](#): process 100M events through five standard streaming units per month.
- [Large ↗](#): process 999 million events through 20 standard streaming units per month.

Deploy this scenario

To deploy this scenario, you can follow this [step-by-step tutorial](#) that demonstrates how to manually deploy each component of the scenario. This tutorial also provides a .NET client application to generate sample phone call metadata and send that data to an event hub instance.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Alex Buck ↗](#) | Senior Content Developer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure Event Hubs](#) — A big data streaming platform and event ingestion service
- [Welcome to Azure Stream Analytics](#)
- [Introduction to Azure Blob storage](#)

Related resources

- [Detect mobile bank fraud](#)
- [Integrate Event Hubs with serverless functions on Azure](#)
- [Serverless event processing](#)
- [Stream processing with Azure Stream Analytics](#)

Restrict interservice communications

Azure Microsoft Entra ID Azure App Service

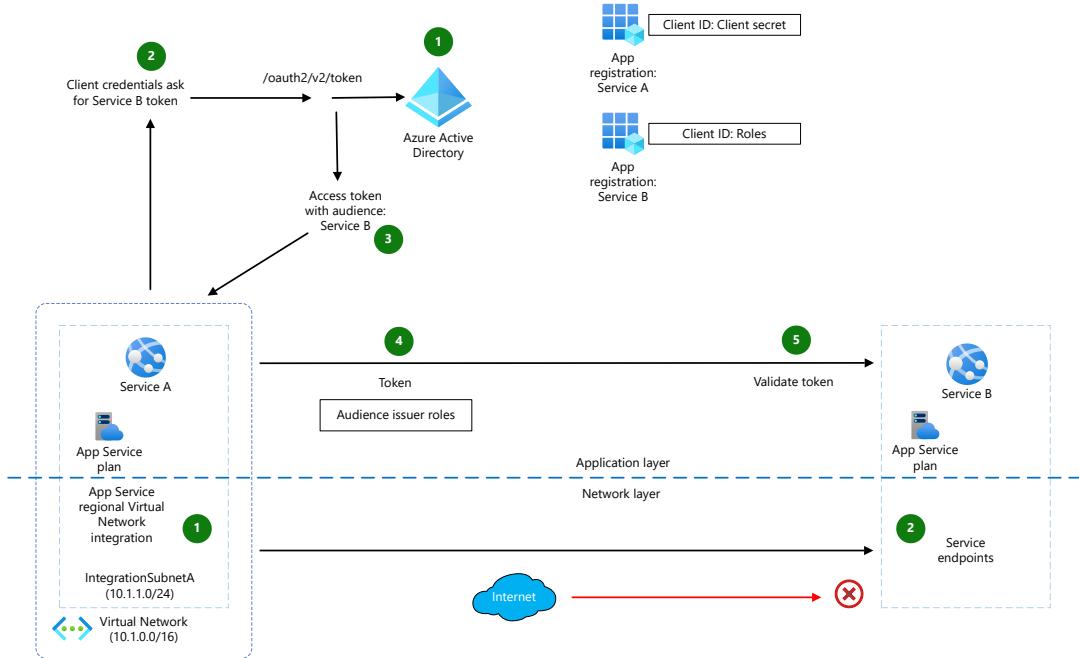
This example scenario restricts communications between two Azure backend services on both the application and network layers. Communications can flow only between services that explicitly allow it, adhering to the [principle of least privilege](#). This example uses Azure App Service to host the services, but you can use similar techniques for Azure Functions Apps.

Interservice communications restrictions are only one part of an overall security strategy based on careful planning, [threat-modeling](#), and the [Security Development Lifecycle](#). Overall security planning should incorporate business, compliance, regulatory, and other non-functional requirements.

Potential use cases

While the current scenario focuses on network restrictions, many organizations now embrace a [zero trust security model](#) that assumes a breach, so the networking layer is of secondary importance.

Architecture



[Download a Visio file](#) of this architecture.

Dataflow

The diagram shows restricted communications from Service A to Service B. Token-based authorization restricts access on the application layer, and service endpoints restrict access on the network layer.

- Both services [register with Microsoft Entra ID](#), and use OAuth 2.0 token-based authorization in the [client credentials flow](#).
- Service A communicates by using [Regional VNet Integration](#) from a private IP address in its virtual network integration subnet. Service B [service endpoints](#) accept inbound communications only from the Service A integration subnet.

Token-based authorization

An OpenID Connect (OIDC)-compatible library like the [Microsoft Authentication Library \(MSAL\)](#) supports this token-based client credentials flow. For more information, see [Scenario: Daemon application that calls web APIs](#) and the [sample application for the daemon scenario](#).

1. Both Service A and Service B register in Microsoft Entra ID. Service A has client credentials in either shared secret or certificate form.

2. Service A can use its own client credentials to request an access token for Service B.
3. Microsoft Entra ID provides an access token with a Service B audience or [aud](#) claim.
4. Service A injects the token as a *bearer token* in the HTTP Authorization header of a request to Service B, according to the [OAuth 2.0 Bearer Token Usage specification](#).
5. Service B [validates the token](#) to ensure that the [aud](#) claim matches the Service B application.

Service B uses one of the following methods to ensure that only specifically allowed clients, Service A in this case, can get access:

- **Validate the token appid claim.** Service B can validate the token [appid](#) claim, which identifies which Microsoft Entra registered application requested the token. Service B explicitly checks the claim against a known access control caller list.
- **Check for roles in the token.** Similarly, Service B can check for certain [roles](#) claimed in the incoming token, to ensure that Service A has explicit access permissions.
- **Require user assignment.** Alternatively, the Service B owner or admin can configure Microsoft Entra ID to require *user assignment*, so only applications that have explicit permissions to the Service B application can get a token toward Service B. Service B then doesn't need to check for specific roles, unless business logic requires it.

To set up a user assignment requirement to access Service B:

1. In Microsoft Entra ID, [enable user assignment](#) on Service B.
2. [Expose at least one app role](#) on Service B that Service A can ask permission for. The [AllowedMemberTypes](#) for this role must include [Application](#).
3. [Request app permission](#) for Service A to the exposed Service B role.
 - a. From the [API permissions](#) section of the Service A app registration, select [Add a permission](#), and then select the Service B application from the list.
 - b. On the [Request API permissions](#) screen, select [Application permissions](#), because this backend application runs without a signed-in user. Select the exposed Service B role, and then select [Add permissions](#).
4. [Grant admin consent](#) to the Service A application permissions request. Only a Service B owner or admin can consent to the Service A permissions request.

Service endpoints

The lower half of the architectural diagram shows how to restrict interservice communications on the network layer:

1. The Service A web app uses [Regional VNet Integration](#) to route all outbound communications through a private IP address within the IP range of the integration subnet.
2. Service B has [service endpoints](#) that allow inbound communications only from web apps on the integration subnet of Service B.

For more information, see [Set up Azure App Service access restrictions](#).

Components

This scenario uses the following Azure services:

- [Azure App Service](#) hosts both Service A and Service B, allowing autoscale and high availability without having to manage infrastructure.
- [Microsoft Entra ID](#) is the cloud-based identity and access management service that authenticates services and enables OAuth 2.0 token-based authorization.
- [Azure Virtual Network](#) is the fundamental building block for private networks in Azure. Azure Virtual Network lets resources like Azure Virtual Machines (VMs) securely communicate with each other, the internet, and on-premises networks.
- [Azure Service Endpoints](#) provide secure and direct connectivity to Azure services over an optimized route on the Azure backbone network, and allow access only from the range of private source IPs in the integration subnet.
- [Microsoft Authentication Library \(MSAL\)](#) is an OIDC-compatible library that allows a service to fetch access tokens from Microsoft Entra ID using a client credentials flow.

Alternatives

There are several alternatives to the example scenario.

Managed identity

Instead of registering as an application with Microsoft Entra ID, Service A could use a [managed identity](#) to fetch an access token. Managed identity frees operators from having to manage credentials for an app registration.

While a managed identity lets Service A fetch a token, it doesn't provide a Microsoft Entra app registration. For other services to request an access token for Service A itself, Service A still needs a Microsoft Entra app registration.

You can't assign a managed identity to an app role through the Azure portal, only through the Azure PowerShell command line. For more information, see [Assign a](#)

managed identity access to an application role using PowerShell.

Azure Functions

You can host the services in [Azure Functions](#) instead of App Service. To restrict access on the network layer by using Regional VNet Integration, you need to host the Functions apps in an App Service plan or a Premium Plan. For more information, see [Azure Functions networking options](#).

App Service built-in authentication and authorization

By design, this scenario collocates the authorization code with the rest of the business logic by performing token validation as part of application code. [App Service built-in authentication and authorization](#), or Easy Auth, can also perform basic token validation before sending a request to a service. The service then relies on the hosting infrastructure to reject unauthorized requests.

To configure App Service authentication and authorization, set the authorization behavior to **Log in with Microsoft Entra ID**. This setting validates tokens and restricts access to valid tokens only.

The downside of using Easy Auth is that the service loses the authentication and authorization protection if it moves elsewhere. While App Service authentication and authorization works for simple scenarios, complex authorization requirements should use logic from within the application code.

Service endpoints vs. private endpoints

This scenario uses service endpoints rather than [private endpoints](#), because only service endpoints allow restricting access to a web app from a given subnet. Filtering inbound traffic on private endpoints isn't supported through Network Security Groups (NSGs) or by using App Service access restrictions. Every service with network line-of-sight can communicate with the private endpoint of a web application. This limits private endpoint usefulness for locking down traffic on the network layer.

Considerations

- App Service Regional VNet Integration provides a single integration subnet for each App Service Plan. All web apps on the same plan integrate with the same subnet, and share the same set of private outbound IP addresses. Receiving services can't distinguish which web app the traffic originates from. If you need to

identify the originating web app, you must deploy the web apps on separate App Service Plans, each with its own integration subnet.

- Every worker instance in an App Service Plan occupies a separate private IP address within the integration subnet. To plan for scale, ensure that the integration subnet is large enough to accommodate the scale you expect.

Cost optimization

Pricing for this scenario depends on your specific infrastructure and requirements. Microsoft Entra ID has Free up to Premium tiers, depending on needs. Costs for Azure App Service or other hosts vary with your specific scale and security requirements, as described in [Alternatives](#) and [Considerations](#).

To calculate costs for your scenario, see the [Azure pricing calculator](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Christof Claessens](#) | FastTrack for Azure Engineer

Next steps

- [Message encoding considerations for cloud applications](#)
- [Enterprise deployment using App Services Environment](#)
- [Web app private connectivity to Azure SQL database](#)

Related resources

- [App Service networking features](#)
- [Zero to Hero: securing your web app](#)
- [Zero to Hero: multi-tier web apps](#)
- [Microsoft Entra client credentials flow](#)
- [Service endpoints](#)
- [App Service Regional VNet Integration](#)
- [Sample application demonstrating client credentials flow for daemon apps](#)
- [Azure Security Baseline for App Service](#)

Secure OAuth 2.0 On-Behalf-Of refresh tokens for web services

Azure CLIs Azure DevOps Azure Functions Azure Key Vault Azure Pipelines

When developing web services, you may need to get tokens using the [OAuth 2.0 On-Behalf-Of \(OBO\) flow](#). The OBO flow serves the use case where an application invokes a service or web API, which in turn needs to call another service or web API. OBO propagates the delegated user identity and permissions through the request chain. When an application needs to use access and refresh tokens indefinitely, typically in offline access scenarios, it's critical to store the refresh tokens securely.

⚠ Warning

Carefully consider the risk and responsibility involved in storing any security tokens, since these tokens can give a malicious actor access to resources protected by the organization's Microsoft Entra ID. A security breach of an application that targets **Accounts in any organizational directory (Any Microsoft Entra directory - Multitenant)** can be especially disastrous.

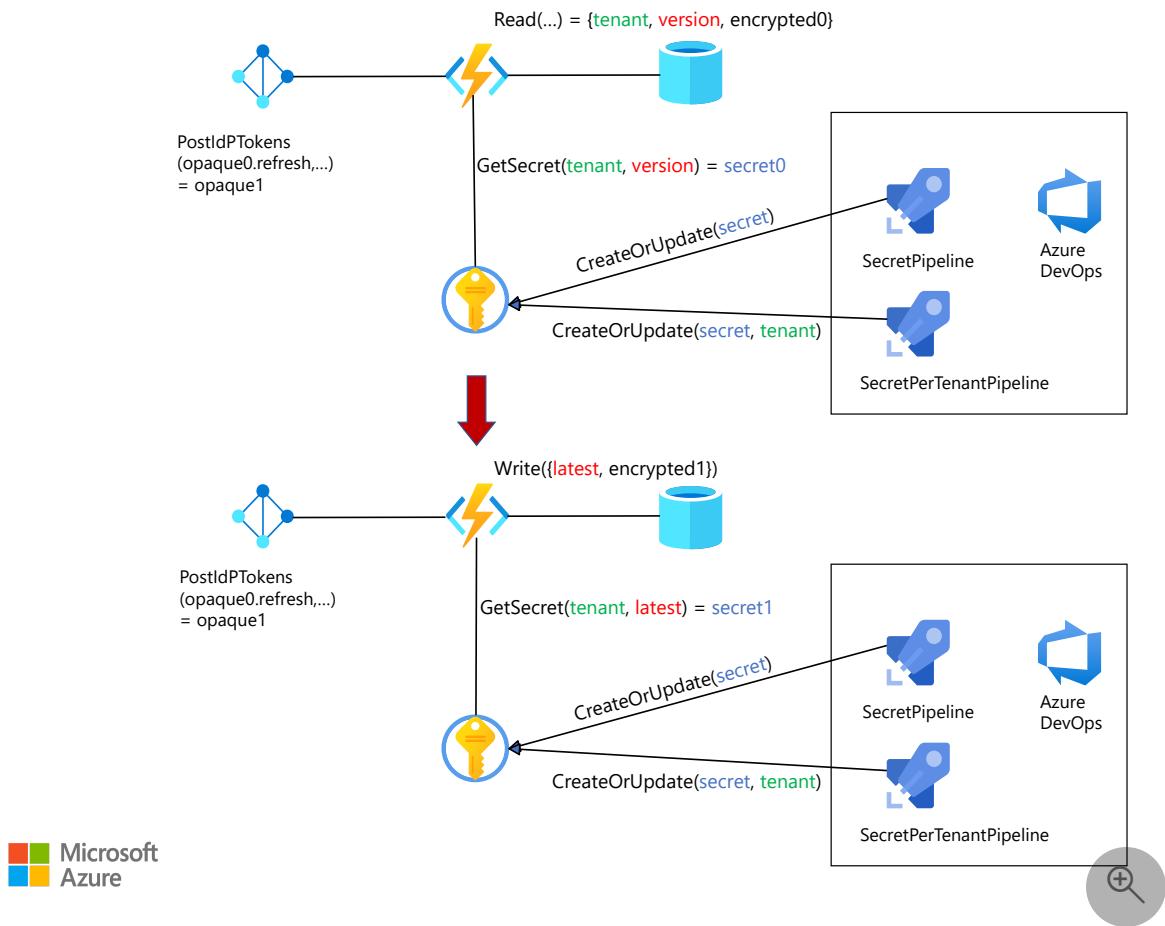
Storing access tokens poses a greater security risk, since an access token in and of itself can access resources. The recommended approach is not to store access tokens, but get the access tokens as needed. Securely store only the refresh tokens, with as much rigor as if they were access tokens.

If necessary, you can [revoke refresh tokens](#) if they become compromised.

Potential use cases

This solution uses Azure Key Vault, Azure Functions, and Azure DevOps to securely update and store OBO refresh tokens.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

- Azure Key Vault [holds secret encryption keys for each Microsoft Entra ID](#) tenant.
- An Azure Functions [timer-triggered function gets the latest secret key from Key Vault. Another Azure Functions function retrieves the refresh token from the Microsoft identity platform and saves it with the latest secret key version.](#)
- A database stores the latest encrypted key and opaque data.
- An Azure DevOps [continuous delivery pipeline manages and syncs the secret rotation and token refresh processes.](#)

Azure Pipelines [is a convenient place to add your key rotation strategy, if you're already using Pipelines for infrastructure-as-code \(IaC\) or continuous integration and delivery \(CI/CD\). You don't have to use Azure Pipelines, as long as you limit the paths for setting and retrieving secrets.](#)

Apply the following policy to allow the Service Principal for your Azure DevOps service connection to set secrets in Key Vault. Replace the `<Key Vault Name>` and `<Service Connection Principal>` variables with the correct values for your environment.

Azure CLI

```
az keyvault set-policy --name $<Key Vault Name> --spn $<Service Connection Principal> --secret-permissions set
```

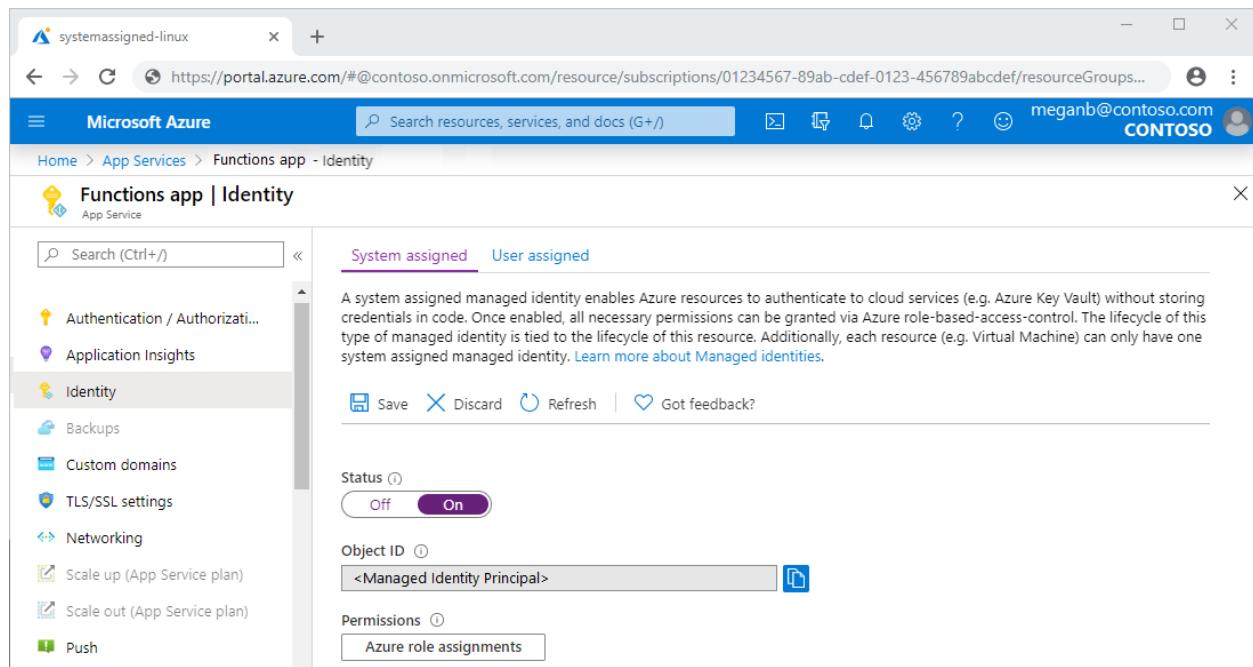
After you set up Azure Pipelines to create and update keys, you can schedule the pipeline to run periodically. The pipeline updates the Key Vault secret to sync with key rotation, and saves the encrypted token with the new secret version. For more information, see [Configure schedules for pipelines](#).

Managed identity

The preferred way for an Azure service like Azure Functions to access Key Vault is to use the service's [managed identity](#). You can grant access through the Azure portal, Azure CLI, or through an Azure Resource Manager (ARM) template for IaC scenarios.

Azure portal

In the Azure portal, add a Key Vault access policy to allow the Azure Functions managed identity Object ID to **Get** and **Set** secrets. For more information, see [Add a system-assigned identity](#) and [Use Key Vault references for App Service and Azure Functions](#).



Azure CLI

You can also set Azure Key Vault policy by using the [Azure CLI](#):

Azure CLI

```
az keyvault set-policy --name $<Key Vault Name> --spn $<Service Connection Principal> --secret-permissions set
az keyvault set-policy --name $<Key Vault Name> --spn $<Managed Identity Principal> --secret-permissions get
```

ARM template

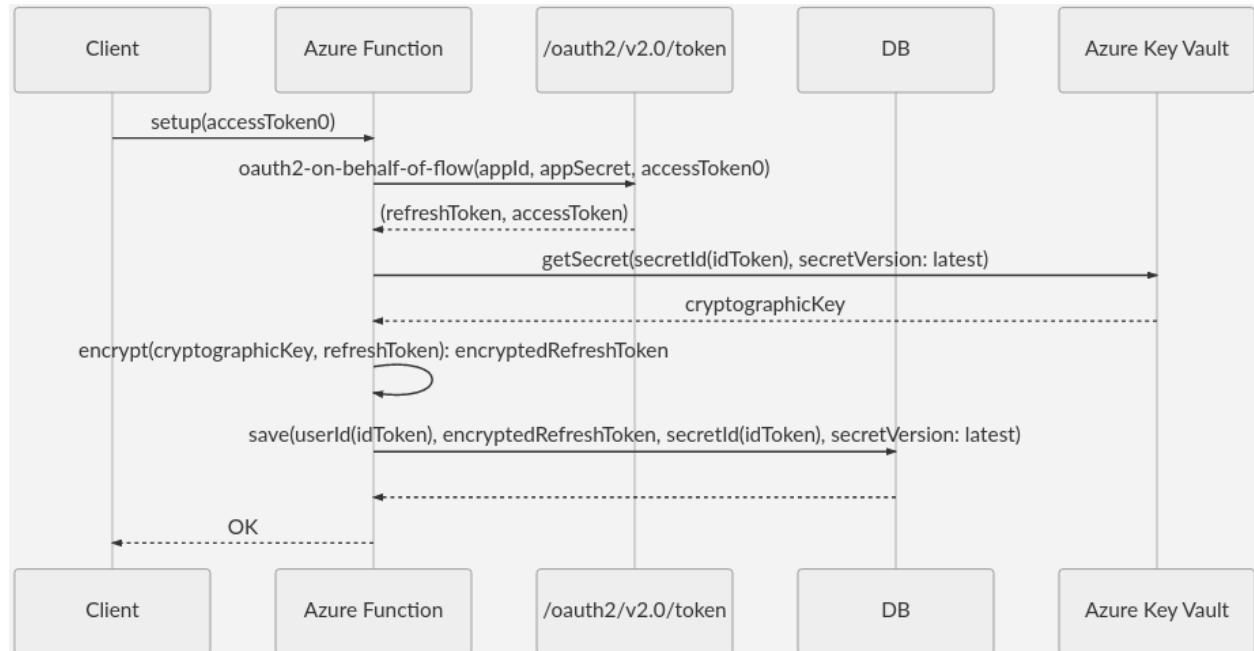
The following [ARM template](#) gives Azure Functions access to Azure Key Vault. Replace the `***` variables with the correct values for your environment.

JSON

```
{
  "type": "Microsoft.KeyVault/vaults",
  "apiVersion": "2019-09-01",
  "name": "***",
  "location": "***",
  "properties": {
    "sku": {
      "family": "A",
      "name": "standard"
    },
    "tenantId": "***",
    "enableSoftDelete": true,
    "enabledForDeployment": false,
    "enabledForTemplateDeployment": false,
    "enabledForDiskEncryption": false,
    "accessPolicies": [
      {
        "tenantId": "***",
        "objectId": "<Managed Identity Principal>",
        "permissions": {
          "secrets": [
            "get"
          ]
        }
      },
      {
        "tenantId": "***",
        "objectId": "<Service Connection Principal>",
        "permissions": {
          "secrets": [
            "set"
          ]
        }
      }
    ]
  }
}
```

Token storage

You can use any database to store the tokens in encrypted form. The following diagram shows the sequence to store refresh tokens in a database:



The sequence has two functions, `userId()` and `secretId()`. You can define these functions as some combination of `token.oid`, `token.tid`, and `token.sub`. For more information, see [Using the id_token](#).

With the cryptographic key stored as a secret, you can look up the latest version of the key in Azure Key Vault.

Token usage

Using the key is straightforward. The following sequence queries the key based on the latest key version.



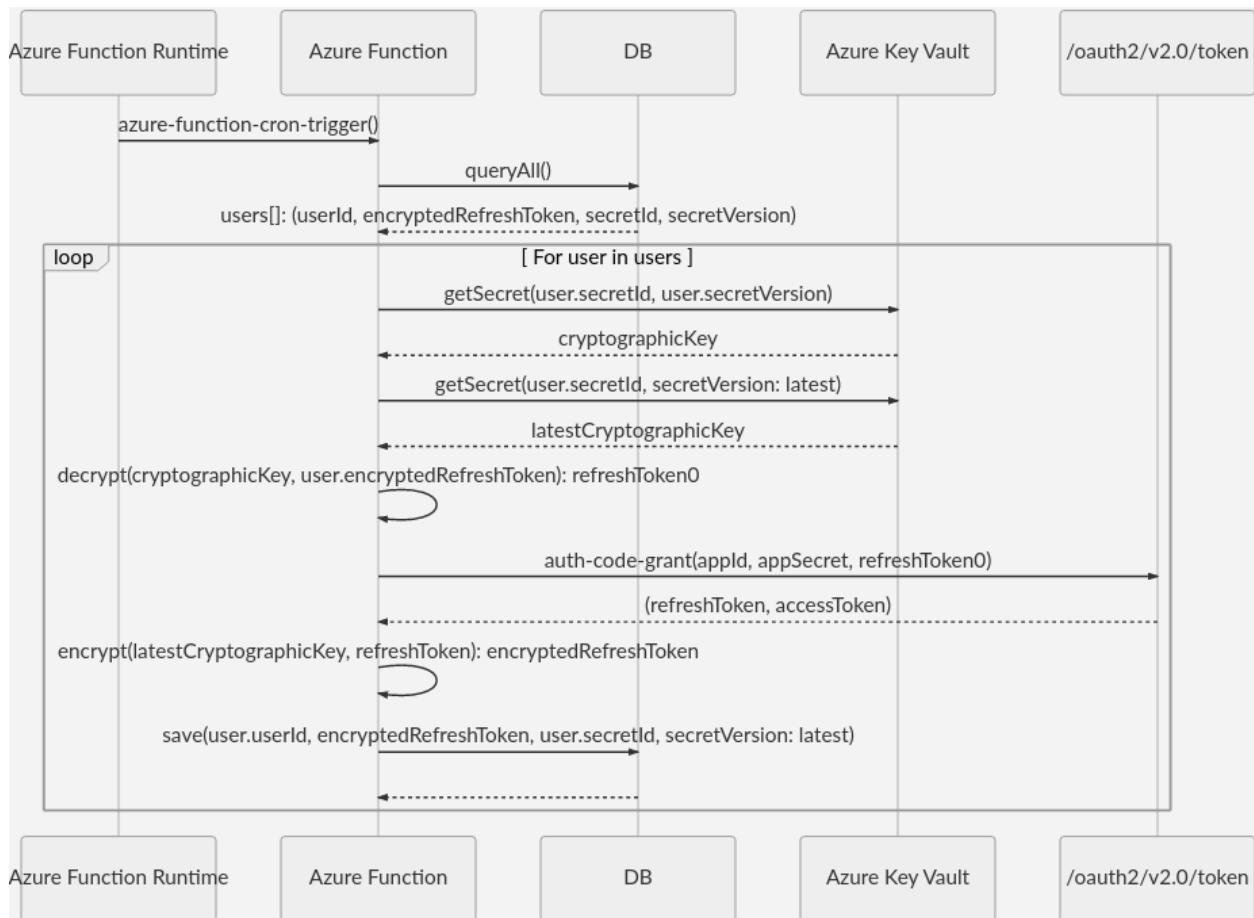
The token refresh is orthogonal to the `DoWork` function, so Azure Functions can perform `DoWork` and token refresh asynchronously by using [Durable Functions](#). For more information about HTTP-triggered functions with Durable Functions, see [HTTP features](#).

It's not recommended to use Azure Key Vault in the HTTP request pipeline, so cache responses whenever reasonable. In the example, Key Vault's response to the `getSecret(secretId, secretVersion)` call is cacheable.

Key rotation and token refresh

You can rotate the secret key at the same time that you refresh the refresh token, so the latest token gets encrypted with the latest version of the encryption secret. This process uses the built-in Azure Functions support for timer triggers. For more information, see [Timer trigger for Azure Functions](#).

The following sequence diagram illustrates the process of syncing the token refresh with the key rotation:



User and access control

Microsoft identity platform offers the ability to revoke refresh tokens in case of compromise. See [Token revocation](#) and [Revoke-AzureADUserAllRefreshToken](#).

To remove a user from Microsoft Entra ID, just remove the user's record. To remove application access per user, remove the `refreshToken` part of the user data.

To remove access for a group of users, such as all users in a target tenant, you can use Azure Pipelines to delete the group's secret based on `secretId()`.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Jason Mostella](#) | Senior Software Engineer

Next steps

- Microsoft identity platform and OAuth 2.0 On-Behalf-Of flow
- How to use managed identities for App Service and Azure Functions
- Use Key Vault references for App Service and Azure Functions
- Securing Azure Functions

Securely managed web applications

Azure App Service

Azure Application Gateway

Azure SQL Database

Azure VPN Gateway

Azure Web Application Firewall

This article provides an overview of deploying secure applications using the [Azure App Service Environment](#). To restrict application access from the internet, the [Azure Application Gateway](#) service and [Azure Web Application Firewall](#) are used. This article also provides guidance about continuous integration and continuous deployment (CI/CD) for App Service Environments using Azure DevOps.

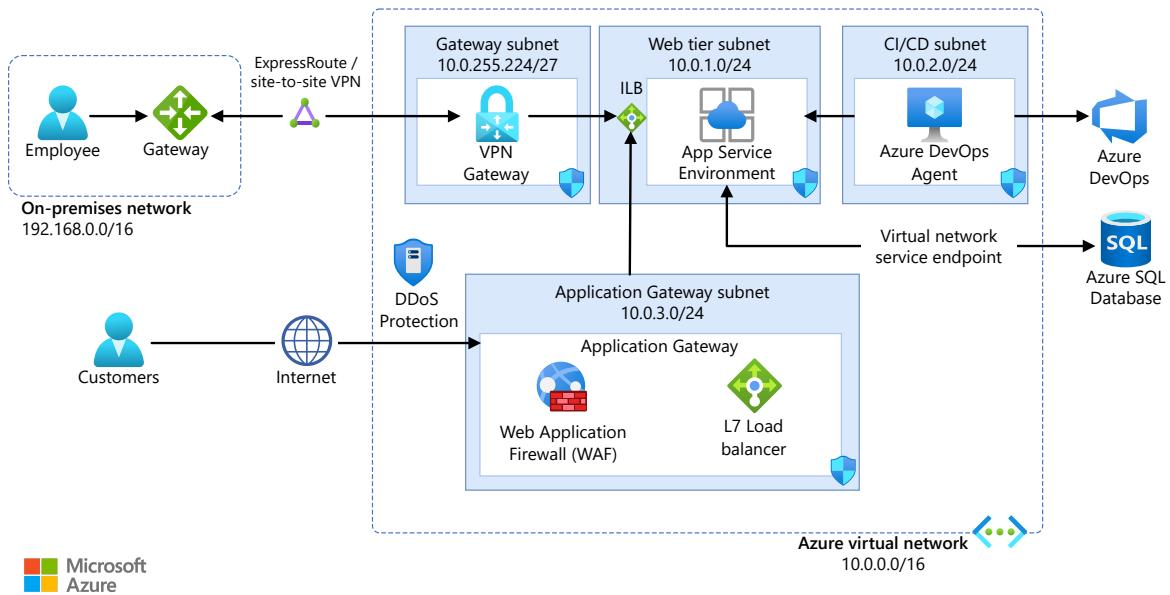
This scenario is commonly deployed in industries such as banking and insurance, where customers are conscious of platform-level security in addition to application level security. To demonstrate these concepts, we'll use an application that allows users to submit expense reports.

Potential use cases

Consider this scenario for the following use cases:

- Building an Azure Web App where extra security is required.
- Providing dedicated tenancy, rather than shared tenant App Service Plans.
- Using Azure DevOps with an [internally load-balanced\(ILB\)](#) Application Service Environment.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. HTTP/HTTPS requests first hit the Application Gateway.
2. Optionally (not shown in the diagram), you can have Microsoft Entra authentication enabled for the Web App. After the traffic first hits the Application Gateway, the user would be prompted to supply credentials to authenticate with the application.
3. User requests flow through the internal load balancer (ILB) of the environment, which in turn routes the traffic to the Expenses Web App.
4. The user then proceeds to create an expense report.
5. As part of creating the expense report, the deployed API App is invoked to retrieve the user's manager name and email.
6. The created expense report is stored in Azure SQL Database.
7. To facilitate continuous deployment, code is checked into the Azure DevOps instance.
8. The build VM has the Azure DevOps Agent installed, allowing the build VM to pull the bits for the Web App to deploy to the App Service Environment (since the Build VM is deployed in a subnet inside the same virtual network).

Components

- The [App Service Environment](#) provides a fully isolated, dedicated environment for securely running the application at high scale. In addition, because the App Service Environment and the workloads that run on it are behind a virtual network, it also provides an extra layer of security and isolation. The requirement of high scale and isolation drove the selection of ILB App Service Environment.
- This workload uses the [isolated App Service pricing tier](#), so the application runs in a private dedicated environment in an Azure datacenter using faster processors, SSD storage, and double the memory-to-core ratio compared to Standard.
- Azure App Services [Web App](#) and [API App](#) host web applications and RESTful APIs. These apps and APIs are hosted on the Isolated service plan, which also offers autoscaling, custom domains, and so on, but in a dedicated tier.
- Azure [Application Gateway](#) is a web traffic load balancer operating at Layer 7 that manages traffic to the web application. It offers SSL offloading, which removes extra overhead from the web servers hosting the web app to decrypt traffic again.
- [Web Application Firewall](#) (WAF) is a feature of Application Gateway. Enabling the WAF in the Application Gateway further enhances security. The WAF uses OWASP rules to protect the web application against attacks such as cross-site scripting, session hijacks, and SQL injection.
- [Azure SQL Database](#) was selected because most of the data in this application is relational data, with some data as documents and Blobs.
- [Azure Networking](#) provides various networking capabilities in Azure, and the networks can be peered with other virtual networks in Azure. Connections can also be established with on-premises datacenters via ExpressRoute or site-to-site. In this case, a [service endpoint](#) is enabled on the virtual network to ensure the data is flowing only between the Azure virtual network and the SQL Database instance.
- [Azure DevOps](#) is used to help teams collaborate during sprints, using features that support Agile Development, and to create build and release pipelines.
- An Azure build [VM](#) was created so that the installed agent can pull down the respective build, and deploy the web app to the environment.

Alternatives

An App Service Environment can run regular web apps on Windows or, as in this example, web apps deployed inside the environment that are each running as Linux containers. An App Service Environment was selected to host these single-instance containerized applications. There are alternatives available—review the considerations below when designing your solution.

- [Azure Service Fabric](#): If your environment is mostly Windows-based, and your workloads are primarily .NET Framework-based, and you aren't considering rearchitecting to .NET Core, then use Service Fabric to support and deploy

Windows Server Containers. Additionally, Service Fabric supports C# or Java programming APIs, and for developing native microservices, the clusters can be provisioned on Windows or Linux.

- [Azure Kubernetes Service](#) (AKS) is an open-source project and an orchestration platform more suited to hosting complex multicontainer applications that typically use a microservices-based architecture. AKS is a managed Azure service that abstracts away the complexities of provisioning and configuring a Kubernetes cluster. However, significant knowledge of the Kubernetes platform is required to support and maintain it, so hosting a handful of single-instance containerized web applications might not be the best option.

Other options for the data tier include:

- [Azure Cosmos DB](#): If most of your data is in non-relational format, Azure Cosmos DB is a good alternative. This service provides a platform to run other data models such as MongoDB, Cassandra, Graph data, or simple table storage.

Considerations

There are certain considerations when dealing with certificates on ILB App Service Environment. You need to generate a certificate that is chained up to a trusted root without requiring a Certificate Signing Request generated by the server where the cert will eventually be stored. With Internet Information Services (IIS), for example, the first step is to generate a CSR from your IIS server and then send it to the SSL certificate-issuing authority.

You can't issue a CSR from the Internal Load Balancer (ILB) of an App Service Environment. The way to handle this limitation is to use [the wildcard procedure](#).

The wildcard procedure allows you to use proof of DNS name ownership instead of a CSR. If you own a DNS namespace, you can put in special DNS TXT record, the wildcard procedure checks that the record is there, and if found, knows that you own the DNS server because you have the right record. Based on that information, it issues a certificate that is signed up to a trusted root, which you can then upload to your ILB. You don't need to do anything with the individual certificate stores on the Web Apps because you have a trusted root SSL certificate at the ILB.

Make self-signed or internally issued SSL cert work if you want to make secure calls between services running in an ILB App Service Environment. Another [solution to consider](#) on how to make ILB App Service Environment work with internally issued SSL certificate and how to load the internal CA to the trusted root store.

While provisioning the App Service Environment, consider the following limitations when choosing a domain name for the environment. Domain names can't be:

- net
- azurewebsites.net
- p.azurewebsites.net
- nameofthease.p.azurewebsites.net

Additionally, the custom domain name used for apps and the domain name used by the ILB App Service Environment can't overlap. For an ILB App Service Environment with the domain name contoso.com, you can't use custom domain names for your apps like:

- www.contoso.com
- abcd.def.contoso.com
- abcd.contoso.com

Choose a domain for the ILB App Service Environment that won't conflict with those custom domain names. You can use something like contoso-internal.com for the domain of your environment for this example, because that won't conflict with custom domain names that end in .contoso.com.

Another point to consider is DNS. In order to allow applications within the App Service Environment to communicate with each other, for instance a web application to talk to an API, you'll need to have DNS configured for your virtual network holding the environment. You can either [bring your own DNS](#) or you can use [Azure DNS private zones](#).

Availability

- Consider applying the [typical design patterns for availability](#) when building your cloud application.
- Review the availability considerations in the appropriate [App Service web application reference architecture](#).
- For other considerations concerning availability, see the [availability checklist](#) in the Azure Architecture Center.

Scalability

- Understand how [scale works](#) in App Service Environments.
- Review best practices for [cloud apps autoscale](#).
- When building a cloud application, be aware of the [typical design patterns for scalability](#).

- Review the scalability considerations in the appropriate [App Service web application reference architecture](#).
- For other scalability articles, see the [performance efficiency checklist](#) available in the Azure Architecture Center.

Security

- Review the overview of the [security pillar](#).
- Review the security considerations in the appropriate [App Service web application reference architecture](#).
- Consider following a [secure development lifecycle](#) process to help developers build more secure software and address security compliance requirements while reducing development cost.
- Review the blueprint architecture for [Azure PCI DSS compliance](#).
- [Azure DDoS Protection](#), combined with application-design best practices, provides enhanced DDoS mitigation features to provide more defense against DDoS attacks. You should enable [Azure DDOS Protection](#) on any perimeter virtual network.

Resiliency

- Consider using [Geo Distributed Scale with App Service Environments](#) for greater resiliency and scalability.
- Review the [typical design patterns for resiliency](#) and consider implementing these where appropriate.
- You can find several [recommended practices for App Service](#) in the Azure Architecture Center.
- Consider using active geo-replication for the data tier and [geo-redundant storage](#) for images and queues.
- For a deeper discussion on [resiliency](#), see the relevant article in the Azure Architecture Center.

Deploy this scenario

To deploy this scenario, follow this [step-by-step tutorial](#) demonstrating how to manually deploy each component. Select App Service Environment v3 instead of v2, when following the tutorial. This tutorial also provides a .NET sample application that runs a simple Contoso expense reporting application.

Pricing

Explore the cost of running this scenario. All of the services are pre-configured in the cost calculator. To see how pricing would change for your particular use case, change the appropriate variables to match your expected traffic.

We've provided three sample cost profiles based on amount of traffic you expect to get:

- [Small ↗](#) : This pricing example represents the components necessary for a minimum production-level instance serving a few thousand users per month. The app is using a single instance of a standard web app that will be enough to enable autoscaling. Each of the other components is scaled to a basic tier that will minimize cost but still ensure that there's SLA support and enough capacity to handle a production-level workload.
- [Medium ↗](#) : This pricing example represents the components needed for a moderate size deployment. Here we estimate approximately 100,000 users over the course of a month. The expected traffic is handled in a single app service instance with a moderate standard tier. Additionally, moderate tiers of cognitive and search services are added to the calculator.
- [Large ↗](#) : This pricing example represents an application meant for high scale, at the order of millions of users per month, moving terabytes of data. At this level of usage, high performance, premium tier web apps deployed in multiple regions fronted by traffic manager are required. Data consists of the following components: storage, databases, and CDN, all configured for terabytes of data.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- Faisal Mustafa | Senior Customer Engineer

Next steps

- [Step-by-step deployment tutorial ↗](#)
- [Integrate your ILB App Service Environment with the Azure Application Gateway](#)
- [Integrate your Web Apps with the Azure Application Gateway ↗](#)
- [Geo distributed scale with App Service Environments](#)

Related resources

- [App Service web application reference architecture](#)
- [High availability enterprise deployment using App Services Environment](#)
- [Publish internal APIs to external users](#)

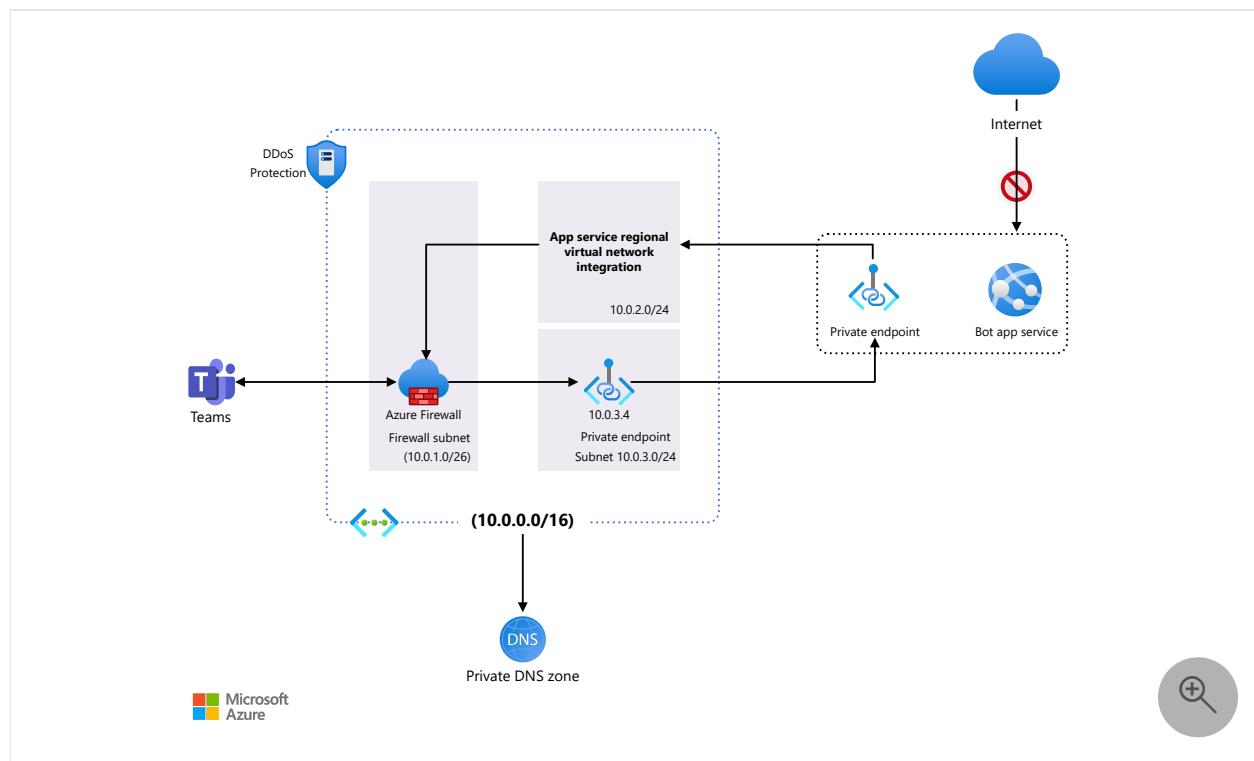
Help secure your Microsoft Teams channel bot and web app behind a firewall

Azure App Service

Azure Web Application Firewall

This example scenario helps secure the connection to a Microsoft Teams channel bot's web app by using Azure Private Link and Azure Private Endpoint. At the same time, it enables channels in the Teams client to communicate with the bot through an IP that's exposed through an Azure Firewall instance.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

- **Azure Virtual Network** enables communications between Azure resources. The virtual network in this example uses the address space of 10.0.0.0/16, and contains three subnets for use by the scenario's required components:
 - **Azure Firewall Subnet** (10.0.1.0/26).

- *Virtual Network Integration Subnet* (10.0.2.0/24), which is used to route traffic from the bot's private endpoint to the firewall.
- *Private Endpoint Subnet* (10.0.3.0/24), which is used to route traffic from the firewall to the bot's private endpoint.
- **Azure Firewall** exposes a single public IP address that clients can use to communicate with the underlying bot services. Ordinarily, a firewall is placed in its own virtual network, which is a common pattern for **hub and spoke** architectures, but this simplified example deploys all services and resources into a single virtual network. The Azure Firewall instance is placed in its own subnet.
- **Route table** defines the routes that traffic takes within the virtual network. It ensures that traffic coming to and from the bot passes through the firewall.
 - The default route with the 0.0.0.0/0 address prefix instructs Azure to route traffic that isn't within the address prefix of any other route to the subnet where the Azure Firewall instance is deployed. In this example, it's the only route.
 - The *Virtual Network Integration Subnet* and the *Private Endpoint Subnet* are associated with the route table, ensuring that any traffic passing through them is routed through the firewall.
- **Bot Service** consists of the bot **app service plan**, **app service**, and **bot channels registration**.
 - The app service has a registered custom domain that points to the IP address of the firewall. This way, the app service can be accessed only through the firewall.
- **Azure Private Link** service for inbound access to the bot app service over an **Azure private endpoint**.
- **Virtual network integration** connects the app service to the virtual network, ensuring that outbound traffic from the bot app service passes through the firewall.

Components

- [Virtual Network ↗](#)
- [Azure Firewall ↗](#)
- [Azure Bot Services ↗](#)
- [Azure App Service ↗](#)
- [Azure Private Link ↗](#)

Alternatives

- An [App Service Environment](#) can provide a fully isolated and dedicated environment for securely running App Service apps at high scale. This example doesn't make use of an App Service Environment to reduce costs, but the sample architecture could support it, with modifications.

Scenario details

Bots allow Teams users to interact with web services through text, interactive cards, and task modules. The Microsoft Bot Framework and Azure Bot Services give you an easy-to-use set of tools for creating and managing these bots.

You can develop bots by using a variety of languages, such as C#, JavaScript, and Python. After they're developed, you can deploy them to Azure. A key component of a bot is the web app, which contains the core logic and interface that users communicate with. One of the key requirements for the bot to work is that it must expose a publicly accessible HTTPS endpoint.

InfoSec policy commonly requires that all incoming traffic to web apps go through a corporate firewall. This means that all traffic that goes to a bot, and responses from the bot, must route through a corporate firewall, as with any other web app.

Potential use cases

Organizations can utilize bots for mobile and desktop users. Some examples include:

- Simple queries. Bots can deliver an exact match to a query or a group of related matches to help with disambiguation.
- Multi-turn interactions. By helping anticipate possible next steps, bots make it much easier for people to a complete task flow.
- Reaching out to users. Bots can send a message when something has changed in a document or a work item is closed.

Considerations

Monitoring

Although monitoring isn't implemented in this example scenario, a bot's app service can utilize [Azure Monitor](#) services to monitor its availability and performance.

Scalability

The bots used in this scenario are hosted on Azure App Service. As a result, you can use the standard App Service autoscaling features to automatically scale the number of instances running your bot, which allows your bot to keep up with demand. For more information about autoscaling, see [Autoscaling best practices](#).

For other scalability topics, see the Azure Architecture Center [Performance efficiency checklist](#).

DevOps

It's a common practice to deploy web apps, API apps, and mobile apps to an Azure App Service plan by using continuous deployment pipelines. Because a secured bot's app service is protected with a private endpoint, externally hosted build agents don't have the access that's required to deploy updates. To work around this, you might need to use a solution such as Azure Pipeline [self-hosted DevOps agents](#).

Security

[Azure DDoS Protection](#), combined with application-design best practices, provides enhanced DDoS mitigation features to provide more defense against DDoS attacks. You should enable [Azure DDOS Protection](#) on any perimeter virtual network.

Deploy this scenario

Prerequisites

You must have an existing Azure account. If you don't have an Azure subscription, create a [free account](#) before you begin.

Walkthrough

1. Run the following Azure CLI commands in Azure Cloud Shell or your preferred deployment shell.

This set of commands creates the necessary resource group, virtual network, and subnets that are required for this walkthrough. The IP range used by Teams is [52.112.0.0/14,52.122.0.0/15](#).

```

# Declare variables (bash syntax)
export PREFIX='SecureBot'
export RG_NAME='rg-${PREFIX}'
export VNET_NAME='vnet-${PREFIX}'
export SUBNET_INT_NAME='VnetIntegrationSubnet'
export SUBNET_PVT_NAME='PrivateEndpointSubnet'
export LOCATION='eastus'
export TEAMS_IP_RANGE='52.112.0.0/14 52.122.0.0/15'
export FIREWALL_NAME='afw-${LOCATION}-${PREFIX}'

# Create a resource group
az group create --name ${RG_NAME} --location ${LOCATION}

# Create a virtual network with a subnet for the firewall
az network vnet create \
--name ${VNET_NAME} \
--resource-group ${RG_NAME} \
--location ${LOCATION} \
--address-prefix 10.0.0.0/16 \
--subnet-name AzureFirewallSubnet \
--subnet-prefix 10.0.1.0/26

# Add a subnet for the Virtual network integration
az network vnet subnet create \
--name ${SUBNET_INT_NAME} \
--resource-group ${RG_NAME} \
--vnet-name ${VNET_NAME} \
--address-prefix 10.0.2.0/24

# Add a subnet where the private endpoint will be deployed for the app
# service
az network vnet subnet create \
--name ${SUBNET_PVT_NAME} \
--resource-group ${RG_NAME} \
--vnet-name ${VNET_NAME} \
--address-prefix 10.0.3.0/24

```

When you create a private endpoint subnet, the private endpoint policies are disabled by default.

When the deployment is complete, you should see the following subnets within your virtual network:

Name	IPv4
AzureFirewallSubnet	10.0.1.0/26 (59 available)
VnetIntegrationSubnet	10.0.2.0/24 (251 available)
PrivateLinkSubnet	10.0.3.0/24 (251 available)

2. Deploy an Azure Firewall instance into the firewall subnet that you created in step 1 by running the following CLI commands:

```
Azure CLI

# Create a firewall
az network firewall create \
    --name ${FIREWALL_NAME} \
    --resource-group ${RG_NAME} \
    --location ${LOCATION}

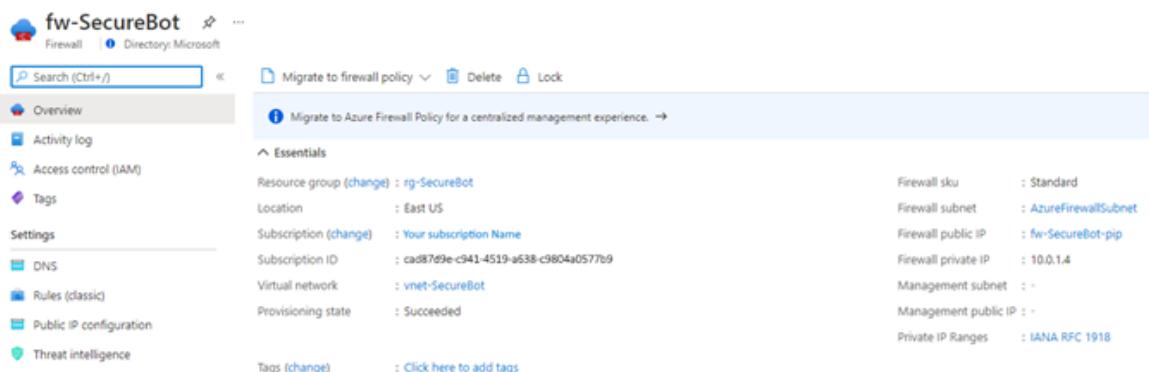
# Create a public IP for the firewall
az network public-ip create \
    --name ${FIREWALL_NAME}-pip \
    --resource-group ${RG_NAME} \
    --location ${LOCATION} \
    --allocation-method static \
    --sku standard

# Associate the IP with the firewall
az network firewall ip-config create \
    --firewall-name ${FIREWALL_NAME} \
    --name ${FIREWALL_NAME}-Config \
    --public-ip-address ${FIREWALL_NAME}-pip \
    --resource-group ${RG_NAME} \
    --vnet-name ${VNET_NAME}

# Update the firewall
az network firewall update \
    --name ${FIREWALL_NAME} \
    --resource-group ${RG_NAME}

# Get the public IP address for the firewall and take note of it for
# later use
az network public-ip show \
    --name ${FIREWALL_NAME}-pip \
    --resource-group ${RG_NAME}
```

Your firewall configuration should look something like this:



The screenshot shows the Azure Firewall blade for the resource group 'rg-SecureBot'. The 'Overview' tab is selected. The 'Essentials' section displays the following configuration details:

Setting	Value
Resource group	rg-SecureBot
Location	East US
Subscription	Your subscription Name
Subscription ID	ca987d9e-c941-4519-a638-c9804a0577b9
Virtual network	vnet-SecureBot
Provisioning state	Succeeded
Tags	Click here to add tags

Other tabs visible in the sidebar include 'Activity log', 'Access control (IAM)', 'Tags', 'DNS', 'Rules (classic)', 'Public IP configuration', and 'Threat intelligence'.

3. Create a basic bot.

4. Deploy the basic bot into the resource group that you created in step 1.

As part of this process, you'll create an app registration, which you need to interact with the bot via channels. During this process, you'll also deploy the necessary App Service plan, app service, and web app bot.

ⓘ Note

Select an App Service plan that supports Azure Private Link.

5. Map a custom domain to the app service that you deployed to the resource group in step 3.

This step requires access to your domain registrar, and it requires you to add an A-record to the custom domain that points to the public IP of the firewall you created in step 2.

6. Secure the mapped custom domain by either uploading an existing certificate for the domain or purchasing an App Service Certificate in Azure and importing it. You can do this by following the steps in [Secure a custom DNS name with a TLS/SSL binding in Azure App Service](#).

You should now have a fully functional bot that you can add to a channel in Teams or test through Web Chat by using the directions found in the [Bot Framework SDK documentation](#).

ⓘ Note

At this point the bot's app service is still publicly accessible over both the `azurewebsites.net` URL and over the custom URL you configured. In the next steps, you'll use private endpoints to disable public access. You'll also configure the firewall to allow the bot service to communicate only with Teams clients.

7. Run the following Azure CLI script to [deploy and configure the private endpoint](#).

This step also implements virtual network integration for the bot's app service, which connects it to your virtual network's integration subnet.

Azure CLI

```
# Disable private endpoint network policies (this step is not required
# if you're using the Azure portal)
az network vnet subnet update \
```

```

--name ${SUBNET_PVT_NAME} \
--resource-group ${RG_NAME} \
--vnet-name ${VNET_NAME} \
--disable-private-endpoint-network-policies true

# Create the private endpoint, being sure to copy the correct resource
# ID from your deployment of the bot app service
# The ID can be viewed by using the following CLI command:
# az resource show --name wapp-securebot --resource-group rg-securebot
# --resource-type Microsoft.web/sites --query "id"
az network private-endpoint create \
--name pvt-${PREFIX}Endpoint \
--resource-group ${RG_NAME} \
--location ${LOCATION} \
--vnet-name ${VNET_NAME} \
--subnet ${SUBNET_PVT_NAME} \
--connection-name conn-${PREFIX} \
--private-connection-resource-id /subscriptions/cad87d9e-c941-4519-
a638-c9804a0577b9/resourceGroups/rg-
securebot/providers/Microsoft.Web/sites/wapp-securebot \
--group-id sites

# Create a private DNS zone to resolve the name of the app service
az network private-dns zone create \
--name ${PREFIX}privatelink.azurewebsites.net \
--resource-group ${RG_NAME}

az network private-dns link vnet create \
--name ${PREFIX}-DNSLink \
--resource-group ${RG_NAME} \
--registration-enabled false \
--virtual-network ${VNET_NAME} \
--zone-name ${PREFIX}privatelink.azurewebsites.net

az network private-endpoint dns-zone-group create \
--name chatBotZoneGroup \
--resource-group ${RG_NAME} \
--endpoint-name pvt-${PREFIX}Endpoint \
--private-dns-zone ${PREFIX}privatelink.azurewebsites.net \
--zone-name ${PREFIX}privatelink.azurewebsites.net

# Establish virtual network integration for outbound traffic
az webapp vnet-integration add \
-g ${RG_NAME} \
-n wapp-${PREFIX} \
--vnet ${VNET_NAME} \
--subnet ${SUBNET_INT_NAME}

```

After you've run these commands, you should see the following resources in your resource group:

	Name ↑↓	Type ↑↓	Location ↑↓
<input type="checkbox"/>	asp-wapp-SecureBot	App Service plan	East US
<input type="checkbox"/>	fw-SecureBot	Firewall	East US
<input type="checkbox"/>	fw-SecureBot-pip	Public IP address	East US
<input type="checkbox"/>	pvt-SecureBotEndpoint	Private endpoint	East US
<input type="checkbox"/>	pvt-SecureBotEndpoint.nic.f66d6885-13f6-4db9-84c1-d268723fb49c	Network interface	East US
<input type="checkbox"/>	securebotprivatelink.azurewebsites.net	Private DNS zone	Global
<input type="checkbox"/>	vnet-SecureBot	Virtual network	East US
<input type="checkbox"/>	wapp-securebot	App Service	East US

The **VNet Integration** option under the **Networking** section of your app service should look like this:

wapp-securebot | Networking

App Service | Directory: Microsoft

Search (Ctrl + /)

VNet Integration

Securely access resources available in or through your Azure VNet. [Learn More](#)

[Click here to configure](#)

Private Endpoint connections

Private access to services hosted on the Azure platform, keeping your data on the Azure network. [Learn More](#)

[Configure your private endpoint connections](#)

Hybrid connections

Securely access applications in private networks. [Learn More](#)

[Configure your hybrid connection endpoints](#)

Azure Front Door with Web Application Firewall

Scalable and secure entry point for accelerated delivery of your web applications. [Learn More](#)

[Configure Azure Front Door with WAF for your app](#)

Azure CDN

Secure, reliable content delivery with broad global reach and rich feature set. [Learn More](#)

[Configure Azure CDN for your app](#)

Access Restrictions

[Networking](#)

Home > wapp-securebot >

VNet Integration ...

wapp-securebot

Disconnect Refresh

VNet Configuration

Securely access resources available in or through your Azure VNet. [Learn more](#)

VNet Details

VNet NAME	vnet-SecureBot
LOCATION	East US

VNet Address Space

Start Address	End Address
10.0.0.0	10.0.255.255

Subnet Details

Subnet NAME	VnetIntegrationSubnet
-------------	-----------------------

Subnet Address Space

Start Address	End Address
10.0.2.0	10.0.2.255

Home > wapp-securebot >

Private Endpoint connections ...

+ Add Refresh | ✓ Approve ✘ Reject 🗑 Remove

Private Endpoint connections

Private access to services hosted on the Azure platform, keeping your data on the Microsoft network. [Learn more](#)

Filter by name or description	All connection states	Connection name ↑↓	Connection state ↑↓	Private endpoint ↑↓
conn-SecureBot-8c1b4502-afaf-424d-8744-cfa0469cfabb	Approved	conn-SecureBot-8c1b4502-afaf-424d-8744-cfa0469cfabb	Approved	pvt-SecureBotEndpoint

8. Next, you create a route table to ensure that traffic to and from each subnet goes through the firewall. You'll need the private IP address of the firewall that you created in the previous step.

Azure CLI

```

# Create a route table
az network route-table create \
-g ${RG_NAME} \
-n rt-${PREFIX}RouteTable

# Create a default route with 0.0.0.0/0 prefix and the next hop as the
# Azure firewall virtual appliance to inspect all traffic. Make sure you
# use your firewall's internal IP address instead of 10.0.1.4
az network route-table route create -g ${RG_NAME} \
--route-table-name rt-${PREFIX}RouteTable -n default \
--next-hop-type VirtualAppliance \
--address-prefix 0.0.0.0/0 \
--next-hop-ip-address 10.0.1.4

```

```

# Associate the two subnets with the route table
az network vnet subnet update -g ${RG_NAME} \
-h ${SUBNET_INT_NAME} --vnet-name ${VNET_NAME} \
--route-table rt-${PREFIX}RouteTable

az network vnet subnet update -g ${RG_NAME} \
-h ${SUBNET_PVT_NAME} \
--vnet-name ${VNET_NAME} \
--route-table rt-${PREFIX}RouteTable

```

After you've run the commands, your route table resource should look like this:

Name	Address prefix	Next hop type
default	0.0.0.0	10.0.1.4

After you've created the route table, you add rules to your firewall to deliver traffic from the public IP to the bot app service, and to restrict traffic from any endpoint other than Microsoft Teams. In addition, you'll allow traffic between the virtual network and Azure Bot Services or Microsoft Entra ID by using service tags.

9. Run the following commands:

Azure CLI

```

# Create a NAT rule collection and a single rule. The source address is
# the public IP range of Microsoft Teams
# Destination address is that of the firewall.
# The translated address is that of the app service's private link.
az network firewall nat-rule create \
--resource-group ${RG_NAME} \
--collection-name coll-${PREFIX}-nat-rules \
--priority 200 \
--action DNAT \
--source-addresses ${TEAMS_IP_RANGE} \
--dest-addr 23.100.26.84 \
--destination-ports 443 \
--firewall-name ${FIREWALL_NAME} \
--name rl-ip2appservice \
--protocols TCP \
--translated-address 10.0.3.4 \

```

```

--translated-port 443

# Create a network rule collection and add three rules to it.
# The first one is an outbound network rule to only allow traffic to
# the Teams IP range.
# The source address is that of the virtual network address space,
# destination is the Teams IP range.
az network firewall network-rule create \
    --resource-group ${RG_NAME} \
    --collection-name coll-${PREFIX}-network-rules \
    --priority 200 \
    --action Allow \
    --source-addresses 10.0.0.0/16 \
    --dest-addr ${TEAMS_IP_RANGE} \
    --destination-ports 443 \
    --firewall-name ${FIREWALL_NAME} \
    --name rl-OutboundTeamsTraffic \
    --protocols TCP

# This rule will enable traffic to all IP addresses associated with
# Azure AD service tag
az network firewall network-rule create \
    --resource-group ${RG_NAME} \
    --collection-name coll-${PREFIX}-network-rules \
    --source-addresses 10.0.0.0/16 \
    --dest-addr AzureActiveDirectory \
    --destination-ports '*' \
    --firewall-name ${FIREWALL_NAME} \
    --name rl-AzureAD \
    --protocols TCP

# This rule will enable traffic to all IP addresses associated with
# Azure Bot Services service tag
az network firewall network-rule create \
    --resource-group ${RG_NAME} \
    --collection-name coll-${PREFIX}-network-rules \
    --source-addresses 10.0.0.0/16 \
    --dest-addr AzureBotService \
    --destination-ports '*' \
    --firewall-name ${FIREWALL_NAME} \
    --name rl-AzureBotService \
    --protocols TCP

```

After you've run the commands, your firewall rules will look something like this:

Name	coll-SecureBot-nat-rules						
Priority *	200						
Action	Destination Network Address Translation (DNAT)						
Rules							
name	Protocol	Source type	Source	Destination Addresses	Destination Ports	Translated address	Translated port
rl-ip2appservice	TCP	IP address	52.112.0.0/14	23.100.26.84	443	10.0.3.4	443

IP Addresses						
name	Protocol	Source type	Source	Destination type	Destination Addresses	Destination Ports
rl-OutboundTeamsTraffic	TCP	IP address	10.0.0.0/16	IP address	52.112.0.0/14	443
			*, 192.168.10.1, 192.168.10.0/24, ...		*, 192.168.10.1, 192.168.10.0/24, ...	8080, 8080-8090, *

Service Tags						
name	Protocol	Source type	Source	Service Tags	Destination Ports	
rl-AzureAD	TCP	IP address	10.0.0.0/16	AzureActiveDirectory	*	
rl-AzureBotService	TCP	IP address	10.0.0.0/16	AzureBotService	*	
			*, 192.168.10.1, 192.168.10.0/24, 192.16...	0 selected	8080, 8080-8090, *	

FQDNs						
name	Protocol	Source type	Source	Destination FQDNs	Destination Ports	
			*, 192.168.10.1, 192.168.10.0/24, 192.16...	time.windows.com	8080, 8080-8090, *	

10. Confirm that your bot is accessible only from a channel in Teams, and that all traffic to and from the bot app service goes through your firewall.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ali Jafry](#) | Cloud Solution Architect

Next steps

- Review the [Bot Framework SDK Documentation](#) to start building bots.
- See [Bots Secured Behind a Firewall & Teams](#).

Related resources

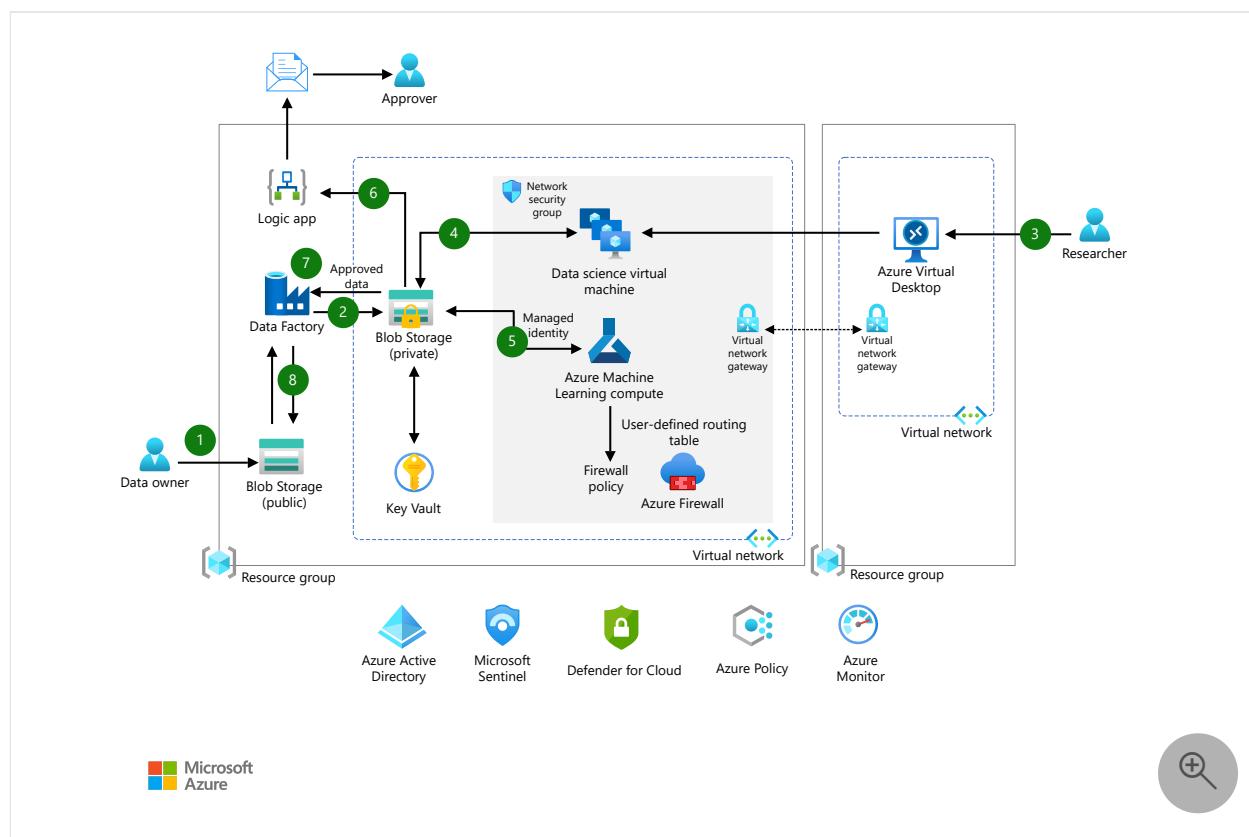
- Visit the [Azure Architecture Center](#) to review related architectures and guides.
- [Azure Firewall Architecture Guide - Azure Architecture Center](#)
- [Microsoft Entra IDaaS in Security Operations - Azure Example Scenarios](#)
- [Threat indicators for cyber threat intelligence in Microsoft Sentinel - Azure Example Scenarios](#)
- [Confidential computing on a healthcare platform - Azure Example Scenarios](#)
- [Hub-spoke network topology in Azure](#)

Secure research environment for regulated data

Azure Data Science Virtual Machines Azure Machine Learning Azure Data Factory

This architecture shows a secure research environment intended to allow researchers to access sensitive data under a higher level of control and data protection. This article is applicable for organizations that are bound by regulatory compliance or other strict security requirements.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Data owners upload datasets into a public blob storage account. The data is encrypted by using Microsoft-managed keys.
2. **Azure Data Factory** uses a trigger that starts copying of the uploaded dataset to a specific location (import path) on another storage account with security controls.

The storage account can only be reached through a private endpoint. Also, it's accessed by a service principal with limited permissions. Data Factory deletes the original copy making the dataset immutable.

3. Researchers access the secure environment through a streaming application using [Azure Virtual Desktop](#) as a privileged jump box.
4. The dataset in the secure storage account is presented to the data science VMs provisioned in a secure network environment for research work. Much of the data preparation is done on those VMs.
5. The secure environment has [Azure Machine Learning](#) compute that can access the dataset through a private endpoint for users for AML capabilities, such as to train, deploy, automate, and manage machine learning models. At this point, models are created that meet regulatory guidelines. All model data is de-identified by removing personal information.
6. Models or de-identified data is saved to a separate location on the secure storage (export path). When new data is added to the export path, a Logic App is triggered. In this architecture, the Logic App is outside the secure environment because no data is sent to the Logic App. Its only function is to send notification and start the manual approval process.

The app starts an approval process requesting a review of data that is queued to be exported. The manual reviewers ensure that sensitive data isn't exported. After the review process, the data is either approved or denied.

 **Note**

If an approval step is not required on exfiltration, the Logic App step could be omitted.

7. If the de-identified data is approved, it's sent to the Data Factory instance.
8. Data Factory moves the data to the public storage account in a separate container to allow external researchers to have access to their exported data and models. Alternately, you can provision another storage account in a lower security environment.

Components

This architecture consists of several Azure services that scale resources according to need. The services and their roles are described below. For links to product documentation to get started with these services, see [Next steps](#).

Core workload components

Here are the core components that move and process research data.

- **Azure Data Science Virtual Machine (DSVM):** [VMs](#) that are configured with tools used for data analytics and machine learning.
- **Azure Machine Learning:** [Used](#) to train, deploy, automate, and manage machine learning models and to manage the allocation and use of ML compute resources.
- **Azure Machine Learning Compute:** A cluster of nodes that are used to train and test machine learning and AI models. The compute is allocated on demand based on an automatic scaling option.
- **Azure Blob storage:** [There](#) are two instances. The public instance is used to temporarily store the data uploaded by data owners. Also, it stores deidentified data after modeling in a separate container. The second instance is private. It receives the training and test data sets from Machine Learning that are used by the training scripts. Storage is mounted as a virtual drive onto each node of a Machine Learning Compute cluster.
- **Azure Data Factory:** [Automatically](#) moves data between storage accounts of differing security levels to ensure separation of duties.
- **Azure Virtual Desktop** [is used](#) as a jump box to gain access to the resources in the secure environment with streaming applications and a full desktop, as needed. Alternately, you can use [Azure Bastion](#). But, have a clear understanding of the security control differences between the two options. Virtual Desktop has some advantages:
 - Ability to stream an app like VSCode to run notebooks against the machine learning compute resources.
 - Ability to limit copy, paste, and screen captures.
 - Support for Microsoft Entra authentication to DSVM.
- **Azure Logic Apps** [provides](#) automated low-code workflow to develop both the *trigger* and *release* portions of the manual approval process.

Posture management components

These components continuously monitor the posture of the workload and its environment. The purpose is to discover and mitigate risks as soon as they are discovered.

- [Microsoft Defender for Cloud](#) is used to evaluate the overall security posture of the implementation and provide an attestation mechanism for regulatory compliance. Issues that were previously found during audits or assessments can be discovered early. Use features to track progress such as secure score and compliance score.
- [Microsoft Sentinel](#) is Security Information and Event Management (SIEM) and security orchestration automated response (SOAR) solution. You can centrally view logs and alerts from various sources and take advantage of advanced AI and security analytics to detect, hunt, prevent, and respond to threats.
- [Azure Monitor](#) provides observability across your entire environment. View metrics, activity logs, and diagnostics logs from most of your Azure resources without added configuration. Management tools, such as those in Microsoft Defender for Cloud, also push log data to Azure Monitor.

Governance components

- [Azure Policy](#) helps to enforce organizational standards and to assess compliance at-scale.

Alternatives

- This solution uses Data Factory to move the data to the public storage account in a separate container, in order to allow external researchers to have access to their exported data and models. Alternately, you can provision another storage account in a lower security environment.
- This solution uses Azure Virtual Desktop as a jump box to gain access to the resources in the secure environment, with streaming applications and a full desktop. Alternately, you can use Azure Bastion. But, Virtual Desktop has some advantages, which include the ability to stream an app, to limit copy/paste and screen captures, and to support AAC authentication. You can also consider configuring Point to Site VPN for offline training locally. This will also help save costs of having multiple VMs for workstations.
- To secure data at rest, this solution encrypts all Azure Storage with Microsoft-managed keys using strong cryptography. Alternately, you can use customer-managed keys. The keys must be stored in a managed key store.

Scenario details

Potential use cases

This architecture was originally created for higher education research institutions with HIPAA requirements. However, this design can be used in any industry that requires isolation of data for research perspectives. Some examples include:

- Industries that process regulated data as per NIST requirements
- Medical centers collaborating with internal or external researchers
- Banking and finance

By following the guidance you can maintain full control of your research data, have separation of duties, and meet strict regulatory compliance standards while providing collaboration between the typical roles involved in a research-oriented workload; data owners, researchers, and approvers.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

The main objective of this architecture is to provide a secure and trusted research environment that strictly limits the exfiltration of data from the secure area.

Network security

Azure resources that are used to store, test, and train research data sets are provisioned in a secure environment. That environment is an Azure Virtual Network (VNet) that has network security groups (NSGs) rules to restrict access, mainly:

- Inbound and outbound access to the public internet and within the VNet.
- Access to and from specific services and ports. For example, this architecture blocks all ports ranges except the ones required for Azure Services (such as Azure

Monitor). A full list of Service Tags and the corresponding services can be found [here](#).

Also, access from VNet with Azure Virtual Desktop (AVD) on ports limited to approved access methods is accepted, all other traffic is denied. When compared to this environment, the other VNet (with AVD) is relatively open.

The main blob storage in the secure environment is off the public internet. It's only accessible within the VNet through [private endpoint connections](#) and Azure Storage Firewalls. It's used to limit the networks from which clients can connect to Azure file shares.

This architecture uses credential-based authentication for the main data store that is in the secure environment. In this case, the connection information like the subscription ID and token authorization is stored in a key vault. Another option is to create identity-based data access, where your Azure account is used to confirm if you have access to the Storage service. In an identity-based data access scenario, no authentication credentials are saved. For the details on how to use identity-based data access, see [Connect to storage by using identity-based data access](#).

The compute cluster can solely communicate within the virtual network, by using the Azure Private Link ecosystem and service/private endpoints, rather than using Public IP for communication. Make sure you enable **No public IP**. For details about this feature, which is currently in preview (as of 3/7/2022), see [No public IP for compute instances](#).

The secure environment uses Azure Machine Learning compute to access the dataset through a private endpoint. Additionally, Azure Firewall can be used to control outbound access from Azure Machine Learning compute. To learn about how to configure Azure Firewall to control access to Azure Machine Learning compute, which resides in a machine learning workspace, see [Configure inbound and outbound network traffic](#).

To learn one of the ways to secure an Azure Machine Learning environment, see the blog post, [Secure Azure Machine Learning Service \(AMLS\) Environment](#) ↗.

For Azure services that cannot be configured effectively with private endpoints, or to provide stateful packet inspection, consider using Azure Firewall or a third-party network virtual appliance (NVA).

Identity management

The Blob storage access is through Azure Role-based access controls (RBAC).

Azure Virtual Desktop supports Microsoft Entra authentication to DSVM.

Data Factory uses managed identity to access data from the blob storage. DSVMs also uses managed identity for remediation tasks.

Data security

To secure data at rest, all Azure Storage is encrypted with Microsoft-managed keys using strong cryptography.

Alternately, you can use customer-managed keys. The keys must be stored in a managed key store. In this architecture, Azure Key Vault is deployed in the secure environment to store secrets such as encryption keys and certificates. Key Vault is accessed through a private endpoint by the resources in the secure VNet.

Governance considerations

Enable Azure Policy to enforce standards and provide automated remediation to bring resources into compliance for specific policies. The policies can be applied to a project subscription or at a management group level as a single policy or as part of a regulatory initiative.

For example, in this architecture Azure Policy Guest Configuration was applied to all VMs in scope. The policy can audit operating systems and machine configuration for the Data Science VMs.

VM image

The Data Science VMs run customized base images. To build the base image, we highly recommend technologies like Azure Image Builder. This way you can create a repeatable image that can be deployed when needed.

The base image might need updates, such as additional binaries. Those binaries should be uploaded to the public blob storage and flow through the secure environment, much like the datasets are uploaded by data owners.

Other considerations

Most research solutions are temporary workloads and don't need to be available for extended periods. This architecture is designed as a single-region deployment with availability zones. If the business requirements demand higher availability, replicate this architecture in multiple regions. You would need other components, such as global load

balancer and distributor to route traffic to all those regions. As part of your recovery strategy, capturing and creating a copy of the customized base image with Azure Image Builder is highly recommended.

The size and type of the Data Science VMs should be appropriate to the style of work being performed. This architecture is intended to support a single research project and the scalability is achieved by adjusting the size and type of the VMs and the choices made for compute resources available to AML.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

The cost of DSVMs depends on the choice of the underlying VM series. Because the workload is temporary, the consumption plan is recommended for the Logic App resource. Use the [Azure pricing calculator](#) to estimate costs based on estimated sizing of resources needed.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Clayton Barlow](#) | Senior Azure Specialist

Next steps

- [Microsoft Data Science Virtual Machine \(DSVM\)](#)
- [What is Azure Machine Learning?](#)
- [Azure Machine Learning Compute](#)
- [Introduction to Azure Blob storage](#)
- [Introduction to Azure Data Factory](#)
- [Azure Virtual Desktop](#)
- [Microsoft Defender for Cloud](#)
- [Microsoft Sentinel](#)
- [Azure Monitor](#)
- [Azure Policy](#)

- [Azure Policy Guest Configuration](#)

Related resources

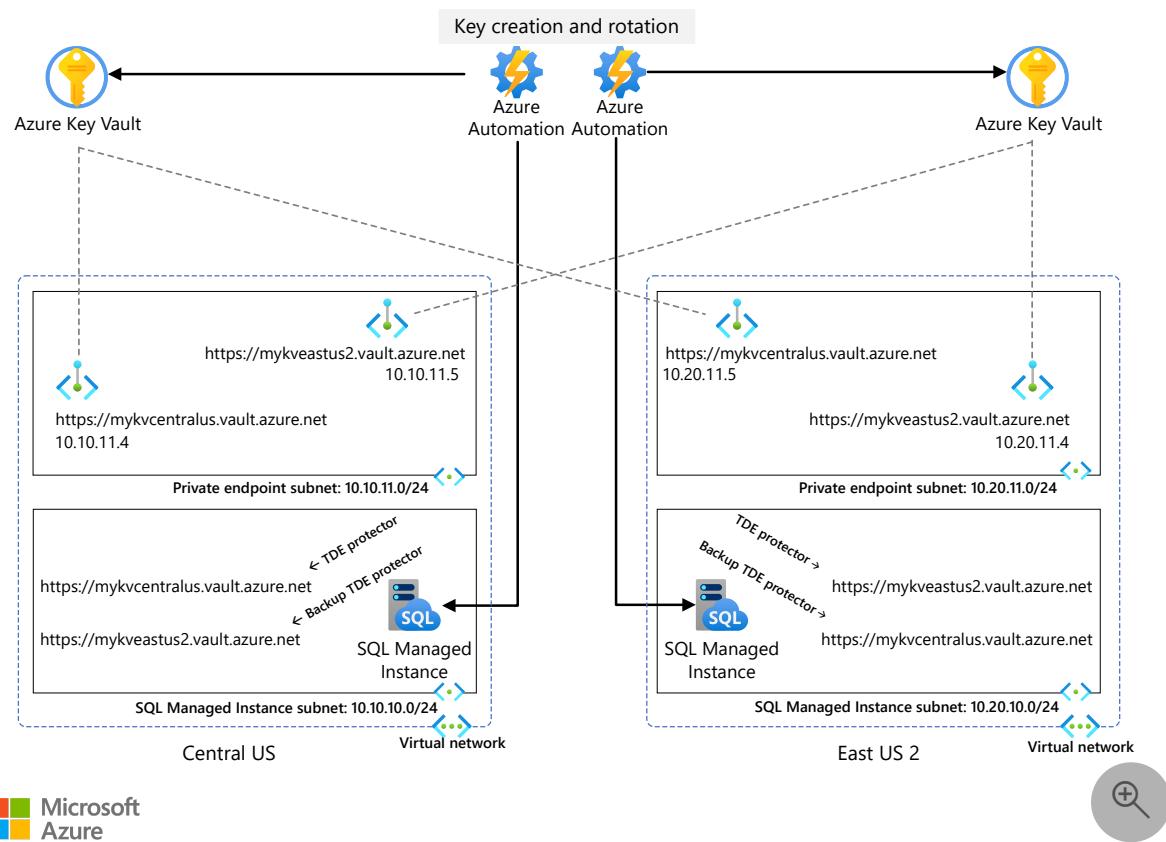
- [Compare the machine learning products and technologies from Microsoft](#)
- [Azure Machine Learning architecture](#)
- [Scale AI and machine learning initiatives in regulated industries](#)
- [Many models machine learning \(ML\) at scale with Azure Machine Learning](#)

SQL Managed Instance with customer-managed keys

Azure SQL Managed Instance Azure Key Vault Azure Private Link

This article describes how you can manage your own Transparent Data Encryption (TDE) keys for SQL managed instances in a cross-region auto-failover group by using Azure Key Vault.

Architecture



Download a [Visio file](#) of this architecture.

For greater redundancy of the TDE keys, Azure SQL Managed Instance is configured to use the key vault in its own region as the primary and the key vault in the remote region as the secondary.

The secondary key vault instance, while in a remote region, has a [private endpoint](#) in the same region as the SQL managed instance. So, as far as a SQL managed instance is concerned, requests made to both primary and secondary key vaults are logically within

the same virtual network and region. This design allows for easier firewall or network security group rules. Many organizations use a private endpoint rather than accessing the public endpoint. We recommend that you use a private endpoint.

Dataflow

1. Every 10 minutes, SQL Managed Instance checks to make sure it can access the TDE wrapper at the key vault that's defined as primary.
2. If the primary key vault of SQL Managed Instance becomes unavailable, that instance checks the key vault that's set as secondary. If that key vault is also unavailable, [SQL Managed Instance marks the databases as "inaccessible."](#)

Components

- [Key Vault](#) is a cloud service for storing and accessing secrets with enhanced security. In this architecture, it's used to store keys that are used by TDE. You can also use it to create keys.
- [SQL Managed Instance](#) is a managed instance in Azure that's based on the latest stable version of SQL Server. In this architecture, the key management process is applied to data that's stored in SQL Managed Instance.
- [Azure Private Link](#) enables you to access Azure PaaS services and Azure-hosted services over a private endpoint in your virtual network.

Alternatives

- Instead of using customer-managed TDE keys, you can use service-managed TDE keys. When you use service-managed keys, Microsoft handles securing and rotating the keys. The entire process is abstracted away from you.
- An alternative to having key vaults in two regions is to just have one in a single region. SQL Managed Instance can access keys from a vault that's in another region. You can still use private endpoint. The traffic to Key Vault is low and infrequent, so any latency isn't noticeable. SQL Managed Instance only queries the vault to see whether the key exists. It doesn't copy the material.

Scenario details

When you use customer-managed keys (CMK), also referred to as bring your own key (BYOK), you're responsible for the security, availability, and optional rotation of the keys. These responsibilities are critical because if the key is lost, the databases and backups

are also permanently lost. This article describes the key management process and provides options so that you have the information you need to make an informed decision about the best process for your business.

Potential use cases

Many organizations have policies that require that certificates or encryption keys be created and managed internally. If your organization has a similar policy, this architecture might apply to you. If your customers require internal management of these items, the architecture also might apply to you. If neither of those situations apply, consider using system-managed keys.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

General recommendations

See these articles:

- [Recommendations for configuring customer-managed TDE](#)
- [Recommendations for configuring TDE protector](#)

Key management

Your method of key rotation will differ depending on what you're using to create your TDE asymmetric keys. When you bring your own TDE wrapper key, you have to decide how you'll create this key. Your options are:

- Use Key Vault to create the keys. This option ensures that the private key material never leaves Key Vault and can't be seen by any human or system. The private keys aren't exportable, but they can be backed up and restored to another key vault. This point is important. In order to have the same key material in multiple key vaults, as required by this design, you have to use the [backup and restore feature](#). This option has several [limitations](#). **Both key vaults must be in the same Azure geography and subscription**. If they aren't, the restore won't work. The only way around this limitation is to keep the key vaults in separate subscriptions and move one subscription to another region.

- Generate the asymmetric keys offline by using a utility like OpenSSL and then import the keys into Key Vault. When you import a key into Key Vault, you can [mark it as exportable](#). If you do that, you can either throw away the keys after you import them into Key Vault or you can store them somewhere else, like on-premises or in another key vault. This option gives you the most flexibility. However, it can be the least secure if you don't properly ensure the keys don't get into the wrong hands. The system generating the keys and the method used to place the keys in Key Vault aren't controlled by Azure. You can automate this process by using [Azure DevOps](#), [Azure Automation](#), or another orchestration tool.
- Use a [supported on-premises hardware security module \(HSM\)](#) to generate your keys. By using a supported HSM, you can import keys into Key Vault with improved security. The same-geography limitation described earlier doesn't apply when you use an HSM. This option provides a high level of safety for your keys because the key material is in three separate places (two key vaults in Azure and on-premises). This option also provides the same level of flexibility, if you use a supported HSM.

Availability

When you add Key Vault to your architecture, it becomes a critical component. At least one of the key vaults in the design must be accessible. Additionally, the keys that are necessary for TDE must be accessible. Azure Monitor Insights provides comprehensive monitoring of Key Vault. For more information, see [Monitoring your key vault service](#).

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

When you move from service-managed keys to customer-managed keys, your operations will be:

- [Generating keys or importing them to Key Vault](#)
- [Rotating keys](#)
- [Backing up and restoring keys](#)
- [Monitoring customer-managed TDE](#)

DevOps

You can use [Azure Pipelines](#) in Azure DevOps to automate the [key rotation process](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

SQL Managed Instance auto-failover groups [perform significantly better when you use paired regions](#).

SQL Managed Instance only checks to see whether the key exists, and it only does that every 10 minutes. Therefore, SQL Managed Instances doesn't require region-affinity with Key Vault. The location of your TDE keys has no effect on performance.

Scalability

When it comes to managing your TDE keys, scaling isn't a concern. The request size and frequency are so small that you won't need to scale.

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

The biggest security consideration is ensuring that you keep your TDE wrapper key safe and always available to SQL Managed Instances. Any database encrypted via TDE is inaccessible if it can't access the required key in Key Vault. If you use service-managed keys, you don't have to worry about this consideration.

Resiliency

Each SQL managed instance is configured to use two key vaults. If the SQL managed instance primary TDE key is unavailable or inaccessible, the instance attempts to find a key with a matching thumbprint in the secondary key vault.

Cost optimization

For information about the additional costs of managing your own TDE keys, outside of added operational costs, see these resources:

- [Azure Key Vault pricing](#)
- [Private endpoint pricing](#)

For information about the optional components, see these resources:

- [Azure DevOps pricing ↗](#)
- [Azure Automation pricing ↗](#)

Deploy this scenario

You can deploy this scenario by using these ARM templates:

- [SQL Managed Instance ↗](#)
- [Azure Key Vault ↗](#)

Contributors

This article is being updated and maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ahmet Arsan ↗](#) | Senior Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Encryption of backup data using customer-managed keys](#)
- [What is Azure SQL Managed Instance?](#)
- [Azure Key Vault basic concepts](#)

Related resources

- [High availability for Azure SQL Database and SQL Managed Instance](#)

Virtual network integrated serverless microservices

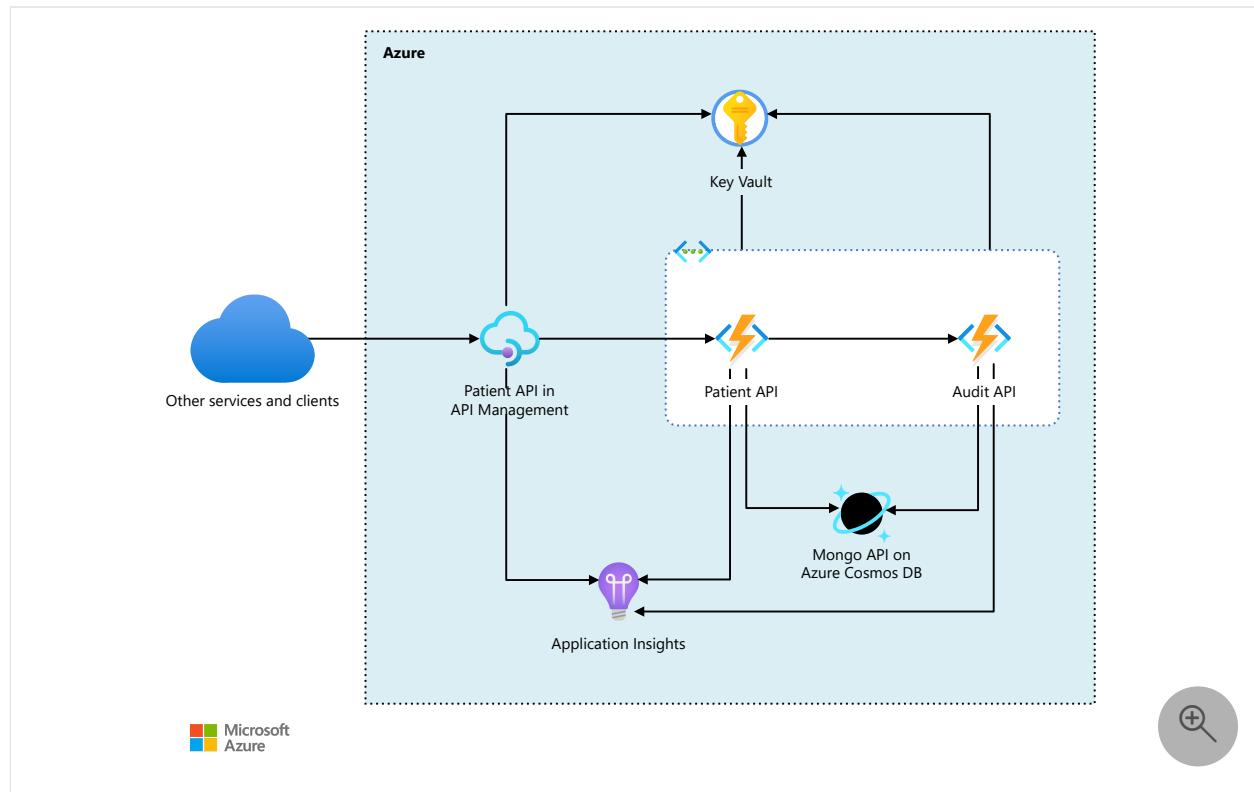
Azure API Management Azure Cosmos DB Azure Functions Azure Key Vault Azure Virtual Network

In this Azure solution, [Azure API Management \(APIM\)](#) controls access to the API through a single managed endpoint. The application backend consists of two interdependent [Azure Functions](#) microservice apps that create and manage patient records and audit records. APIM and the two function apps access each other through a locked-down [virtual network](#).

This article and the [associated code project](#) distill the example scenario down to the main technical components, to serve as scaffolding for specific implementations. The solution automates all code and infrastructure deployments with [Terraform](#), and includes automated integration, unit, and load testing.

Architecture

The following diagram shows the patient record creation request flow:



Download a [Visio file](#) of this architecture.

Workflow

1. Outside services and clients make a POST request to APIM, with a data body that includes patient information.
2. APIM calls the `CreatePatient` function in the **Patient API** with the given patient information.
3. The `CreatePatient` function in **Patient API** calls the `CreateAuditRecord` function in the **Audit API** function app to create an audit record.
4. The **Audit API** `CreateAuditRecord` function creates the audit record in Azure Cosmos DB, and returns a success response to the **Patient API** `CreatePatient` function.
5. The `CreatePatient` function creates the patient document in Azure Cosmos DB, and returns a success response to APIM.
6. The outside services and clients receive the success response from APIM.

Components

The solution uses the following components:

- [Azure API Management \(APIM\)](#) ↗ is a hybrid, multicloud platform for managing APIs across all environments. In this solution, APIM controls internal and third-party access to the Patient API that allows reading and/or writing data. APIM allows for easy integration with different authentication mechanisms.
- [Azure Functions](#) is a serverless compute platform that handles small, event-driven pieces of code. The cloud infrastructure provides the necessary updated servers to run the functions at scale. The current solution uses a set of two Azure Functions API microservices that create and manage operations for patient test results and auditing records.
- [Azure Virtual Network](#) provides an isolated and highly secure application environment by restricting network access to specific IP addresses or subnets. Both APIM and Azure Functions support access restriction and deployment in virtual networks. This solution uses [regional virtual network integration](#) to deploy both function apps in the same virtual network in the same region.
- [Azure Key Vault](#) centrally stores, encrypts, and manages access to keys, certificates, and connection strings. This solution maintains the Azure Functions host keys and Azure Cosmos DB connection strings in a Key Vault that only specified identities can access.

- [Azure Cosmos DB](#) is a fully managed serverless database with instant, automatic scaling. In the current solution, both microservices store data in Azure Cosmos DB, using the [MongoDB Node.js driver](#). The services don't share data, and you can deploy each service to its own independent database.
- [Application Insights](#), a feature of [Azure Monitor](#), reports on application performance, usage, availability, and behavior to detect and help diagnose anomalies.

Failures in microservices-based architecture are often distributed over a variety of components, and can't be diagnosed by looking at the services in isolation. The ability to correlate telemetry across components is vital to diagnosing these issues. Application Insights telemetry centralizes logging along the whole request pipeline to detect performance anomalies. The telemetry shares a common operation ID, allowing correlation across components.

APIM and the Azure Functions runtime have built-in support for Application Insights to generate and correlate a wide variety of telemetry, including standard application output. The function apps use the Application Insights Node.js SDK to manually track dependencies and other custom telemetry.

For more information about the distributed telemetry tracing in this solution, see [Distributed telemetry](#).

Alternatives

- The current solution requires a subscription key to access the APIM endpoint, but you can also use [Microsoft Entra authentication](#).
- In addition to requiring API access keys, you can use Azure Functions' built-in [App Service authentication](#) to enable Microsoft Entra authorization for the APIs' managed identities.
- You can replace the Azure Cosmos DB endpoint in this solution with another MongoDB service without changing the code.
- For additional [Azure Cosmos DB security](#), you can lock down traffic from the Azure Cosmos DB databases to the function apps.
- Components such as Azure Cosmos DB can send telemetry to [Azure Monitor](#), where it can be correlated with the telemetry from Application Insights.
- Instead of Terraform, you can use the Azure portal or Azure CLI for [Key Vault key rotation](#) tasks.
- Instead of Terraform, you can use a system like [Azure DevOps](#) or [GitHub Actions](#) to automate solution deployment.

- For higher availability, this solution can be deployed to multiple regions. Set Azure Cosmos DB to multi-master, use APIM's built-in multi-region support, and deploy the Azure Function apps to paired regions.

Scenario details

This article describes an integrated solution for patient records management. A health organization needs to digitally store large amounts of highly sensitive patient medical test data in the cloud. Internal and third-party systems must be able to securely read and write the data through an application programming interface (API). All interactions with the data must be recorded in an audit register.

Potential use cases

- Access highly sensitive data from designated external endpoints.
- Implement secure auditing for data access operations.
- Integrate interdependent microservices apps with common access and security.
- Use virtual network security features while taking advantage of serverless cost savings and flexibility.

Benefits

Some benefits of serverless applications like Azure Functions are the cost savings and flexibility of using only necessary compute resources, rather than paying up front for dedicated servers. This solution lets Azure Functions use virtual network access restrictions for security, without incurring the cost and operational overhead of full [Azure App Service Environments \(ASEs\)](#).

APIM controls internal and third-party access to a set of API microservices built on Azure Functions. The **Patient API** provides *create, read, update, and delete (CRUD)* operations for patients and their test results. The **Audit API** function app provides operations to create auditing entries.

Each function app stores its data in an independent [Azure Cosmos DB](#) database. [Azure Key Vault](#) securely holds all keys, secrets, and connection strings associated with the apps and databases. Application Insights telemetry and Azure Monitor centralize logging across the system.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Consider the following aspects when implementing this solution.

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

Due to the sensitivity of the data, security is paramount in this solution. The solution uses several mechanisms to protect the data:

- APIM gateway management
- Virtual network access restrictions
- Service access keys and connections strings
- Key and connection string management in Key Vault
- Key Vault key rotation
- Managed service identities

You can protect your Azure API Management instance against distributed denial of service (DDoS) attacks using [Azure DDoS protection](#). Azure DDoS Protection provides enhanced DDoS mitigation features to defend against volumetric and protocol DDoS attacks.

For more details about the security pattern for this solution, see [Security pattern for communication between API Management, Functions apps, and Azure Cosmos DB](#).

API gateway management

The system is publicly accessible only through the single managed APIM endpoint. The APIM subnet restricts incoming traffic to specified gateway node IP addresses.

APIM allows for easy integration with different authentication mechanisms. The current solution requires a subscription key, but you could also use Microsoft Entra ID to secure the APIM endpoint without needing to manage subscription keys in APIM.

Virtual network

To avoid exposing APIs and functions publicly, [Azure Virtual Network](#) restricts network access for APIs and functions to specific IP addresses or subnets. Both API Management and Azure Functions support access restriction and deployment in virtual networks.

Function apps can restrict IPv4, IPv6, and virtual network subnet access. By default, a function app allows all access, but once you add one or more address or subnet restrictions, the app denies all other network traffic.

In this solution, the function apps allow interactions only within their own virtual network. The Patient API allows calls from the APIM subnet by adding the APIM subnet to its access restriction allowlist. The Audit API allows communication with the Patient API by adding the Patient API subnet to its access restriction allowlist. The APIs reject traffic from other sources.

The solution uses [regional virtual network integration](#) to integrate APIM and the function apps with the same virtual network and Azure region. There are several important considerations for using regional virtual network integration:

- You need to use the [Azure Functions Premium SKU](#) to have both regional virtual network integration and scalability.
- You need to use the [APIM Developer or Premium SKU](#) to enable VNET connectivity
- Since you deploy the function apps in a subnet of the virtual network, you configure the function apps' access restrictions to allow traffic from other subnets in the virtual network.
- Regional virtual network integration only limits outbound traffic from the Azure Function to the virtual network. Inbound traffic is still routed outside of the virtual network, although limited by the app's access list.

Only [App Service Environments](#) offer complete network-level virtual network isolation. ASEs can require considerably more expense and effort to implement than Azure Functions that support regional virtual network integration. ASE scaling is also less elastic.

Access keys

You can call APIM and function apps without using access keys. However, disabling the access keys isn't good security practice, so all components in this solution require keys for access.

- Accessing APIM requires a subscription key, so users need to include `Ocp-Apim-Subscription-Key` in HTTP headers.
- All functions in the Patient API function app require an API access key, so APIM must include `x-functions-key` in the HTTP header when calling the Patient API.
- Calling `CreateAuditRecord` in the Audit API function app requires an API access key, so Patient API needs to include `x-functions-key` in the HTTP header when calling the `CreateAuditRecord` function.

- Both Functions apps use Azure Cosmos DB as their data store, so they must use connection strings to access the Azure Cosmos DB databases.

Key Vault storage

Although it's possible to keep access keys and connection strings in the application settings, it's not good practice, because anyone who can access the app can see the keys and strings. The best practice, especially for production environments, is to keep the keys and strings in Azure Key Vault, and use the Key Vault references to call the apps. Key Vault allows access only to specified managed identities.

APIM uses an inbound policy to cache the Patient API host key for improved performance. For subsequent attempts, APIM looks for the key in its cache first.

- APIM retrieves the Patient API host key from Key Vault, caches it, and puts it into an HTTP header when calling the Patient API function app.
- The Patient API function app retrieves the Audit API host key from Key Vault and puts it into an HTTP header when calling the Audit API function app.
- The Azure Function runtime validates the keys in the HTTP headers on incoming requests.

Key rotation

Rotating Key Vault keys helps make the system more secure. You can automatically rotate keys periodically, or you can rotate keys manually or on demand in case of leakage.

Key rotation involves updating several settings:

- The function app host key itself
- The secret in Key Vault that stores the host key
- The Key Vault reference in the function app application settings, to refer to the latest secret version
- The Key Vault reference in the APIM caching policy for the Patient API

The current solution uses Terraform for most of the key rotation tasks. For more information, see [Key rotation pattern with Terraform](#).

Managed identities

In this solution, APIM and the function apps use Azure [system-assigned managed service identities \(MSIs\)](#) to access the Key Vault secrets. Key Vault has the following

individual access policies for each service's managed identity:

- APIM can get the host key of the Patient API function app.
- The Patient API function app can get the Audit API host key and the Azure Cosmos DB connection string for its data store.
- The Audit API function app can get the Azure Cosmos DB connection string for its data store.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

One of the primary benefits of serverless applications like Azure Functions is the cost savings of paying only for consumption, rather than paying up front for dedicated servers. Virtual network support requires the [Azure Functions Premium](#) plan, at additional charge. Azure Functions Premium has support for regional virtual network integration, while still supporting dynamic scaling. The Azure Functions Premium SKU includes virtual network integration on APIM.

For details and pricing calculator, see [Azure Functions pricing](#).

Functions can also be hosted on [App Service virtual machines](#). Only [App Service Environments \(ASEs\)](#) offer complete network-level virtual network isolation. ASEs can be considerably more expensive than an Azure Functions plan that supports regional virtual network integration, and ASE scaling is less elastic.

Deploy this scenario

The source code for this solution is at [Azure VNet-Integrated Serverless Microservices](#).

The [TypeScript](#) source code for the [PatientTest API](#) and the [Audit API](#) are in the `/src` folder. Each API's source includes a [dev container](#) that has all the prerequisites installed, to help you get going quickly.

Both APIs have a full suite of automated integration and unit tests to help prevent regressions when you make changes. The project is also configured for *linting* with ESLint, to maintain code styles and help guard against unintentional errors. The services' respective README files contain information on how to run the tests and linting.

Terraform deployment

The code project's [/env](#) folder includes scripts and templates for [Terraform](#) deployment. Terraform deploys APIM and the function apps, and configures them to use the deployed Application Insights instance. Terraform also provisions all resources and configurations, including networking lockdown and the access key security pattern.

The deployment [README](#) explains how to deploy the Terraform environment in your own Azure subscription. The `/env` folder also includes a [dev container](#) that has all the prerequisites installed for Terraform deployment.

Locust load testing

To gauge API performance, you can run load testing against the APIs with the included [Locust load tests](#). [Locust](#) is an open-source load testing tool, and the tests are written in Python. You can run the load tests locally, or remotely in an Azure Kubernetes Service (AKS) cluster. The tests perform a variety of operations against the APIM endpoint, and verify behaviors against success and failure criteria.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Hannes Nel](#) | Principal Software Engineering Lead

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Use Azure API Management with microservices deployed in Azure Kubernetes Service](#)
- [How to use Azure API Management with virtual networks](#)
- [How to use managed identities for App Service and Azure Functions](#)
- [Use Key Vault references for App Service and Azure Functions](#)
- [APIs and microservices e-book](#)
- [API Management access restriction policies](#)
- [Azure Functions networking options](#)
- [Azure Functions scale and hosting](#)

Related resources

The following architectures cover key API Management scenarios:

- [Migrate a web app using Azure API Management](#)
- [Protect APIs with Application Gateway and API Management](#)
- [Azure API Management landing zone accelerator](#)

The following articles cover key functions scenarios:

- [Integrate Event Hubs with serverless functions on Azure](#)
- [Monitor Azure Functions and Event Hubs](#)
- [Azure Functions in a hybrid environment](#)
- [Performance and scale for Event Hubs and Azure Functions](#)
- [Code walkthrough: Serverless application with Functions](#)
- [Azure App Service and Azure Functions considerations for multitenancy](#)

Centralized app configuration and security

Microsoft Entra ID

Azure App Configuration

Azure Key Vault

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

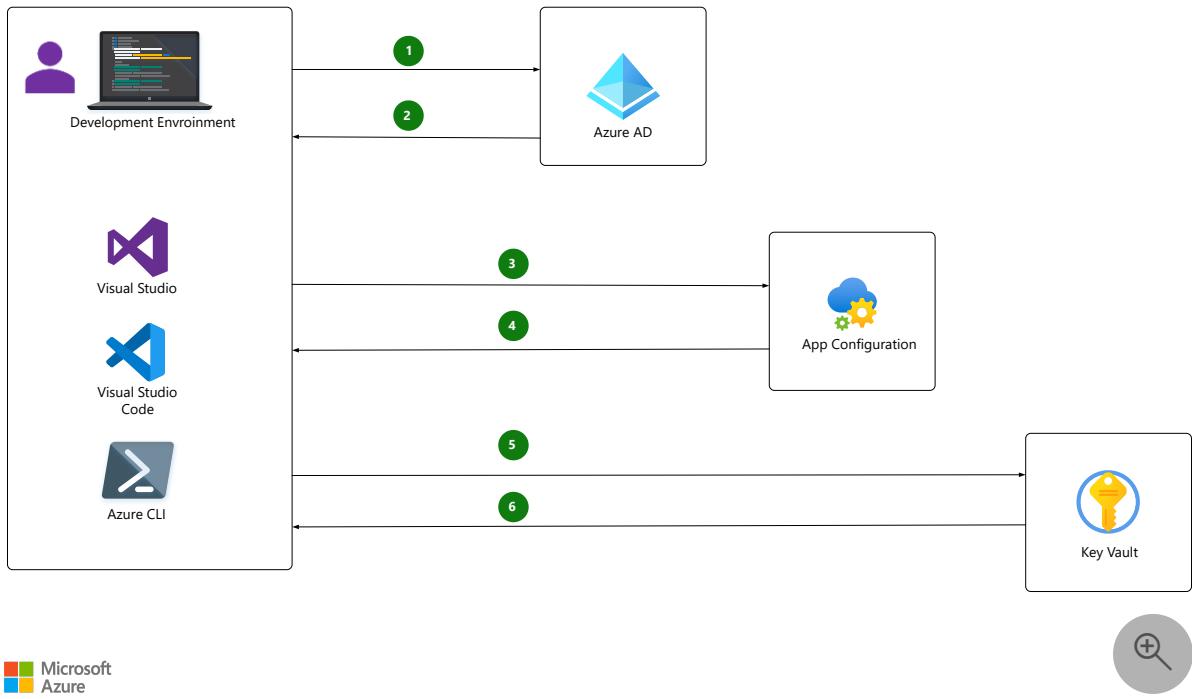
This article outlines a solution for creating a robust and scalable application in a distributed environment. The solution uses Azure App Configuration and Azure Key Vault to manage and store app configuration settings, feature flags, and secure access settings in one place.

Architecture

The following diagrams show how App Configuration and Key Vault can work together to manage and secure apps in **development** and **Azure** environments.

Development environment

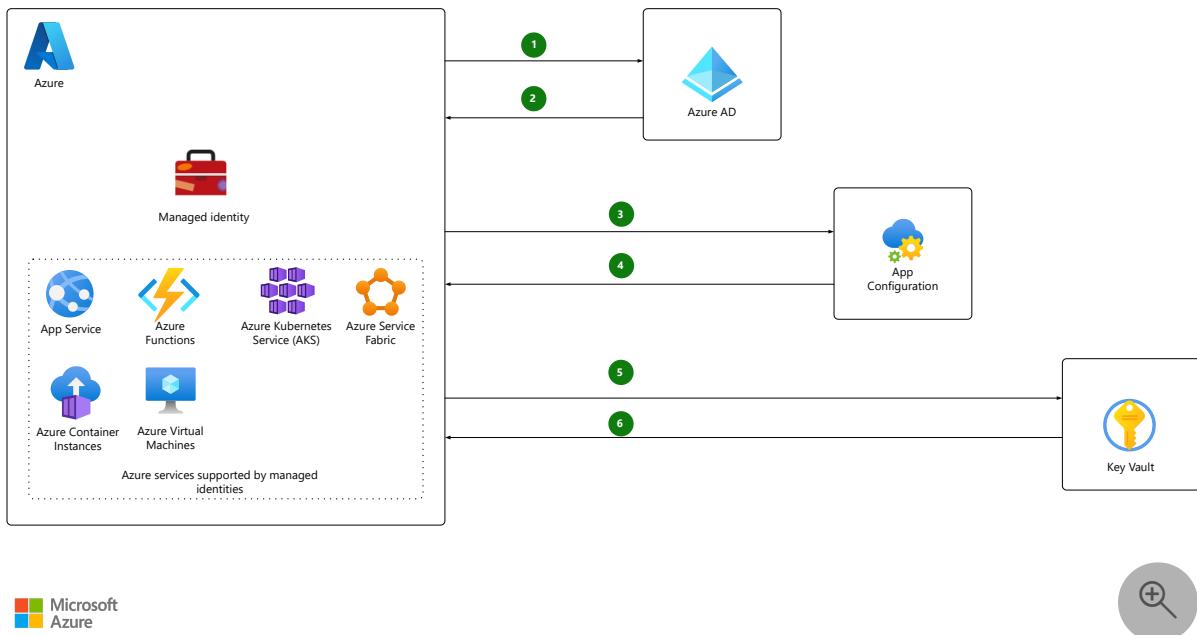
In the development environment, the app uses an identity via Visual Studio or version 2.0 of the Azure CLI to sign in and send an authentication request to Microsoft Entra ID.



[Download a Visio file](#) of this architecture.

Azure staging or production environment

The Azure staging and production environments use a [managed identity](#) for sign-in and authentication.



[Download a Visio file](#) of this architecture.

Dataflow

1. The application sends an authentication request during debugging in Visual Studio, or authenticates via the MSI in Azure.
2. Upon successful authentication, Microsoft Entra ID returns an access token.
3. The App Configuration SDK sends a request with the access token to read the app's App Configuration Key Vault **secretURI** value for the app's key vault.
4. Upon successful authorization, App Configuration sends the configuration value.
5. Utilizing the sign-in identity, the app sends a request to Key Vault to retrieve the application secret for the **secretURI** that App Configuration sent.
6. Upon successful authorization, Key Vault returns the secret value.

Components

- [Microsoft Entra ID](#) is a universal platform for managing and securing identities.
- [App Configuration](#) provides a way to store configurations for all your Azure apps in a universal, hosted location.
- [Managed identities](#) provide an identity for applications to use when connecting to resources that support Microsoft Entra authentication.
- [Key Vault](#) safeguards cryptographic keys and other secrets that are used by cloud apps and services.

Scenario details

Cloud-based applications often run on multiple virtual machines or containers in multiple regions, and use multiple external services. Creating a robust and scalable application in a distributed environment presents a significant challenge.

By using App Configuration, you can manage and store all your app's configuration settings, feature flags, and secure access settings in one place. App Configuration works seamlessly with Key Vault, which stores passwords, keys, and secrets for secure access.

Potential use cases

Any application can use App Configuration, but the following types of applications benefit most from it:

- Microservices that are based on Azure Kubernetes Service (AKS), Azure Service Fabric, or other containerized apps that are deployed in one or more regions.
- Serverless apps, which include Azure Functions or other event-driven stateless compute apps.
- Apps that use a continuous deployment (CD) pipeline.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

- It's best to use a different key vault for each application in each environment: development, Azure pre-production, and Azure production. Using different vaults helps prevent sharing secrets across environments, and reduces threats in the event of a breach.
- To use these scenarios, the sign-in identity must have the **App Configuration Data Reader** role in the App Configuration resource, and have explicit **access policies** for retrieving the secrets in Key Vault.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Sowmyan Soman](#) | Principal Cloud Solution Architect

Next steps

Learn more about the component technologies:

- [Azure App Configuration](#)
- [Azure Key Vault](#)
- [Use Key Vault references for App Service and Azure Functions](#)
- [Use managed identities to access App Configuration](#)
- [Local development and security](#)

Related resources

- [Security architecture design](#)
- [Microservices architecture on Azure Kubernetes Service](#)
- [Microservices architecture on Azure Service Fabric](#)
- [External Configuration Store pattern](#)

Multilayered protection for Azure virtual machine access

Microsoft Entra ID

Azure Bastion

Azure Role-based access control

Microsoft Defender for Cloud

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution provides a multilayered approach for protecting virtual machines (VMs) in Azure. Users need to connect to VMs for management and administrative purposes. It's critical to minimize the attack surface that connectivity creates.

This solution achieves non-persistent granular access to VMs by incorporating several protection mechanisms. It aligns with the *principle of least privilege (PoLP)* and the concept of *separation of duties*. To reduce exposure to attacks, this solution locks down inbound traffic to VMs, but it makes VM connections accessible when needed.

Implementing this type of protection minimizes the risk of many popular cyber attacks on VMs, such as brute-force attacks and distributed denial-of-service (DDoS) attacks.

This solution uses many Azure services and features including:

- Microsoft Entra Privileged Identity Management (PIM).
- The just-in-time (JIT) VM access feature of Microsoft Defender for Cloud.
- Azure Bastion.
- Azure role-based access control (Azure RBAC) custom roles.
- Microsoft Entra Conditional Access, optionally.

Potential use cases

Defense in depth is the main idea behind this architecture. This strategy challenges users with several lines of defense before granting the users access to VMs. The goal is to ensure that:

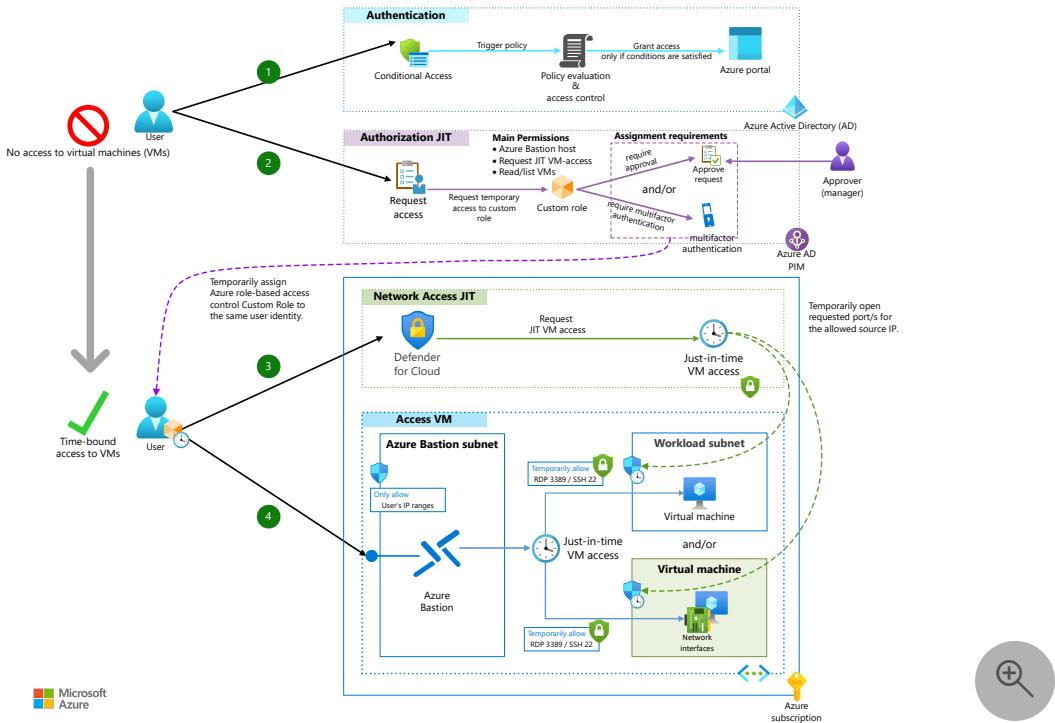
- Each user is legitimate.
- Each user has legal intentions.
- Communication is secure.

- Access to VMs in Azure is only provided when needed.

The defense in depth strategy and the solution in this article apply to many scenarios:

- An administrator needs to access an Azure VM under these circumstances:
 - The administrator needs to troubleshoot an issue, investigate behavior, or apply a critical update.
 - The administrator uses Remote Desktop Protocol (RDP) to access a Windows VM or secure shell (SSH) to access a Linux VM.
 - The access should include the minimum number of permissions that the work requires.
 - The access should be valid for only a limited time.
 - After the access expires, the system should lock down the VM access to prevent malicious access attempts.
- Employees need access to a remote workstation that's hosted in Azure as a VM.
The following conditions apply:
 - The employees should access the VM only during work hours.
 - The security system should consider requests to access the VM outside work hours unnecessary and malicious.
- Users would like to connect to Azure VM workloads. The system should approve connections that are only from managed and compliant devices.
- A system has experienced a tremendous number of brute-force attacks:
 - These attacks have targeted Azure VMs on RDP and SSH ports 3389 and 22.
 - The attacks have tried to guess the credentials.
 - The solution should prevent access ports such as 3389 and 22 from being exposed to the internet or on-premises environments.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

- 1. Authentication and access decisions:** The user is authenticated against Microsoft Entra ID for access to the Azure portal, Azure REST APIs, Azure PowerShell, or the Azure CLI. If authentication succeeds, a Microsoft Entra Conditional Access policy takes effect. That policy verifies whether the user meets certain criteria. Examples include using a managed device or signing in from a known location. If the user fulfills the criteria, Conditional Access grants the user access to Azure through the Azure portal or another interface.
- 2. Identity-based just-in-time access:** During authorization, Microsoft Entra PIM assigns the user a custom role of type *eligible*. The eligibility is limited to required resources and is a *time-bound* role, not a *permanent* one. Within a specified time frame, the user requests activation of this role through the Azure PIM interface. That request can trigger other actions, such as starting an approval workflow or prompting the user for multifactor authentication to verify identity. In an approval workflow, another person needs to approve the request. Otherwise the user isn't assigned the custom role and can't continue to the next step.
- 3. Network based just-in-time access:** After authentication and authorization, the custom role is temporarily linked to the user's identity. The user then requests JIT VM access. That access opens a connection from the Azure Bastion subnet on port

3389 for RDP or port 22 for SSH. The connection runs directly to the VM network interface card (NIC) or the VM NIC subnet. Azure Bastion opens an internal RDP session by using that connection. The session is limited to the Azure virtual network and isn't exposed to the public internet.

4. **Connecting to the Azure VM:** The user accesses Azure Bastion by using a temporary token. Through this service, the user establishes an indirect RDP connection to the Azure VM. The connection only works for a limited amount of time.

Components

This solution uses the following components:

- [Azure Virtual Machines](#) is an infrastructure-as-a-service (IaaS) offer. You can use Virtual Machines to deploy on-demand, scalable computing resources. In production environments that use this solution, deploy your workloads on Azure VMs. Then eliminate unnecessary exposure to your VMs and Azure assets.
- [Microsoft Entra ID](#) is a cloud-based identity service that controls access to Azure and other cloud apps.
- [PIM](#) is a Microsoft Entra service that manages, controls, and monitors access to important resources. In this solution, this service:
 - Limits permanent administrator access to standard and custom privileged roles.
 - Provides just-in-time identity-based access to custom roles.
- [JIT VM access](#) is a feature of Defender for Cloud that provides just-in-time network-based access to VMs. This feature adds a deny rule to the Azure network security group that protects the VM network interface or the subnet that contains the VM network interface. That rule minimizes the attack surface of the VM by blocking unnecessary communication to the VM. When a user requests access to the VM, the service adds a temporary allow rule to the network security group. Because the allow rule has higher priority than the deny rule, the user can connect to the VM. Azure Bastion works best for connecting to the VM. But the user can also use a direct RDP or SSH session.
- [Azure RBAC](#) is an authorization system that provides fine-grained access management of Azure resources.
- [Azure RBAC custom roles](#) provide a way to expand on Azure RBAC built-in roles. You can use them to assign permissions at levels that meet your organization's needs. These roles support PoLP. They grant only the permissions that a user needs

for the user's purpose. To access a VM in this solution, the user gets permissions for:

- Using Azure Bastion.
- Requesting JIT VM access in Defender for Cloud.
- Reading or listing VMs.
- [Microsoft Entra Conditional Access](#) is a tool that Microsoft Entra ID uses to control access to resources. Conditional Access policies support the [zero trust](#) security model. In this solution, the policies ensure that only authenticated users get access to Azure resources.
- [Azure Bastion](#) provides secure and seamless RDP and SSH connectivity to VMs in a network. In this solution, Azure Bastion connects users who use Microsoft Edge or another internet browser for HTTPS, or secured traffic on port 443. Azure Bastion sets up the RDP connection to the VM. RDP and SSH ports aren't exposed to the internet or the user's origin.

Azure Bastion is optional in this solution. Users can connect directly to Azure VMs by using the RDP protocol. If you do configure Azure Bastion in an Azure virtual network, set up a separate subnet called `AzureBastionSubnet`. Then associate a network security group with that subnet. In that group, specify a source for HTTPS traffic such as the user's on-premises IP classless inter-domain routing (CIDR) block. By using this configuration, you block connections that don't come from the user's on-premises environment.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Husam Hilal](#) | Senior Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Activate my Azure resource roles in Privileged Identity Management](#)
- [Understanding just-in-time \(JIT\) VM access](#)
- [Configure Bastion and connect to a Windows VM through a browser](#)
- [Secure user sign-in events with Microsoft Entra multifactor authentication](#)

Related resources

- [Hybrid security monitoring via Microsoft Defender for Cloud and Microsoft Sentinel](#)
- [Security considerations for highly sensitive IaaS apps in Azure](#)
- [Microsoft Entra IDaaS in Security Operations](#)

Protect backend APIs by using Azure API Management and Azure AD B2C

Azure API Management

Microsoft Entra External ID

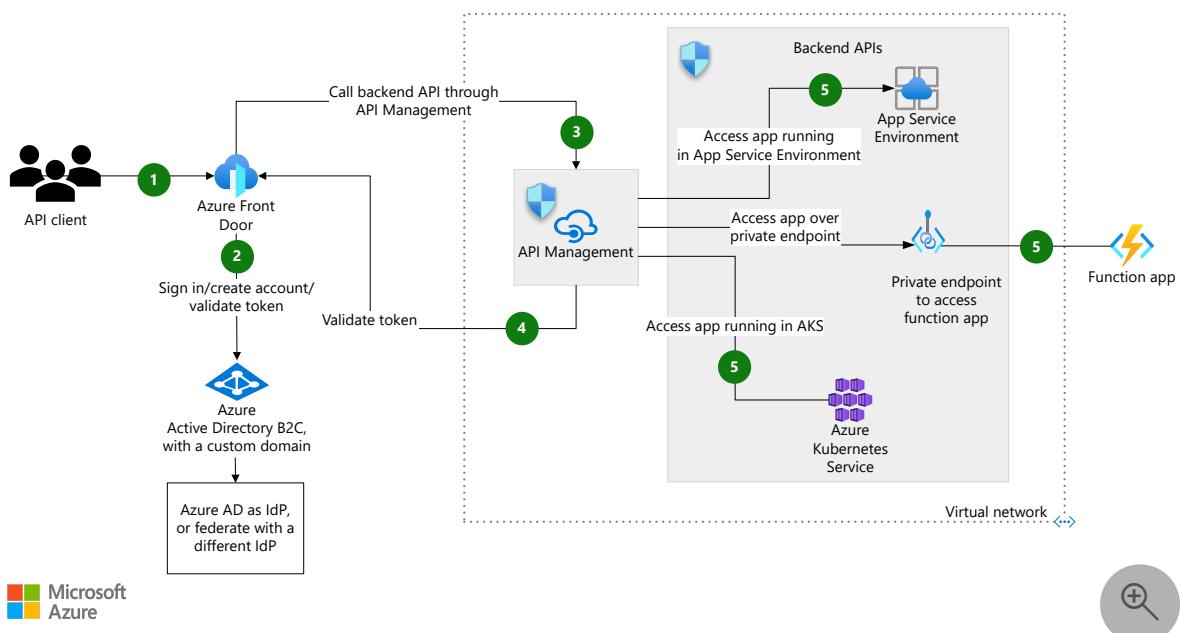
Azure Front Door

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

In this article, we'll look at an architecture that protects backend APIs in Azure and other environments by using API Management and Azure Active Directory (Azure AD) B2C to validate bearer tokens.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. To gain access to an application, an API client authenticates by providing credentials such as username and password. The IdP is Azure AD B2C in this solution, but you can use a different one.
2. The authentication request goes via Azure Front Door to Azure AD B2C, which is configured with a custom domain for sign-in. Azure AD B2C authenticates the user and returns a JSON Web Token (JWT) bearer token back to the user.
3. The client triggers an event that accesses a backend API. This event could be a click of a button on a web application or on a mobile device, or a direct call to the endpoint of the backend API.
4. The request goes through Azure Front Door, whose back end is mapped to the public endpoint of API Management. API Management intercepts the request and validates the bearer token against Azure AD B2C by using its [validate-jwt](#) policy. If the token isn't valid, API Management rejects the request by responding with a 401 code.
5. If the token is valid, API Management forwards the request to the appropriate backend API.

The diagram shows backend APIs running in three environments:

- App Service Environment
- Function Apps
- Azure Kubernetes Services (AKS)

APIs running in on-premises and hybrid cloud environments can also be integrated with API Management if network connectivity is established between the APIs and API Management.

Components

- [Azure Virtual Network](#) provides secure communications among Azure resources such as virtual machines (VMs). It also provides access to the internet, and to on-premises networks.
- [Azure Front Door](#) is a modern cloud content delivery network (CDN) service that delivers high performance, scalability, and secure user experiences for web content and applications. It offers Layer 7 capabilities such as SSL offload, path-based routing, fast failover, and caching to improve the performance and availability of your applications.
- [Azure Application Gateway](#) is a Layer 7 load balancer that manages traffic to web applications.

- [API Management](#) is a turnkey solution for publishing APIs to external and internal clients. It provides features that are useful for managing a public-facing API, including rate limiting, IP restrictions, and authentication that uses Microsoft Entra ID or another IdP. API Management doesn't perform any load balancing, so you should use it with a load balancer such as Application Gateway or a reverse proxy. For information about using API Management with Application Gateway, see [Integrate API Management in an internal virtual network with Application Gateway](#).
- [Azure AD B2C](#) is a highly available global identity management service, built on Microsoft Entra ID, for consumer-facing applications. It scales to hundreds of millions of identities. It's the IdP that this solution uses. It returns the bearer token (JWT) on successful authentication.
- [Azure App Service](#) is a fully managed service for building, deploying, and scaling web apps. You can build apps by using .NET, .NET Core, Node.js, Java, Python, or PHP. The apps can run in containers or on Windows or Linux. In a mainframe migration, the front-end screens or web interface can be coded as HTTP-based REST APIs. They can be segregated as in the mainframe application, and can be stateless to orchestrate a microservices-based system.
- [Azure App Service Environment](#) is a single-tenant deployment of App Service. It enables hosting of applications in a fully isolated and dedicated environment for securely running App Service apps at high scale.
- [Azure Kubernetes Service \(AKS\)](#) is a Microsoft managed Kubernetes environment for running containerized applications.
- [Azure Functions](#) is an event-driven serverless compute platform. Azure Functions runs on demand and at scale in the cloud.

Scenario details

Backend APIs are an entry point to your applications and data, an entry point that should be protected against malicious applications and users. In the architecture described here, Azure API Management acts as a gateway between clients and backend APIs and helps protect the APIs in many ways, including:

- Token validation
- Claims-based authorization
- SSL certificate validation
- IP restrictions
- Throttling
- Rate limiting
- Request and response validation

You can use any OpenID Connect identity provider (IdP), from Microsoft or another supplier, to authenticate clients. This solution uses Azure Active Directory B2C (Azure AD B2C). If you authenticate with something other than OpenID Connect you can, in most cases, use Azure AD B2C to federate. For more information, see [Add an identity provider to your Azure Active Directory B2C tenant](#).

Potential use cases

This architecture addresses the needs of organizations seeking to:

- Protect backend APIs from unauthorized users.
- Use API Management features such as throttling, rate limiting, and IP filtering to prevent overloading of APIs.
- Use Azure AD B2C for authentication with OpenID Connect, or federation with other IdPs, including:
 - Third party IdPs such as Ping Identity and Computer Associates (CA) SiteMinder.
 - Facebook, Microsoft account, Google, Twitter.
 - IdPs that support OAuth 1.0, OAuth 2.0, or SAML protocols.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Arshad Azeem](#) | Senior Cloud Solution Architect

Other contributors:

- [Raj Penchala](#) | Principal Cloud Security Architect
- [Ryan Hudson](#) | Principal Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Enable custom domains for Azure Active Directory B2C](#)
- [Azure Active Directory B2C: Custom CIAM User Journeys](#)
- [Resilience through developer best practices](#)

Related resources

- [Protect APIs with Application Gateway and API Management](#)
- [Automated API deployments with APIOps](#)

Map threats to your IT environment

Azure Office 365

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article explains how to diagram the essential IT environment of your organization and develop a threat map. These diagrams can help you to plan and build your defensive layer of security. Understanding your IT environment and how it's architected is essential to defining the security services that the environment requires for necessary levels of protection.

Computer systems contain information that is valuable to the organizations that produce it—and to malicious actors. A malicious actor can be an individual or a group of people who perform malicious acts against a person or organization. Their efforts can cause harm to the computers, devices, systems, and networks of companies. Their goals are to compromise or steal valuable information by using threats like malware or brute force attacks.

In this article, we look at a way to map the threats against your IT environment so that you can plan how to use Microsoft security services to implement your security strategy. This is the second article in a series of five articles that are introduced in [Use Azure monitoring to integrate security components](#).

The good news is that you don't need to create a threat map from scratch. The MITRE ATT&CK matrix is a great solution to help you develop a threat map. MITRE ATT&CK is a global knowledge database that maps threats that are based on the tactics and techniques that are observed in the real world. The MITRE Corporation catalogs every threat available and discovers many details of how those threats work and how you can defend against them. It's a public service that you can access online at [MITRE ATT&CK®](#).

This article uses a subset of those threats to present an example of how you could map threats against your IT environment.

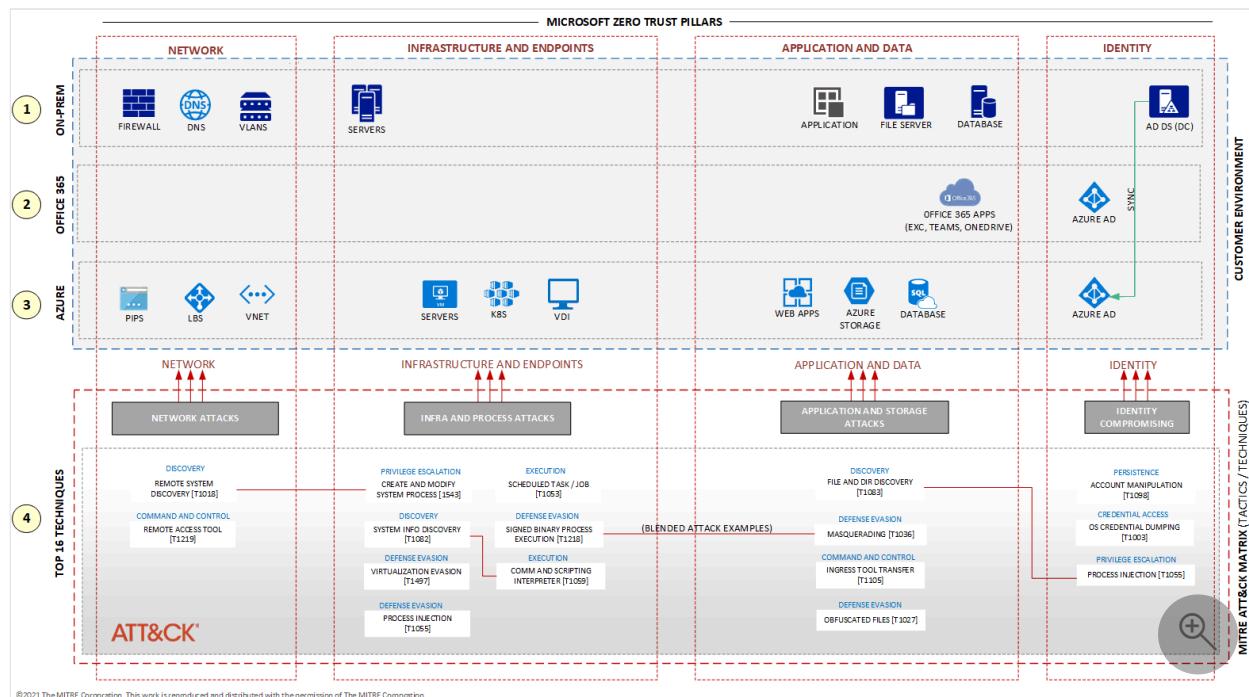
Potential use cases

Some threats are widespread regardless of the industry segment, such as ransomware, DDoS attacks, cross-site scripting, SQL injection, and so on. However, some organizations have concerns about specific types of threats that are particular to their industry or that were the basis of cyber-attacks that they've experienced. The diagram presented in this article can help you map such threats for your organization according to the area that malicious actors are likely to attack. Developing a threat map helps you to plan the layers of defense that are necessary to have a more secure environment.

You can use this diagram with different combinations of attacks to understand how to avoid and mitigate those attacks. You don't necessarily need to use the MITRE ATT&CK framework. The framework is only an example. Microsoft Sentinel, and other Microsoft security services, have worked with MITRE to provide insightful information regarding threats.

Some organizations use Cyber Kill Chain®, a methodology from Lockheed Martin, to map and understand how an attack or a series of attacks are performed against an IT environment. Cyber Kill Chain organizes threats and attacks by considering fewer tactics and techniques than the MITRE ATT&CK framework. Still, it's effective in helping you to understand threats and how they might be executed. For more information about this methodology, see [Cyber Kill Chain](#).

Architecture



Download a [Visio file](#) of this architecture.

©2021 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.

For the IT environment of organizations, we specify the components only for Azure and Microsoft 365. Your specific IT environment might include devices, appliances, and technologies from different technology providers.

For the Azure environment, the diagram shows the components that are listed in the following table.

[] [Expand table](#)

Label	Documentation
VNET	What is Azure Virtual Network ↗
LBS	What is Azure Load Balancer ?
PIPS	Public IP addresses
SERVERS	Virtual Machines ↗
K8S	Azure Kubernetes Service
VDI	What is Azure Virtual Desktop?
WEB APPS	App Service overview
AZURE STORAGE	Introduction to Azure Storage
DB	What is Azure SQL Database?
Microsoft Entra ID	What is Microsoft Entra ID?

The diagram represents Microsoft 365 through the components listed in the following table.

[] [Expand table](#)

Label	Description	Documentation
OFFICE 365	Microsoft 365 services (formerly Office 365). The applications that Microsoft 365 makes available depends on the type of license.	Microsoft 365 - Subscription for Office Apps
Microsoft Entra ID	Microsoft Entra ID, the same one utilized by Azure. Many companies use the same Microsoft Entra service for Azure and Microsoft 365.	What is Microsoft Entra ID?

Workflow

To help you understand which part of your IT environment those threats are likely to attack, the architecture diagram in this article is based on a typical IT environment for an organization that has on-premises systems, a Microsoft 365 subscription, and an Azure subscription. The resources in each of these layers are services that are common to many companies. They're classified in the diagram according to the pillars of Microsoft Zero Trust: network, infrastructure, endpoint, application, data, and identity. For more information about Zero Trust, see [Embrace proactive security with Zero Trust](#).

The architecture diagram includes the following layers:

1. On-premises

The diagram includes some essential services such as servers (VMs), network appliances, and DNS. It includes common applications that are found in most IT environments and run on virtual machines or physical servers. It also includes various types of databases, both SQL and non-SQL. Organizations usually have a file server that shares files throughout the company. Lastly, the Active Directory Domain Service, a widespread infrastructure component, handles user credentials. The diagram includes all these components in the on-premises environment.

2. Office 365 environment

This example environment contains traditional office applications, such as Word, Excel, PowerPoint, Outlook, and OneNote. Depending on the type of license, it might also include other applications, such as OneDrive, Exchange, Sharepoint, and Teams. In the diagram, these are represented by an icon for Microsoft 365 (formerly Office 365) apps and an icon for Microsoft Entra ID. Users must be authenticated to obtain access to Microsoft 365 applications, and Microsoft Entra ID acts as the identity provider. Microsoft 365 authenticates users against the same

type of Microsoft Entra ID that Azure uses. In most organizations, the [Microsoft Entra ID tenant](#) is the same for both Azure and Microsoft 365.

3. Azure environment

This layer represents Azure public cloud services, including virtual machines, virtual networks, platforms as services, web applications, databases, storage, identity services, and more. For more information about Azure, see [Azure documentation](#).

4. MITRE ATT&CK tactics and techniques

This diagram shows the top 16 threats, according to the tactics and techniques as published by The MITRE Corporation. In red lines, you can see an example of a blended attack, which means that a malicious actor might coordinate multiple attacks simultaneously.

How to use the MITRE ATT&CK framework

You can start with a simple search for the name of the threat or of the attack code on the main web page, [MITRE ATT&CK®](#).

You can also browse threats on the tactics or techniques pages:

- [Enterprise tactics](#)
- [Enterprise techniques](#)

You can still use [MITRE ATT&CK® Navigator](#), an intuitive tool provided by MITRE that helps you discover tactics, techniques, and details about threats.

Components

The example architecture in this article uses the following Azure components:

- [Microsoft Entra ID](#) is a cloud-based identity and access management service. Microsoft Entra ID helps your users to access external resources, such as Microsoft 365, the Azure portal, and thousands of other SaaS applications. It also helps them access internal resources, like apps on your corporate intranet network.
- [Azure Virtual Network](#) is the fundamental building block for your private network in Azure. Virtual Network enables many types of Azure resources to securely communicate with each other, the internet, and on-premises networks. Virtual Network provides a virtual network that benefits from Azure's infrastructure, such as scale, availability, and isolation.

- [Azure Load Balancer](#) is a high-performance, low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. It's built to handle millions of requests per second while ensuring that your solution is highly available. Azure Load Balancer is zone-redundant, ensuring high availability across Availability Zones.
- [Virtual machines](#) is one of several types of on-demand, scalable computing resources that Azure offers. An Azure virtual machine (VM) gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it.
- [Azure Kubernetes service](#) (AKS) is a fully managed Kubernetes service for deploying and managing containerized applications. AKS provides serverless Kubernetes, continuous integration/continuous delivery (CI/CD), and enterprise-grade security and governance.
- [Azure Virtual Desktop](#) is a desktop and app virtualization service that runs on the cloud to provide desktops for remote users.
- [Web Apps](#) is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, and applications run and scale with ease on both Windows and Linux-based environments.
- [Azure Storage](#) is highly available, massively scalable, durable, and secure storage for various data objects in the cloud, including object, blob, file, disk, queue, and table storage. All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- [Azure SQL database](#) is a fully managed PaaS database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring. It provides these functions without user involvement. SQL Database provides a range of built-in security and compliance features to help your application meet security and compliance requirements.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Rudnei Oliveira](#) | Senior Customer Engineer

Other contributors:

- [Gary Moore](#) | Programmer/Writer
- [Andrew Nathan](#) | Senior Customer Engineering Manager

Next steps

This document refers to some services, technologies, and terminologies. You can find more information about them in the following resources:

- [MITRE ATT&CK®](#)
- [ATT&CK® Navigator](#)
- [Public Preview: The MITRE ATT&CK Framework Blade in Microsoft Sentinel](#), a post from the Azure Cloud & AI Domain Blog
- [The Cyber Kill Chain®](#)
- [Embrace proactive security with Zero Trust](#)
- [Blended threat](#) on Wikipedia
- [How cyberattacks are changing according to new Microsoft Digital Defense Report](#) from Microsoft Security Blog

Related resources

For more details about this reference architecture, see the other articles in this series:

- [Part 1: Use Azure monitoring to integrate security components](#)
- [Part 3: Build the first layer of defense with Azure Security services](#)
- [Part 4: Build the second layer of defense with Microsoft Defender XDR Security services](#)
- [Part 5: Integrate Azure and Microsoft Defender XDR security services](#)

Build the first layer of defense with Azure Security services

Azure

Microsoft Entra ID

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

You can build an entire IT infrastructure to run your organization by using various Azure services. Azure also offers security services to protect your infrastructure. By using Azure security services, you can improve the security posture of your IT environment. You can mitigate vulnerabilities and avoid breaches by implementing a well-architected solution that follows recommendations from Microsoft.

Some security services incur fees while others have no additional charges. Free services include network security groups (NSGs), storage encryption, TLS/SSL, shared access signature tokens, and many others. This article covers such services.

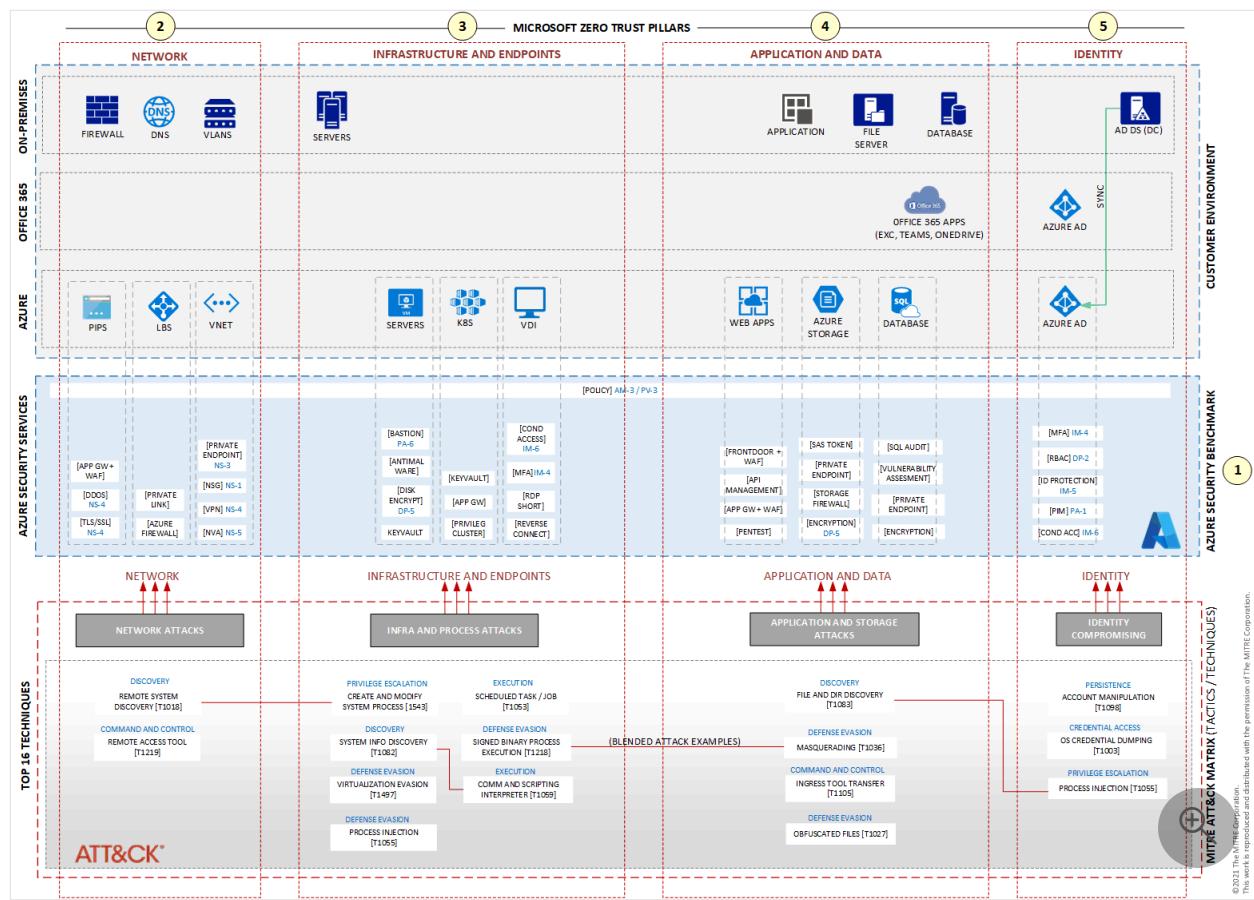
This article is the third in a series of five. To review the previous two articles in this series, including the introduction and a review of how you can map threats against an IT environment, see the following articles:

- [Use Azure monitoring to integrate security components](#)
- [Map threats to your IT environment](#)

Potential use cases

This article presents Azure security services according to each Azure service. In this way, you can think of a specific threat against resource—a virtual machine (VM), an operating system, an Azure network, an application—or an attack that might compromise users and passwords. Then use the diagram in this article to help you understand which Azure security services to use to protect resources and user identities from that type of threat.

Architecture



Download a [Visio file](#) of this architecture.

©2021 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.

The Azure security layer in this diagram is based on Azure Security Benchmark (ASB) v3, which is a set of security rules that are implemented through Azure policies. ASB is based on a combination of rules from [CIS Center for Internet Security](#) and [National Institute of Standards and Technology](#). For more information about ASB, see [Overview of the Azure Security Benchmark v3](#).

The diagram doesn't contain all the Azure security services that are available, but it shows the security services that are most commonly used by organizations. All the security services that are identified in the architectural diagram can work together in any combination according to your IT environment and your organization's security requirements.

Workflow

This section describes the components and services that appear in the diagram. Many of those are labeled with their ASB control codes, in addition to their abbreviated labels. The control codes correspond to the control domains that are listed in [Controls](#).

1. AZURE SECURITY BENCHMARK

Each security control refers to one or more specific Azure security services. The architecture reference in this article shows some of them and their control numbers according to the ASB documentation. The controls include:

- Network security
- Identity management
- Privileged access
- Data protection
- Asset management
- Logging and threat detection
- Incident response
- Posture and vulnerability management
- Endpoint security
- Backup and recovery
- DevOps security
- Governance and strategy

For more information about security controls, see [Overview of the Azure Security Benchmark \(v3\)](#).

2. NETWORK

The following table describes the network services in the diagram.

[] [Expand table](#)

Label	Description	Documentation
NSG	A free service that you attach to a network interface or subnet. An NSG allows you to filter TCP or UDP protocol traffic by using IP address ranges and ports for inbound and outbound connections.	Network security groups
VPN	A virtual private network (VPN) gateway that delivers a tunnel with IPSEC (IKE v1/v2) protection.	VPN Gateway
AZURE FIREWALL	A platform as a service (PaaS) that delivers protection in layer 4 and is attached to an entire virtual network.	What is Azure Firewall?

Label	Description	Documentation
APP GW + WAF	Azure Application Gateway with Web Application Firewall (WAF). Application Gateway is a load balancer for web traffic that works in layer 7 and adds WAF to protect applications that use HTTP and HTTPS.	What is Azure Application Gateway?
NVA	Network Virtual Appliance (NVA), a virtual security services from the marketplace that is provisioned on VMs on Azure.	Network Virtual Appliances
DDOS	DDoS protection implemented on the virtual network to help you mitigate different types of DDoS attacks.	Azure DDoS Network Protection overview
TLS/SSL	TLS/SSL deliver encryption in transit for most Azure services that exchange information, such as Azure Storage and Web Apps.	Configure end-to-end TLS by using Application Gateway with PowerShell
PRIVATE LINK	Service that allows you to create a private network for an Azure service that initially is exposed to the internet.	What is Azure Private Link?
PRIVATE ENDPOINT	Creates a network interface and attaches it to the Azure service. Private Endpoint is part of Private Link. This configuration lets the service, by using a private endpoint, be part of your virtual network.	What is a private endpoint?

3. INFRASTRUCTURE AND ENDPOINTS

The following table describes infrastructure and endpoint services that are shown in the diagram.

[+] [Expand table](#)

Label	Description	Documentation
BASTION	Bastion provides jump server functionality. This service allows you to access your VMs through remote desktop protocol (RDP) or SSH without exposing your VMs to the internet.	What is Azure Bastion?
ANTIMALWARE	Microsoft Defender provides anti-malware service and is part of Windows 10, Windows 11, Windows Server 2016, and Windows Server 2019.	Microsoft Defender Antivirus in Windows
DISK ENCRYPT	Disk Encryption allows you to encrypt the disk of a VM.	Azure Disk Encryption for Windows VMs
KEYVAULT	Key Vault, a service to store keys, secrets, and certificates with FIPS 140-2 Level 2 or 3.	Azure Key Vault basic concepts
RDP SHORT	Azure Virtual Desktop RDP Shortpath. This feature allows remote users to connect to the Virtual Desktop service from a private network.	Azure Virtual Desktop RDP Shortpath for managed networks
REVERSE CONNECT	A built-in security feature from Azure Virtual Desktop. Reverse connect guarantees that remote users receive only pixel streams and don't reach the host VMs.	Understanding Azure Virtual Desktop network connectivity

4. APPLICATION AND DATA

The following table describes application and data services that are shown in the diagram.

[] [Expand table](#)

Label	Description	Documentation
FRONDOOR + WAF	A content delivery network (CDN). Front Door combines multiple points of presence to deliver a better connection for users who access the service and adds WAF.	What is Azure Front Door?
API MANAGEMENT	A service that delivers security for API calls and manages APIs across environments.	About API Management
PENTEST	A set of best practices to execute a penetration test in your environment, including Azure resources.	Penetration testing
STORAGE SAS TOKEN	A shared access token to allow others to access your Azure storage account.	Grant limited access to Azure Storage resources using shared access signatures (SAS)
PRIVATE ENDPOINT	Create a network interface and attach it to your storage account to configure it inside a private network on Azure.	Use private endpoints for Azure Storage
STORAGE FIREWALL	Firewall that allows you to set a range of IP addresses that can access your storage account.	Configure Azure Storage firewalls and virtual networks
ENCRYPTION (Azure Storage)	Protects your storage account with encryption at rest.	Azure Storage encryption for data at rest
SQL AUDIT	Tracks database events and writes them to an audit log in your Azure storage account.	Auditing for Azure SQL Database and Azure Synapse Analytics
VULNERABILITY ASSESSMENT	Service that helps you discover, track, and remediate	SQL vulnerability assessment helps you

Label	Description	Documentation
	potential database vulnerabilities.	identify database vulnerabilities
ENCRYPTION (Azure SQL)	Transparent data encryption (TDE) helps protect Azure SQL database services by encrypting data at rest.	Transparent data encryption for SQL Database, SQL Managed Instance, and Azure Synapse Analytics

5. IDENTITY

The following table describes identity services that are shown in the diagram.

[\[+\] Expand table](#)

Label	Description	Documentation
RBAC	Azure role-based access control (Azure RBAC) helps you manage access to Azure services by using granular permissions that are based on users' Microsoft Entra credentials.	What is Azure role-based access control (Azure RBAC)?
MFA	Multifactor authentication offers additional types of authentication beyond user names and passwords.	How it works: Microsoft Entra multifactor authentication
ID PROTECTION	Identity Protection, a security service from Microsoft Entra ID, analyzes trillions of signals per day to identify and protect users from threats.	What is Identity Protection?
PIM	Privileged Identity Management (PIM), a security service from Microsoft Entra ID. It helps you to provide superuser privileges temporarily for Microsoft Entra ID (for example, Global Admin) and Azure subscriptions (for example, owner or contributor).	What is Microsoft Entra Privileged Identity Management?

Label	Description	Documentation
COND ACC	Conditional Access is an intelligent security service that uses policies that you define for various conditions to block or grant access to users.	What is Conditional Access?

Components

The example architecture in this article uses the following Azure components:

- [Microsoft Entra ID](#) is a cloud-based identity and access management service. Microsoft Entra ID helps your users to access external resources, such as Microsoft 365, the Azure portal, and thousands of other SaaS applications. It also helps them access internal resources, like apps on your corporate intranet network.
- [Azure Virtual Network](#) is the fundamental building block for your private network in Azure. Virtual Network enables many types of Azure resources to securely communicate with each other, the internet, and on-premises networks. Virtual Network provides a virtual network that benefits from Azure's infrastructure, such as scale, availability, and isolation.
- [Azure Load Balancer](#) is a high-performance, low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. It's built to handle millions of requests per second while ensuring that your solution is highly available. Azure Load Balancer is zone-redundant, ensuring high availability across Availability Zones.
- [Virtual machines](#) is one of several types of on-demand, scalable computing resources that Azure offers. An Azure virtual machine (VM) gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it.
- [Azure Kubernetes service](#) (AKS) is a fully managed Kubernetes service for deploying and managing containerized applications. AKS provides serverless Kubernetes, continuous integration/continuous delivery (CI/CD), and enterprise-grade security and governance.
- [Azure Virtual Desktop](#) is a desktop and app virtualization service that runs on the cloud to provide desktops for remote users.

- [Web Apps](#) is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, and applications run and scale with ease on both Windows and Linux-based environments.
- [Azure Storage](#) is highly available, massively scalable, durable, and secure storage for various data objects in the cloud, including object, blob, file, disk, queue, and table storage. All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- [Azure SQL database](#) is a fully managed PaaS database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring. It provides these functions without user involvement. SQL Database provides a range of built-in security and compliance features to help your application meet security and compliance requirements.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Rudnei Oliveira](#) | Senior Customer Engineer

Other contributors:

- [Gary Moore](#) | Programmer/Writer
- [Andrew Nathan](#) | Senior Customer Engineering Manager

Next steps

Microsoft has more documentation that can help you secure your IT environment, and the following articles can be particularly helpful:

- [Security in the Microsoft Cloud Adoption Framework for Azure](#). The Cloud Adoption Framework provides security guidance for your cloud journey by clarifying the processes, best practices, models, and experience.
- [Microsoft Azure Well-Architected Framework](#). The Azure Well-Architected Framework is a set of guiding tenets that you can use to improve the quality of a workload. The framework is based on five pillars: reliability, security, cost optimization, operational excellence, and performance efficiency.

- [Microsoft Security Best Practices](#). Microsoft Security Best Practices (formerly known as the *Azure Security Compass* or *Microsoft Security Compass*) is a collection of best practices that provide clear, actionable guidance for security-related decisions.
- [Microsoft Cybersecurity Reference Architectures](#) (MCRA). MCRA is a compilation of various Microsoft security reference architectures.

In the following resources, you can find more information about the services, technologies, and terminologies that are mentioned in this article:

- [What are public, private, and hybrid clouds?](#)
- [Overview of the Azure Security Benchmark \(v3\)](#)
- [Embrace proactive security with Zero Trust](#)
- [Microsoft 365 subscription information](#)
- [Microsoft Defender XDR](#)

Related resources

For more details about this reference architecture, see the other articles in this series:

- [Part 1: Use Azure monitoring to integrate security components](#)
- [Part 2: Map threats to your IT environment](#)
- [Part 4: Build the second layer of defense with Microsoft Defender XDR Security services](#)
- [Part 5: Integration between Azure and Microsoft Defender XDR security services](#)

Build the second layer of defense with Microsoft Defender XDR Security services

Microsoft Defender for Office 365

Microsoft Defender for Cloud Apps

Microsoft Defender for Identity

Microsoft 365

Microsoft Endpoint Manager

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

It's common for organizations to use a hybrid environment, with resources running both on Azure and on-premises. Most Azure resources, such as virtual machines (VMs), Azure applications, and Microsoft Entra ID, can be protected by security services that run on Azure.

Organizations often also subscribe to Microsoft 365 to provide users with applications like Word, Excel, PowerPoint, and Exchange online. Microsoft 365 also offers security services that you can use to build an additional layer of security for some of the most used Azure resources.

To consider using security services from Microsoft 365, it's helpful to know some terminology and understand the structure of Microsoft 365 services. This fourth article in a series of five can help with that. This article builds on topics that are covered in the previous articles, particularly:

- [Map threats to your IT environment](#)
- [Build the first layer of defense with Azure Security services](#)

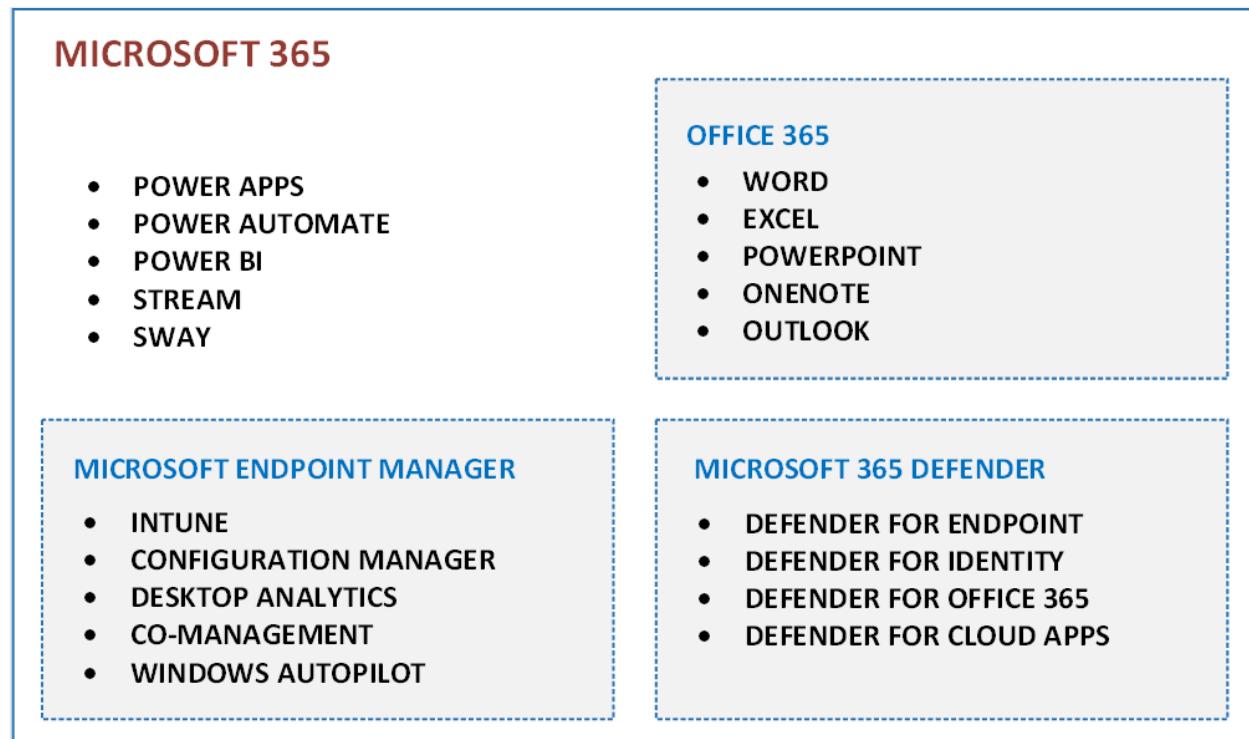
Microsoft 365 and Office 365 are cloud-based services that are designed to help you meet your organization's needs for robust security, reliability, and user productivity.

Microsoft 365 includes services like Power Automate, Forms, Stream, Sway, and Office 365. Office 365 includes the well-known suite of productivity applications. For more information about subscription options for these two services, see [Microsoft 365 and Office 365 plan options](#).

Depending on the license that you acquire for Microsoft 365, you can also get the security services for Microsoft 365. These security services are called Microsoft Defender XDR, which provides multiple services:

- Microsoft Defender for Endpoint
- Microsoft Defender for Identity
- Microsoft Defender for Office 365
- Microsoft Defender for Cloud Apps

The following diagram illustrates the relationship of solutions and main services that Microsoft 365 offers, though not all services are listed.



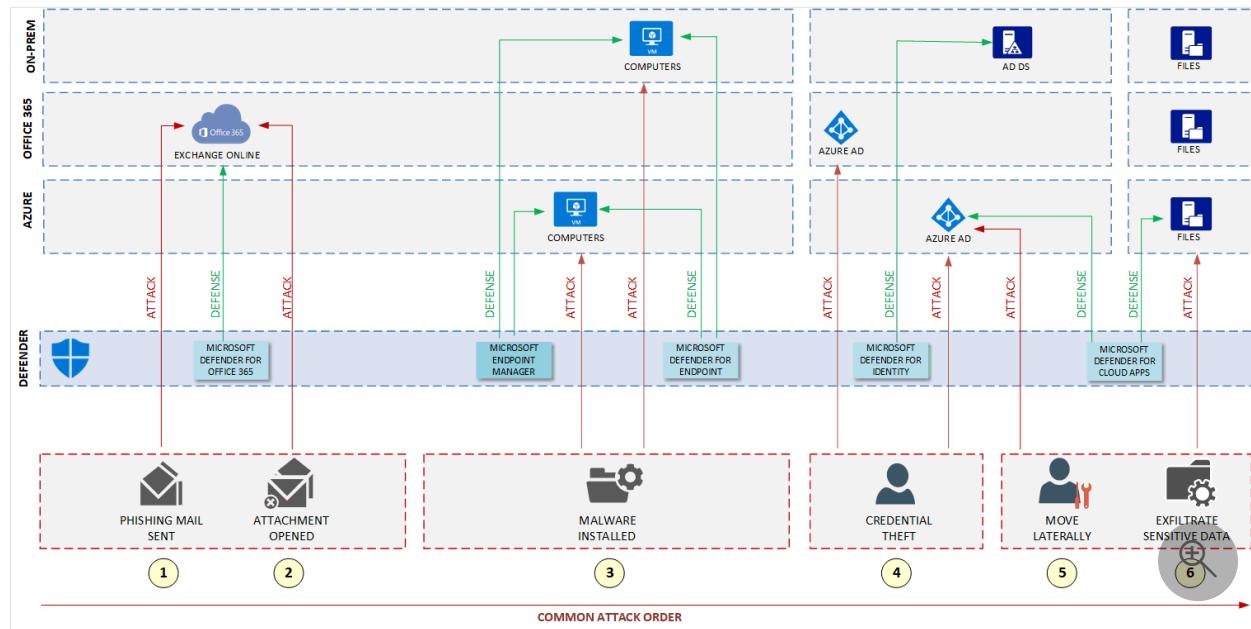
Potential use case

People are sometimes confused about Microsoft 365 security services and their role in IT cybersecurity. The main causes are names that are similar to each other, including some security services that run on Azure, such as Microsoft Defender for Cloud (formerly known as Azure Security Center) and Defender for Cloud Apps (formerly known as Microsoft Cloud Application Security).

But the confusion isn't only about terminology. Some services deliver similar protection but for different resources, such as Defender for Identity and Azure Identity Protection. Both services offer protection for identity services, but Defender for Identity protects identity on-premises (through Active Directory Domain Services, based on Kerberos authentication) while Azure Identity Protection protects identity in the cloud (through Microsoft Entra ID, based on OAuth authentication).

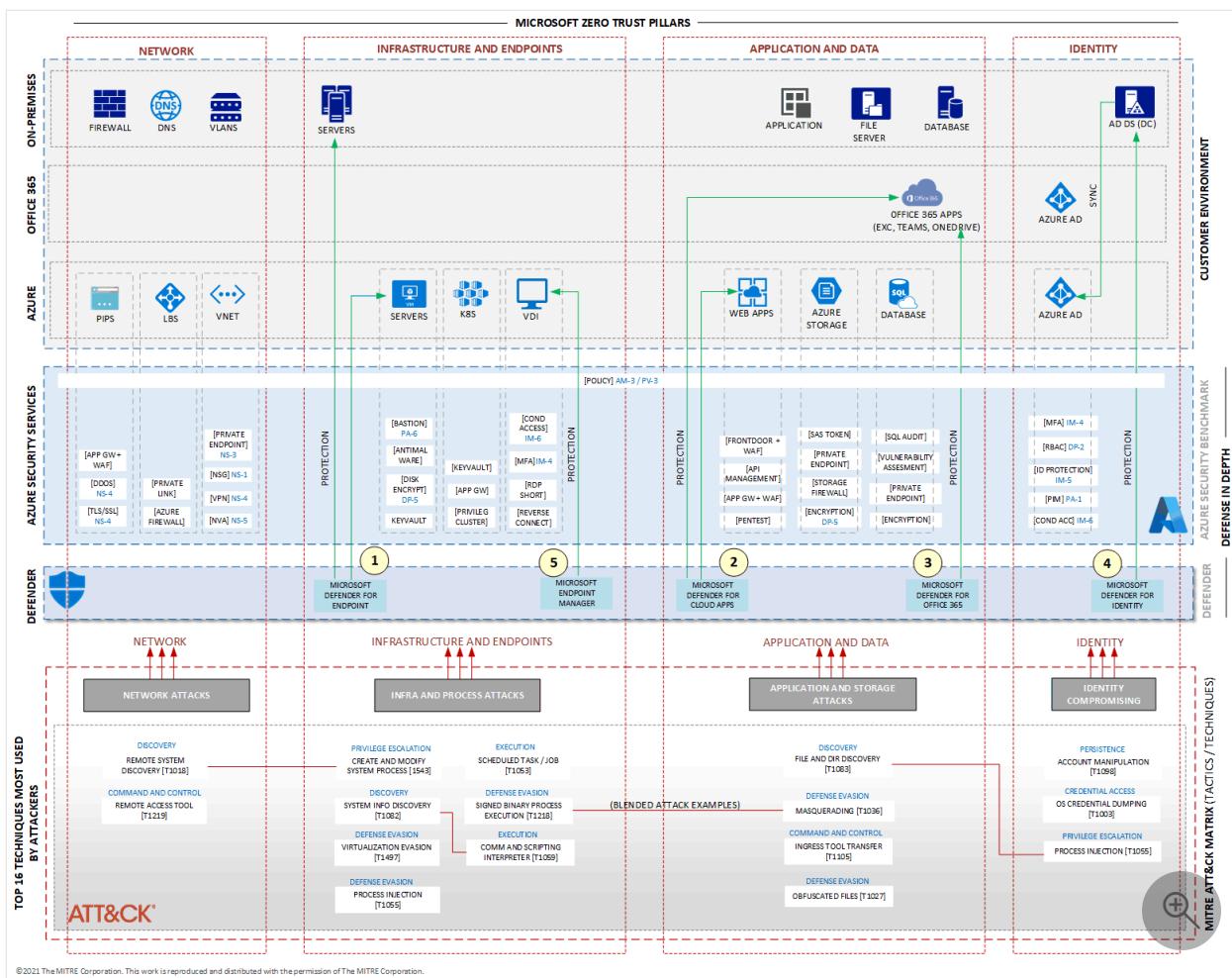
These examples show that if you understand how Microsoft 365 security services work and the differences compared to Azure security services, you're able to plan your strategy for security in the Microsoft cloud in an effective way and still provide a great security posture for your IT environment. That is the purpose of this article.

The following diagram illustrates a real use case in which you might consider using Microsoft Defender XDR security services. The diagram shows the resources that need to be protected. The services that run in the environment are shown on top. Some potential threats are shown at the bottom. Microsoft Defender XDR services are in the middle, defending the organization's resources from potential threats.



Architecture

The following diagram shows a layer, labeled as **DEFENDER**, that represents the Microsoft Defender XDR security services. Adding these services to your IT environment helps you to build better defense for your environment. The services in the Defender layer can work with Azure security services.



Download a [Visio file](#) of this architecture.

©2021 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.

Workflow

1. Microsoft Defender for Endpoint

Defender for Endpoint secures endpoints in your enterprise and is designed to help networks prevent, detect, investigate, and respond to advanced threats. It creates a layer of protection for VMs that run on Azure and on-premises. For more information about what it can protect, see [Microsoft Defender for Endpoint](#).

2. Microsoft Defender for Cloud Apps

Formerly known as Microsoft Cloud Application Security, Defender for Cloud Apps is a cloud access security broker (CASB) that supports multiple deployment modes. Those modes include log collection, API connectors, and reverse proxy. It provides rich visibility, control over data travel, and sophisticated analytics to identify and combat cyberthreats across all your Microsoft and third-party cloud services. It provides protection and risk mitigation for Cloud Apps and even for some apps

that run on-premises. It also provides a protection layer for users who access those apps. For more information, see [Microsoft Defender for Cloud Apps overview](#).

It's important to not confuse Defender for Cloud Apps with Microsoft Defender for Cloud, which provides recommendations and a score of the security posture of servers, apps, storage accounts, and other resources running in Azure, on-premises, and in other clouds. Defender for Cloud consolidates two previous services, Azure Security Center and Azure Defender.

3. Microsoft Defender for Office 365

Defender for Office 365 safeguards your organization against malicious threats that are posed by email messages, links (URLs), and collaboration tools. It provides protection for email and collaboration. Depending on the license, you're able to add post-breach investigation, hunting, and response, as well as automation and simulation (for training). For more information about licensing options, see [Microsoft Defender for Office 365 security overview](#).

4. Microsoft Defender for Identity

Defender for Identity is a cloud-based security solution that uses your on-premises Active Directory signals to identify, detect, and investigate advanced threats, compromised identities, and malicious insider actions that are directed at your organization. It protects Active Directory Domain Services (AD DS) that run on-premises. Even though this service runs on the cloud, it works to protect identities on-premises. Defender for Identity was formerly named Azure Advanced Threat Protection. For more information, see [What is Microsoft Defender for Identity?](#)

If you need protection for identities that are provided by Microsoft Entra ID and that runs natively on the cloud, consider Microsoft Entra ID Protection.

5. Microsoft Endpoint Manager

Endpoint Manager provides services for cloud services, on-premises services, and for Microsoft Intune, which allows you to control features and settings on Android, Android Enterprise, iOS, iPadOS, macOS, Windows 10, and Windows 11 devices. It integrates with other services, including:

- Microsoft Entra ID.
- Mobile threat defenders.
- Administrative (ADMX) templates.
- Win32 apps.
- Custom line-of-business apps.

Another service that is now part of Endpoint Manager is Configuration Manager, an on-premises management solution that allows you to manage client and server computers that are on your network, connected directly or via the internet. You can enable cloud functionality to integrate Configuration Manager with Intune, Microsoft Entra ID, Defender for Endpoint, and other cloud services. Use it to deploy apps, software updates, and operating systems. You can also monitor compliance, query for objects, act on clients in real time, and much more. To learn about all the services that are available, see [Microsoft Endpoint Manager overview](#).

Attack order of example threats

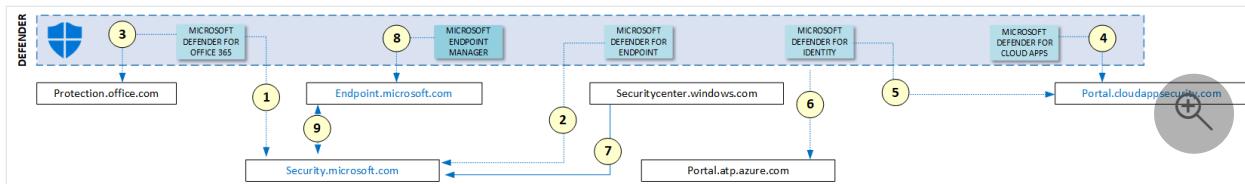
The threats named in the diagram follow a common attack order:

1. An attacker sends a phishing email with malware attached to it.
2. An end user opens the attached malware.
3. The malware installs in the back end without the user noticing.
4. The installed malware steals some users' credentials.
5. The attacker uses the credentials to gain access to sensitive accounts.
6. If the credentials provide access to an account that has elevated privilege, the attacker compromises additional systems.

The diagram also shows in the layer labeled as **DEFENDER** which Microsoft Defender XDR services can monitor and mitigate those attacks. This is an example of how Defender provides an additional layer of security that works with Azure security services to offer additional protection of the resources that are shown in the diagram. For more information about how potential attacks threaten your IT environment, see the second article in this series, [Map threats to your IT environment](#). For more information about Microsoft Defender XDR, see [Microsoft Defender XDR](#).

Access and manage Microsoft Defender XDR Security services

Currently, you might need to use multiple portals to manage Microsoft Defender XDR services. However, Microsoft is working to centralize functionality as much as possible. The following diagram shows which portals are currently available and their relationships with each other.



Security.microsoft.com is currently the most important portal available because it brings functionalities from Microsoft Defender for Office 365 (1) and from Defender for Endpoint (2). However, as of March 2022, you can still access *protection.office.com* for security functionalities regarding Office 365 (3). For Defender for Endpoint, if you try to access the old portal, *securitycenter.windows.com*, you're redirected to the new portal at *security.microsoft.com* (7).

The primary use of *Portal.cloudappsecurity.com* is to manage (4) Defender for Cloud Apps. It allows you to manage cloud apps and some apps that run on-premises, manage unauthorized apps (shadow IT), and review user signals from Identity Protection. You can also use this portal to manage many signals and features from (5) Identity protection on-premises, which allows you to consolidate many functions from (6) *portal.atp.azure.com* on (4) the portal for Defender for Cloud Apps. However, you can still access (6) *portal.atp.azure.com* if you need it.

Lastly, *endpoint.microsoft.com* provides functionality mainly for Intune and Configuration Manager, but also for other services that are part of Endpoint Manager. Because *security.microsoft.com* and *endpoint.microsoft.com* deliver security protection for endpoints, they have many interactions between them (9) to offer a great security posture for your endpoints.

Components

The example architecture in this article uses the following Azure components:

- [Microsoft Entra ID](#) is a cloud-based identity and access management service. Microsoft Entra ID helps your users to access external resources, such as Microsoft 365, the Azure portal, and thousands of other SaaS applications. It also helps them access internal resources, like apps on your corporate intranet network.
- [Azure Virtual Network](#) is the fundamental building block for your private network in Azure. Virtual Network enables many types of Azure resources to securely communicate with each other, the internet, and on-premises networks. Virtual Network provides a virtual network that benefits from Azure's infrastructure, such as scale, availability, and isolation.
- [Azure Load Balancer](#) is a high-performance, low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. It's built to handle

millions of requests per second while ensuring that your solution is highly available. Azure Load Balancer is zone-redundant, ensuring high availability across Availability Zones.

- [Virtual machines](#) is one of several types of on-demand, scalable computing resources that Azure offers. An Azure virtual machine (VM) gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it.
- [Azure Kubernetes service](#) (AKS) is a fully managed Kubernetes service for deploying and managing containerized applications. AKS provides serverless Kubernetes, continuous integration/continuous delivery (CI/CD), and enterprise-grade security and governance.
- [Azure Virtual Desktop](#) is a desktop and app virtualization service that runs on the cloud to provide desktops for remote users.
- [Web Apps](#) is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, and applications run and scale with ease on both Windows and Linux-based environments.
- [Azure Storage](#) is highly available, massively scalable, durable, and secure storage for various data objects in the cloud, including object, blob, file, disk, queue, and table storage. All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- [Azure SQL database](#) is a fully managed PaaS database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring. It provides these functions without user involvement. SQL Database provides a range of built-in security and compliance features to help your application meet security and compliance requirements.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Rudnei Oliveira](#) | Senior Customer Engineer

Other contributors:

- [Gary Moore](#) | Programmer/Writer
- [Andrew Nathan](#) | Senior Customer Engineering Manager

Next steps

- [Defend against threats with Microsoft 365](#)
- [Detect and respond to cyber attacks with Microsoft Defender XDR](#)
- [Get started with Microsoft Defender XDR](#)
- [Implement threat intelligence in Microsoft 365](#)
- [Manage security with Microsoft 365](#)
- [Protect against malicious threats with Microsoft Defender for Office 365](#)
- [Protect on-premises identities with Microsoft Defender for Cloud for Identity](#)

Related resources

For more details about this reference architecture, see the other articles in this series:

- [Part 1: Use Azure monitoring to integrate security components](#)
- [Part 2: Map threats to your IT environment](#)
- [Part 3: Build the first layer of defense with Azure Security services](#)
- [Part 5: Integration between Azure and Microsoft Defender XDR security services](#)

Integrate Azure and Microsoft Defender XDR security services

Microsoft Sentinel

Azure Monitor

Microsoft Defender for Cloud

Azure Log Analytics

Azure Network Watcher

💡 Solution ideas

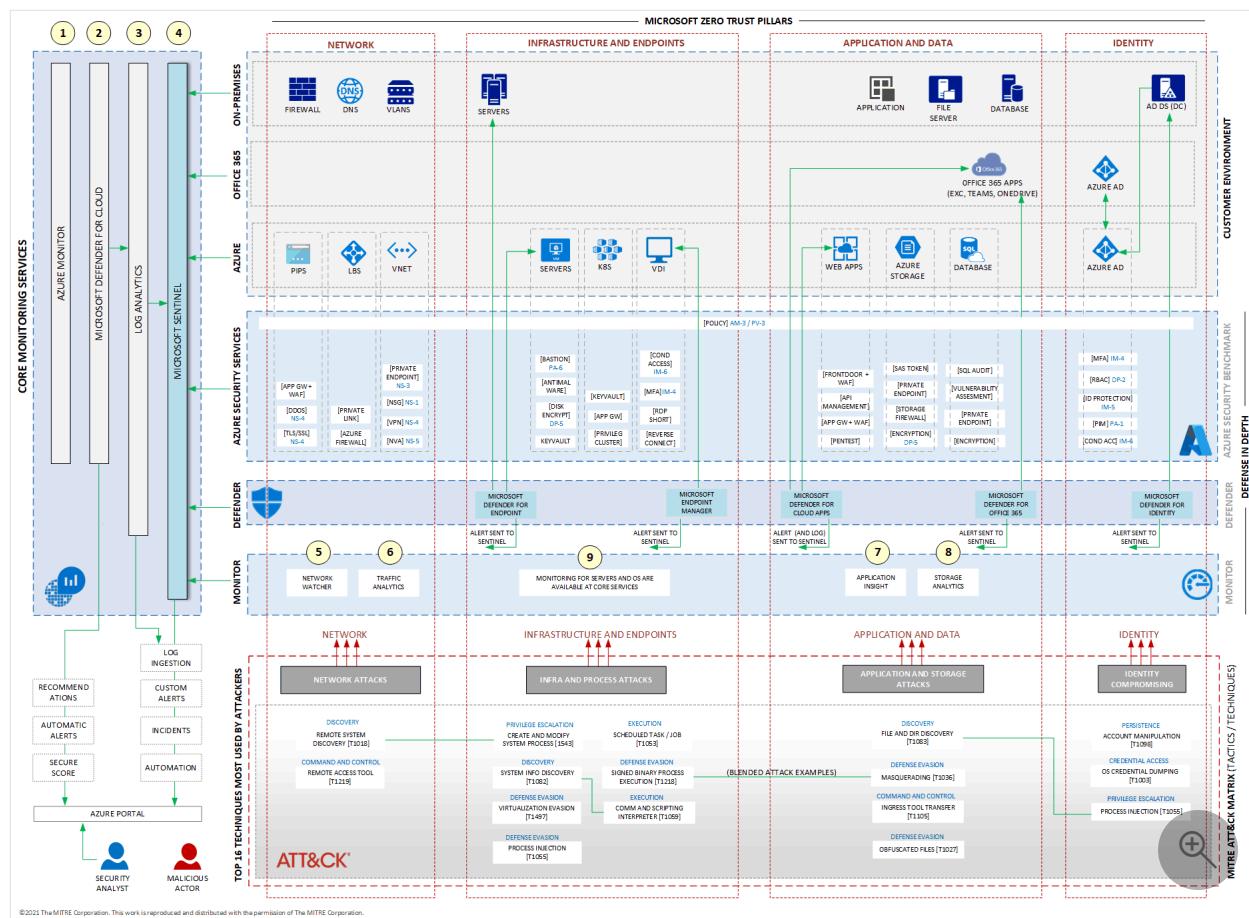
This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

You can enhance the security posture of your organization's IT environment by using the security features of both Microsoft 365 and Azure. This article, the fifth in a series of five, describes how you can integrate the security features of these services by using Microsoft Defender XDR and Azure monitoring services.

This article builds on the previous articles in the series:

1. [Use Azure monitoring to integrate security components](#) provides an overall view of how you can integrate the security services of Azure and Microsoft Defender XDR.
2. [Map threats to your IT environment](#) describes methods to map examples of common threats, tactics, and techniques against an example of a hybrid IT environment that uses both on-premises and Microsoft cloud services.
3. [Build the first layer of defense with Azure Security services](#) maps an example of some Azure security services that create the first layer of defense to protect your Azure environment according to Azure Security Benchmark version 3.
4. [Build the second layer of defense with Microsoft Defender XDR Security services](#) describes an example of a series of attacks against your IT environment and how to add another layer of protection by using Microsoft Defender XDR.

Architecture



Download a [Visio file](#) of this architecture.

©2021 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.

This diagram shows a complete architecture reference. It includes an example of an IT environment, a set of example threats that are described according to their tactics (in blue), and their techniques (in the text box) according to the MITRE ATT&CK matrix. The MITRE ATT&CK matrix is covered in [Map threats to your IT environment](#).

There are important services that are presented in the diagram. Some of those services, such as Network Watcher and Application Insights, are focused on capturing information from specific services. Some of them, like Log Analytics (also known as Azure Monitor Logs) and Microsoft Sentinel, are core services because they can collect, store, and analyze information from various services, regardless of whether they're network, compute, or applications services.

The central part of the diagram has two layers of security services. There's also one layer with specific Azure monitoring services that are integrated through Azure Monitor (on the left side of the diagram). The key component of this integration is Microsoft Sentinel.

The diagram shows the following services in **Core Monitoring Services** and in the **Monitor** layer:

- Azure Monitor
- Log Analytics
- Microsoft Defender for Cloud
- Microsoft Sentinel
- Network Watcher
- Traffic Analytics (part of Network Watcher)
- Application Insights
- Storage Analytics

Workflow

1. **Azure Monitor** is the umbrella for many Azure monitoring services. It includes log management, metrics, and Application Insights, among others. It also provides a collection of dashboards that are ready for use and management of alerts. For more information, see [Azure Monitor overview](#).
2. **Microsoft Defender for Cloud** delivers recommendations for virtual machines (VMs), storage, applications, and other resources, that help an IT environment to be compliant with various regulatory standards, such as ISO and PCI. At the same time, Defender for Cloud offers a score for the security posture of systems that can help you track the security of your environment. Defender for Cloud also offers automatic alerts that are based on the logs that it collects and analyzes. Defender for Cloud was formerly known as Azure Security Center. For more information, see [Microsoft Defender for Cloud](#).
3. **Log Analytics** is one of the most important services. It's responsible for storing all the logs and alerts that are used to create alerts, insights, and incidents. Microsoft Sentinel works on top of Log Analytics. Basically, all data that Log Analytics ingests is available automatically to Microsoft Sentinel. Log Analytics is also known as Azure Monitor Logs. For more information, see [Overview of Log Analytics in Azure Monitor](#).
4. **Microsoft Sentinel** works like a façade for Log Analytics. While Log Analytics stores logs and alerts from various sources, Microsoft Sentinel offers APIs that help with ingestion of logs from various sources. Those sources include on-premises VMs, Azure VMs, alerts from Microsoft Defender XDR and other services. Microsoft Sentinel correlates the logs to provide insights about what is going on in your IT environment, avoiding false positives. Microsoft Sentinel is the core of security and monitoring for Microsoft cloud services. For more information about Microsoft Sentinel, see [What is Microsoft Sentinel?](#).

The preceding services in this list are core services that work throughout Azure, Office 365, and on-premises environments. The following services focus on specific resources:

5. **Network Watcher** provides tools to monitor, diagnose, view metrics, and enable or disable logs for resources in an Azure virtual network. For more information, see [What is Azure Network Watcher?](#)
6. **Traffic Analytics** is part of Network Watcher and works on top of logs from network security groups (NSGs). Traffic Analytics offers many dashboards that are capable of aggregating metrics from outbound and inbound connection in Azure Virtual Network. For more information, see [Traffic Analytics](#).
7. **Application Insights** focuses on applications and provides extensible performance management and monitoring for live web apps, including support for a wide variety of platform such as .NET, Node.js, Java, and Python. Application Insights is a feature of Azure Monitor. For more information, see [Application Insights overview](#).
8. **Azure Storage Analytics** performs logging and provides metrics for a storage account. You can use its data to trace requests, analyze usage trends, and diagnose issues with your storage account. For more information, see [Use Azure Storage analytics to collect logs and metrics data](#).
9. Because this architecture reference is based on [Microsoft Zero Trust](#), the services and components under **Infrastructure and Endpoint** don't have specific monitoring services. Azure Monitor logs and Defender for Cloud are the main services that collect, store, and analyze logs from VMs and others compute services.

The key component in this architecture is Microsoft Sentinel, because it connects all the logs and alerts that are provided by Azure security services, Microsoft Defender XDR, and Azure Monitor. After you implement Microsoft Sentinel and it's receiving logs and alerts from all the sources that are identified in this article, the next step is to map a set of queries of those logs to obtain insights and evidence of indicators of compromise (IOCs). When information is captured by Microsoft Sentinel, you can investigate it or allow an automated response that you configure to mitigate or resolve the incident. Automatic responses include actions like blocking a user on in Microsoft Entra ID or blocking an IP address via the firewall.

For more information about Microsoft Sentinel, see [Microsoft Sentinel documentation](#).

How to access security and monitoring services

The following list provides information about how to access each of the services that are presented in this article:

- **Azure security services.** You can access all the Azure security services that are mentioned in the diagrams in this series of articles by using [Azure portal](#). In the portal, use the search function to locate the services that you're interested in and access them.
- **Azure Monitor.** Azure Monitor is available in all Azure subscriptions. You can access it from a search for *monitor* in the [Azure portal](#).
- **Defender for Cloud.** Defender for Cloud is available to anyone who accesses the [Azure portal](#). In the portal, search for *Defender for Cloud*.
- **Log Analytics.** To access Log Analytics, you must first create the service in the portal, because it doesn't exist by default. In the [Azure portal](#), search for *Log Analytics workspace*, and then select **Create**. After creation, you're able to access the service.
- **Microsoft Sentinel.** Because Microsoft Sentinel works on top of Log Analytics, you must first create a Log Analytics workspace. Next, search for *sentinel* in the [Azure portal](#). Then create the service by choosing the workspace that you want to have behind Microsoft Sentinel.
- **Microsoft Defender for Endpoint.** Defender for Endpoint is part of Microsoft Defender XDR. Access the service through <https://security.microsoft.com>. This is a change from the previous URL, *securitycenter.windows.com*.
- **Microsoft Defender for Cloud Apps.** Defender for Cloud Apps is part of Microsoft 365. Access the service through <https://portal.cloudappsecurity.com>.
- **Microsoft Defender for Office 365.** Defender for Office 365 is part of Microsoft 365. Access the service through <https://security.microsoft.com>, the same portal used for Defender for Endpoint. (This is a change from the previous URL, *protection.office.com*.)
- **Microsoft Defender for Identity.** Defender for Identity is part of Microsoft 365. You access the service through <https://portal.atp.azure.com>. Although it's a cloud service, Defender for Identity is responsible for also protecting identity on on-premises systems.
- **Microsoft Endpoint Manager.** Endpoint Manager is the new name for Intune, Configuration Manager, and other services. Access it through <https://endpoint.microsoft.com>. To learn more about accessing the services that

are provided by Microsoft Defender XDR and how each portal is related, see [Build the second layer of defense with Microsoft Defender XDR Security services](#).

- **Azure Network Watcher.** To access Azure Network Watcher, search for *watcher* in the [Azure portal](#).
- **Traffic Analytics.** Traffic Analytics is part of Network Watcher. You can access it from the menu on the left side in Network Watcher. It's a powerful network monitor that works based on your NSGs that are implemented on your individual network interfaces and subnets. Network Watcher requires collection of information from the NSGs. For instructions on how to collect that information, see [Tutorial: Log network traffic to and from a virtual machine using the Azure portal](#).
- **Application Insight.** Application Insight is part of Azure Monitor. However, you must first create it for the application that you want to monitor. For some applications built on Azure, such as Web Apps, you can create Application Insight directly from the provisioning of Web Apps. To access it, search for *monitor* in the [Azure portal](#). In the **Monitor** page, select **Applications** in the menu on the left side.
- **Storage Analytics.** Azure Storage offers various types of storage under the same storage account technology. You can find blobs, files, tables, and queues on top of storage accounts. Storage analytics offers a broad range of metrics to use with those storage services. Access Storage Analytics from your Storage account in the [Azure portal](#), then select **Diagnostic settings** in the menu on the left side. Choose one log analytics workspace to send that information. Then you can access a dashboard from **Insights**. Everything in your storage account that's being monitored is represented in the menu.

Components

The example architecture in this article uses the following Azure components:

- [Microsoft Entra ID](#) is a cloud-based identity and access management service. Microsoft Entra ID helps your users to access external resources, such as Microsoft 365, the Azure portal, and thousands of other SaaS applications. It also helps them access internal resources, like apps on your corporate intranet network.
- [Azure Virtual Network](#) is the fundamental building block for your private network in Azure. Virtual Network enables many types of Azure resources to securely communicate with each other, the internet, and on-premises networks. Virtual Network provides a virtual network that benefits from Azure's infrastructure, such as scale, availability, and isolation.

- [Azure Load Balancer](#) is a high-performance, low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. It's built to handle millions of requests per second while ensuring that your solution is highly available. Azure Load Balancer is zone-redundant, ensuring high availability across Availability Zones.
- [Virtual machines](#) is one of several types of on-demand, scalable computing resources that Azure offers. An Azure virtual machine (VM) gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it.
- [Azure Kubernetes service](#) (AKS) is a fully managed Kubernetes service for deploying and managing containerized applications. AKS provides serverless Kubernetes, continuous integration/continuous delivery (CI/CD), and enterprise-grade security and governance.
- [Azure Virtual Desktop](#) is a desktop and app virtualization service that runs on the cloud to provide desktops for remote users.
- [Web Apps](#) is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, and applications run and scale with ease on both Windows and Linux-based environments.
- [Azure Storage](#) is highly available, massively scalable, durable, and secure storage for various data objects in the cloud, including object, blob, file, disk, queue, and table storage. All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- [Azure SQL database](#) is a fully managed PaaS database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring. It provides these functions without user involvement. SQL Database provides a range of built-in security and compliance features to help your application meet security and compliance requirements.

Solution details

Monitoring solutions on Azure might seem confusing at first, because Azure offers multiple monitoring services. However, each Azure monitoring service is important in the security and monitoring strategy that's described in this series. The articles in this series describe the various services and how to plan effective security for your IT environment.

1. Use Azure monitoring to integrate security components
2. Map threats to your IT environment
3. Build the first layer of defense with Azure Security services
4. Build the second layer of defense with Microsoft Defender XDR Security services

Potential use cases

This reference architecture can help you understand the whole picture of Microsoft Cloud security services and to how to integrate them for the best security posture.

You don't need to implement all the security services that are presented in this architecture. However, this example and the threat map that's represented in the architecture diagram can help you to understand how to create your own map and then plan accordingly for your security strategy. Select the right Azure security services and the Microsoft Defender XDR services that you want to integrate through Azure so that your IT environment has the security that it needs.

Cost optimization

Pricing for the Azure services that are presented in this series of articles is calculated in various ways. Some services are free of charge, some have a charge for each use, and some have a charge that is based on licensing. The best way to estimate the pricing for any of the Azure security services is to use the [Pricing calculator](#). In the calculator, search for a service that you're interested in, and then select it to get all the variables that determine the price for the service.

Microsoft Defender XDR security services work with licenses. For information about the licensing requirements, see [Microsoft Defender XDR prerequisites](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Rudnei Oliveira](#) | Senior Customer Engineer

Other contributors:

- [Gary Moore](#) | Programmer/Writer
- [Andrew Nathan](#) | Senior Customer Engineering Manager

Next steps

- [Defend against threats with Microsoft 365](#)
- [Detect and respond to cyber attacks with Microsoft Defender XDR](#)
- [Get started with Microsoft Defender XDR](#)
- [Manage security with Microsoft 365](#)
- [Protect against malicious threats with Microsoft Defender for Office 365](#)
- [Protect on-premises identities with Microsoft Defender for Cloud for Identity](#)

Related resources

For more details about this reference architecture, see the other articles in this series:

- [Part 1: Use Azure monitoring to integrate security components](#)
- [Part 2: Map threats to your IT environment](#)
- [Part 3: Build the first layer of defense with Azure Security services](#)
- [Part 4: Build the second layer of defense with Microsoft Defender XDR Security services](#)

For related architectures on Azure Architecture Center, see the following articles:

- [Implement a secure hybrid network](#)
- [Monitor hybrid security using Microsoft Defender for Cloud and Microsoft Sentinel](#)

Microsoft Sentinel automated responses

Microsoft Sentinel Microsoft Entra ID Azure Logic Apps

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Microsoft Sentinel is a scalable cloud solution for security information and event management (SIEM), and for security orchestration, automation, and response (SOAR). It delivers intelligent security analytics for enterprises of all sizes, and provides the following capabilities:

- Business attack detection
- Proactive hunting
- Threat response

Threat response is provided by Microsoft Sentinel playbooks. When a playbook is triggered by a Microsoft Sentinel alert or incident, the playbook runs a series of actions to counter the threat. The playbooks are built by using Azure Logic Apps.

Microsoft Sentinel includes many ready-to-use playbooks, including playbooks for these uses:

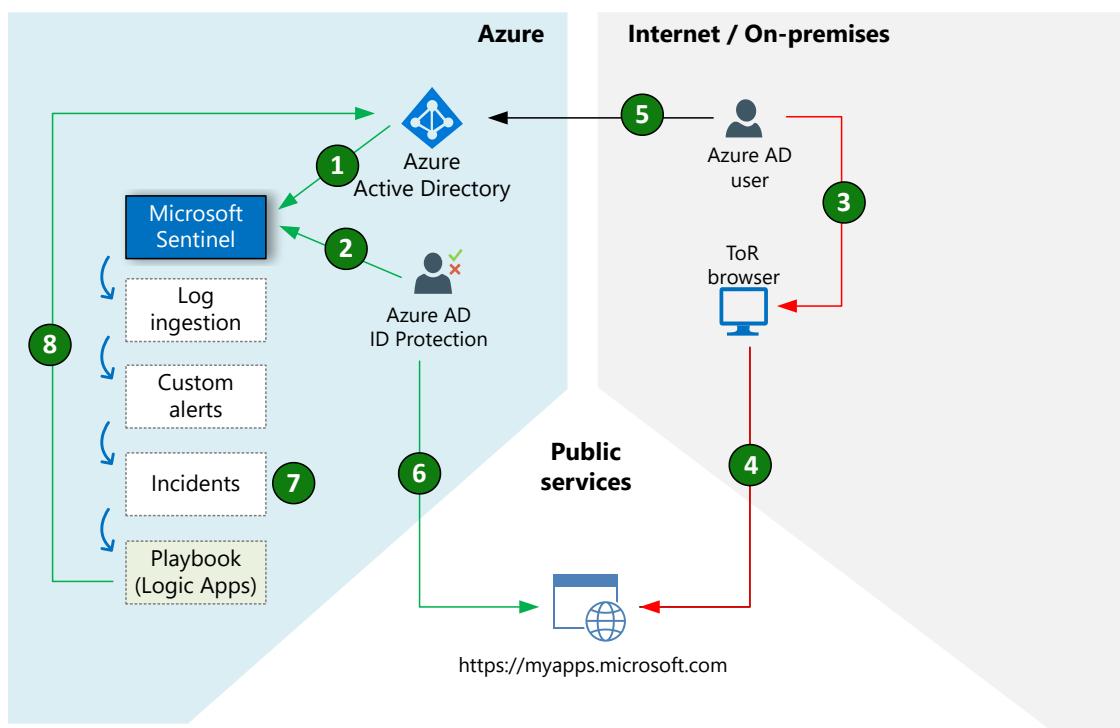
- Block a Microsoft Entra user
- Block a Microsoft Entra user based on an approve or reject email
- Post a message on the Microsoft Teams channel about an incident or alert
- Post a message on Slack
- Send an email that has incident or alert information
- Send an email that has a formatted incident report
- Confirm that a Microsoft Entra user is at risk
- Send an adaptive card via Microsoft Teams to confirm that a user is compromised
- Isolate an endpoint on Microsoft Defender for Endpoint

This article shows an example of implementing a playbook to respond to a threat. The playbook blocks a Microsoft Entra user that's compromised by suspicious activity.

Potential use case

The techniques described in this article apply whenever you need to implement an automatic response to a detectable condition.

Architecture

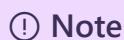


Download a [Visio file](#) of this architecture.

Workflow

This workflow shows the steps to deploy the playbook. Make sure that the [Prerequisites](#) are satisfied before you start. For example, you need to choose a Microsoft Entra user.

1. Follow the steps in [Send logs to Azure Monitor](#) to configure Microsoft Entra ID to send audit logs to the Log Analytics workspace that's used with Microsoft Sentinel.



Note

This solution doesn't use the audit logs, but you can use them to investigate what happens when the user is blocked.

2. Microsoft Entra ID Protection generates the alerts that trigger the threat response playbook to run. To have Microsoft Sentinel collect the alerts, navigate to your Microsoft Sentinel instance and select **Data Connectors**. Search for **Microsoft Entra ID Protection** and enable the collecting of alerts. For more information about Identity Protection, see [What is Identity Protection?](#).
3. [Install the ToR browser](#) onto a computer or virtual machine (VM) that you can use without putting your IT security at risk.
4. Use the Tor Browser to log in anonymously to My apps as the user that you selected for this solution. See [Anonymous IP address](#) for instructions on using the Tor Browser to simulate anonymous IP addresses.
5. Microsoft Entra authenticates the user.
6. Microsoft Entra ID Protection detects that the user used a ToR browser to log in anonymously. This type of login is suspicious activity that puts the user at risk. Identity Protection sends an alert to Microsoft Sentinel.
7. Configure Microsoft Sentinel to create an incident from the alert. See [Automatically create incidents from Microsoft security alerts](#) for information on doing this. The Microsoft security analytics rule template to use is [Create incidents based on Microsoft Entra ID Protection alerts](#).
8. When Microsoft Sentinel triggers an incident, the playbook responds with actions that block the user.

Components

- [Microsoft Sentinel](#) is a cloud-native SIEM and SOAR solution. It uses advanced AI and security analytics to detect and respond to threats across the enterprise. There are many playbooks on Microsoft Sentinel that you can use to automate your responses and protect your system.
- [Microsoft Entra ID](#) is a multi-tenant, cloud-based directory and identity management service that combines core directory services, application access management, and identity protection into a single solution. It can synchronize with on-premises directories. The identity service provides single sign-on, multifactor authentication, and conditional access to guard against cybersecurity attacks. The

solution shown in this article uses Microsoft Entra identity Protect to detect suspicious activity by a user.

- [Logic Apps](#) is a serverless cloud service for creating and running automated workflows that integrate apps, data, services, and systems. Developers can use a visual designer to schedule and orchestrate common task workflows. Logic Apps has [connectors](#) for many popular cloud services, on-premises products, and other software as a service applications. In this solution, Logic Apps runs the threat response playbook.

Considerations

- The Azure Well-Architected Framework is a set of guiding tenets that you can use to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).
- Microsoft Sentinel offers more than 50 playbooks that are ready for use. You can find them on the **Playbook templates** tab of the [Microsoft Sentinel|Automation](#) page for your workspace.
- [GitHub](#) has a variety of Microsoft Sentinel playbooks that are built by the community.

Deploy this scenario

You can deploy this scenario by following the steps in [Workflow](#) after making sure that the [Prerequisites](#) are satisfied.

Prerequisites

- [Prepare the software and choose a test user](#)
- [Deploy the playbook](#)

Prepare the software and choose a test user

To implement and test the playbook, you'll need Azure and Microsoft Sentinel along with the following:

- A Microsoft Entra ID Protection license (Premium P2, E3, or E5).
- A Microsoft Entra user. You can use either an existing user or [create a new user](#). If you do create a new user, you can delete it when you're done using it.
- A computer or VM that can run a ToR browser. You'll use the browser to log in to the My Apps portal as your Microsoft Entra user.

Deploy the playbook

To deploy a Microsoft Sentinel playbook, proceed as follows:

- If you don't have a Log Analytics workspace to use for this exercise, create a new one as follows:
 - Go to the [Microsoft Sentinel](#) main page, and select + **Create** to get to the **Add Microsoft Sentinel to a workspace** page.
 - Select **Create a new workspace**. Follow the instructions to create the new workspace. After a short time, the workspace is created.
- At this point, you have a workspace, perhaps one that you just created. Use the following steps to see whether Microsoft Sentinel has been added to it, and to add it if not:
 - Go to the [Microsoft Sentinel](#) main page.
 - If Microsoft Sentinel has already been added to your workspace, the workspace appears in the displayed list. If it hasn't been added yet, add it as follows.
 - Select + **Create** to get to the **Add Microsoft Sentinel to a workspace** page.
 - Select your workspace from the displayed list, and then select **Add** at the bottom of the page. After a short time, Microsoft Sentinel is added to your workspace.
- Create a playbook, as follows:
 - Go to the [Microsoft Sentinel](#) main page. Select your workspace. Select **Automation** from the left menu to get to the **Automation** page. This page has three tabs.
 - Select the **Playbook templates (Preview)** tab.
 - In the search field, enter **Block Microsoft Entra user - Incident**.
 - In the list of playbooks, select **Block Microsoft Entra user - Incident** and then select **Create playbook** in the bottom right corner to get to the **Create playbook** page.
 - On the **Create playbook** page, do the following:
 - Select values for **Subscription**, **Resource group**, and **Region** from the lists.
 - Enter a value for **Playbook name** if you don't want to use the default name that appears.
 - If you want, select **Enable diagnostics logs in Log Analytics** to enable logs.
 - Leave the **Associate with integration service environment** checkbox unchecked.
 - Leave **Integration service environment** empty.
 - Select **Next: Connections >** to go to the **Connections** tab of **Create playbook**.
 - Choose how you will authenticate within the playbook's components. Authentication is required for:
 - Microsoft Entra ID

- Microsoft Sentinel
- Office 365 Outlook

ⓘ Note

You can authenticate the resources during playbook customization under the logic app resource if you wish to enable later. To authenticate the above resources at this point, you need permissions to update a user on Microsoft Entra ID, and the user must have access to an email mailbox and must be able to send emails.

- Select **Next: Review and create** > to get to the **Review and create** tab of **Create playbook**.
- Select **Create and continue to designer** to create the playbook and access the **Logic app designer** page.

For more information about building logic apps, see [What is Azure Logic Apps](#) and [Quickstart: Create and manage logic app workflow definitions](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Rudnei Oliveira](#) | Senior Customer Engineer

Other contributors:

- [Andrew Nathan](#) | Senior Customer Engineering Manager
- [Lavanya Kasturi](#) | Technical Writer

Next steps

- [Overview of Azure Cloud Services?](#)
- [What is Microsoft Sentinel?](#)
- [Security orchestration, automation and response \(SOAR\) in Microsoft Sentinel.](#)
- [Automate threat response with playbooks in Microsoft Sentinel](#)
- [What is Microsoft Entra ID?](#)
- [What is Identity Protection?](#)
- [Simulating risk detections in Identity Protection](#)

- [What is Azure Logic Apps?](#)
- [Tutorial: Create automated approval-based workflows by using Azure Logic Apps](#)
- [Introduction to Microsoft Sentinel](#)

Related resources

- [Threat indicators for cyber threat intelligence in Microsoft Sentinel](#)
- [Monitor hybrid security using Microsoft Defender for Cloud and Microsoft Sentinel](#)
- [Security architecture design](#)

Storage architecture design

Article • 10/10/2023

The Azure Storage platform is the Microsoft cloud storage solution for modern data storage scenarios.

The Azure Storage platform includes the following data services:

- [Azure Blob Storage](#): A massively scalable object store for text and binary data. Also includes support for big data analytics through Azure Data Lake Storage Gen2.
- [Azure Files](#): Managed file shares for cloud or on-premises deployments.
- [Azure Queue Storage](#): A messaging store for reliable messaging between application components.
- [Azure Table Storage](#): A NoSQL store for schemaless storage of structured data.
- [Azure Disk Storage](#): Block-level storage volumes for Azure VMs.

Introduction to storage on Azure

If you're new to storage on Azure, the best way to learn more is [Microsoft Learn training](#). This free online platform provides interactive learning for Microsoft products and more. Check out the [Store data in Azure](#) learning path.

Path to production

- Choose the storage approach that best meets your needs and then create an account. For more information, see [Storage account overview](#).
- Be sure you understand security and reliability. See these articles:
 - [Azure Storage encryption for data at rest](#)
 - [Use private endpoints - Azure Storage](#)
 - [Data redundancy - Azure Storage](#)
 - [Disaster recovery and storage account failover - Azure Storage](#)
- For information about migrating existing data, see the [Azure Storage migration guide](#).

Best practices

Depending on the storage technology you use, see the following best practices resources:

- [Performance and scalability checklist for Blob Storage](#)
- [Best practices for using Azure Data Lake Storage Gen2](#)
- [Planning for an Azure Files deployment](#)
- [Performance and scalability checklist for Queue Storage](#)
- [Azure Storage table design patterns](#)

Blob Storage

See the following guides for information about Blob Storage:

- [Authorize access to blobs using Microsoft Entra ID](#)
- [Security recommendations for Blob Storage](#)

Azure Data Lake Storage

See the following guides for information about Data Lake Storage:

- [Best practices for using Azure Data Lake Storage Gen2](#)
- [Azure Policy Regulatory Compliance controls for Azure Data Lake Storage Gen1](#)

Azure Files

See the following guides for information about Azure Files:

- [Planning for an Azure Files deployment](#)
- [Overview of Azure Files identity-based authentication options for SMB access](#)
- [Disaster recovery and storage account failover](#)
- [About Azure file share backup](#)

Queue Storage

See the following guides for information about Queue Storage:

- [Authorize access to queues using Microsoft Entra ID](#)
- [Performance and scalability checklist for Queue Storage](#)

Table Storage

See the following guides for information about Table Storage:

- [Authorize access to tables using Microsoft Entra ID \(preview\)](#)
- [Performance and scalability checklist for Table storage](#)
- [Design scalable and performant tables](#)
- [Design for querying](#)

Azure Disk Storage

See the following guides for information about Azure managed disks:

- [Server-side encryption of Azure Disk Storage](#)
- [Azure Disk Encryption for Windows VMs](#)
- [Azure premium storage: design for high performance](#)
- [Scalability and performance targets for VM disks](#)

Stay current with storage

Get the [latest updates on Azure Storage products and features](#).

Additional resources

To plan for your storage needs, see [Review your storage options](#).

Example solutions

Here are a few sample implementations of storage on Azure:

- [Using Azure file shares in a hybrid environment](#)
- [Azure files accessed on-premises and secured by AD DS](#)
- [Enterprise file shares with disaster recovery](#)
- [Hybrid file services](#)
- [Optimized storage with logical data classification](#)
- [Medical data storage solutions](#)
- [HPC media rendering](#)

See more storage examples in the [Azure Architecture Center](#).

AWS or Google Cloud professionals

These articles provide service mapping and comparison between Azure and other cloud services. They can help you ramp up quickly on Azure.

- [Compare AWS and Azure Storage services](#)
- [Google Cloud to Azure services comparison - Storage](#)

Storage account overview

Article • 12/06/2023

An Azure storage account contains all of your Azure Storage data objects: blobs, files, queues, and tables. The storage account provides a unique namespace for your Azure Storage data that's accessible from anywhere in the world over HTTP or HTTPS. Data in your storage account is durable and highly available, secure, and massively scalable.

To learn how to create an Azure Storage account, see [Create a storage account](#).

Types of storage accounts

Azure Storage offers several types of storage accounts. Each type supports different features and has its own pricing model.

The following table describes the types of storage accounts recommended by Microsoft for most scenarios. All of these use the [Azure Resource Manager](#) deployment model.

Expand table

Type of storage account	Supported storage services	Redundancy options	Usage
Standard general-purpose v2	Blob Storage (including Data Lake Storage ¹), Queue Storage, Table Storage, and Azure Files	Locally redundant storage (LRS) / geo-redundant storage (GRS) / read-access geo-redundant storage (RA-GRS) Zone-redundant storage (ZRS) / geo-zone-redundant storage (GZRS) / read-access geo-zone-redundant storage (RA-GZRS) ²	Standard storage account type for blobs, file shares, queues, and tables. Recommended for most scenarios using Azure Storage. If you want support for network file system (NFS) in Azure Files, use the premium file shares account type.
Premium block blobs ³	Blob Storage (including Data Lake Storage ¹)	LRS ZRS ²	Premium storage account type for block blobs and append blobs. Recommended for scenarios with high transaction rates or that use smaller objects or require consistently low storage latency.

Type of storage account	Supported storage services	Redundancy options	Usage
			Learn more about example workloads
Premium file shares ³	Azure Files	LRS ZRS ²	Premium storage account type for file shares only. Recommended for enterprise or high-performance scale applications. Use this account type if you want a storage account that supports both Server Message Block (SMB) and NFS file shares.
Premium page blobs ³	Page blobs only	LRS ZRS ²	Premium storage account type for page blobs only. Learn more about page blobs and sample use cases .

¹ Data Lake Storage is a set of capabilities dedicated to big data analytics, built on Azure Blob Storage. For more information, see [Introduction to Data Lake Storage Gen2](#) and [Create a storage account to use with Data Lake Storage Gen2](#).

² ZRS, GZRS, and RA-GZRS are available only for standard general-purpose v2, premium block blobs, premium file shares, and premium page blobs accounts in certain regions. For more information, see [Azure Storage redundancy](#).

³ Premium performance storage accounts use solid-state drives (SSDs) for low latency and high throughput.

Legacy storage accounts are also supported. For more information, see [Legacy storage account types](#).

The service-level agreement (SLA) for Azure Storage accounts is available at [SLA for Storage Accounts](#).

Note

You can't change a storage account to a different type after it's created. To move your data to a storage account of a different type, you must create a new account and copy the data to the new account.

Storage account name

When naming your storage account, keep these rules in mind:

- Storage account names must be between 3 and 24 characters in length and may contain numbers and lowercase letters only.
- Your storage account name must be unique within Azure. No two storage accounts can have the same name.

Storage account endpoints

A storage account provides a unique namespace in Azure for your data. Every object that you store in Azure Storage has a URL address that includes your unique account name. The combination of the account name and the service endpoint forms the endpoints for your storage account.

There are two types of service endpoints available for a storage account:

- **Standard endpoints** (recommended). By default, you can create up to 250 storage accounts per region with standard endpoints in a given subscription. With a quota increase, you can create up to 500 storage accounts with standard endpoints per region. For more information, see [Increase Azure Storage account quotas](#).
- **Azure DNS zone endpoints** (preview). You can create up to 5000 storage accounts per region with Azure DNS zone endpoints in a given subscription.

Within a single subscription, you can create accounts with either standard or Azure DNS Zone endpoints, for a maximum of 5250 accounts per region per subscription. With a quota increase, you can create up to 5500 storage accounts per region per subscription.

You can configure your storage account to use a custom domain for the Blob Storage endpoint. For more information, see [Configure a custom domain name for your Azure Storage account](#).

Important

When referencing a service endpoint in a client application, it's recommended that you avoid taking a dependency on a cached IP address. The storage account IP address is subject to change, and relying on a cached IP address may result in unexpected behavior.

Additionally, it's recommended that you honor the time-to-live (TTL) of the DNS record and avoid overriding it. Overriding the DNS TTL may result in unexpected behavior.

Standard endpoints

A standard service endpoint in Azure Storage includes the protocol (HTTPS is recommended), the storage account name as the subdomain, and a fixed domain that includes the name of the service.

The following table lists the format for the standard endpoints for each of the Azure Storage services.

[Expand table](#)

Storage service	Endpoint
Blob Storage	<code>https://<storage-account>.blob.core.windows.net</code>
Static website (Blob Storage)	<code>https://<storage-account>.web.core.windows.net</code>
Data Lake Storage Gen2	<code>https://<storage-account>.dfs.core.windows.net</code>
Azure Files	<code>https://<storage-account>.file.core.windows.net</code>
Queue Storage	<code>https://<storage-account>.queue.core.windows.net</code>
Table Storage	<code>https://<storage-account>.table.core.windows.net</code>

When your account is created with standard endpoints, you can easily construct the URL for an object in Azure Storage by appending the object's location in the storage account to the endpoint. For example, the URL for a blob will be similar to:

`https://*mystorageaccount*.blob.core.windows.net/*mycontainer/*myblob*`

Azure DNS zone endpoints (preview)

ⓘ Important

Azure DNS zone endpoints are currently in PREVIEW. See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

When you create an Azure Storage account with Azure DNS zone endpoints (preview), Azure Storage dynamically selects an Azure DNS zone and assigns it to the storage account when it is created. The new storage account's endpoints are created in the dynamically selected Azure DNS zone. For more information about Azure DNS zones, see [DNS zones](#).

An Azure DNS zone service endpoint in Azure Storage includes the protocol (HTTPS is recommended), the storage account name as the subdomain, and a domain that includes the name of the service and the identifier for the DNS zone. The identifier for the DNS zone always begins with `z` and can range from `z00` to `z50`.

The following table lists the format for Azure DNS Zone endpoints for each of the Azure Storage services:

[+] [Expand table](#)

Storage service	Endpoint
Blob Storage	<code>https://<storage-account>.z[00-50].blob.storage.azure.net</code>
Static website (Blob Storage)	<code>https://<storage-account>.z[00-50].web.storage.azure.net</code>
Data Lake Storage Gen2	<code>https://<storage-account>.z[00-50].dfs.storage.azure.net</code>
Azure Files	<code>https://<storage-account>.z[00-50].file.storage.azure.net</code>
Queue Storage	<code>https://<storage-account>.z[00-50].queue.storage.azure.net</code>
Table Storage	<code>https://<storage-account>.z[00-50].table.storage.azure.net</code>

ⓘ Important

You can create up to 5000 accounts with Azure DNS Zone endpoints per subscription. However, you may need to update your application code to query for the account endpoint at runtime. You can call the [Get Properties](#) operation to query for the storage account endpoints.

Azure DNS zone endpoints are supported for accounts created with the Azure Resource Manager deployment model only. For more information, see [Azure Resource Manager overview](#).

To learn how to create a storage account with Azure DNS Zone endpoints, see [Create a storage account](#).

About the preview

The Azure DNS zone endpoints preview is available in all public regions. The preview is not available in any government cloud regions.

To register for the preview, follow the instructions provided in [Set up preview features in Azure subscription](#). Specify `PartitionedDnsPublicPreview` as the feature name and `Microsoft.Storage` as the provider namespace.

CNAME records, subdomains and IP addresses

Each storage account endpoint points to a chain of DNS CNAME records which eventually point to a DNS A record. The number of records and the subdomains that are associated with each record can vary between accounts and can depend on the storage account type and how the account is configured.

The storage account endpoint is stable and does not change. However, the CNAME records in a given chain can change and you won't be notified when a change occurs. If you host a private DNS service in Azure, then these changes can impact your configuration.

Consider the following guidelines:

- The CNAME chain associated with a storage account endpoint can change without notice. Applications and environments should not take a dependency on the number of CNAME records or the sub-domains that are associated with those CNAME records.
- The A record's IP address that is returned by the DNS resolution of a storage account endpoint can change frequently.
- The applications and operating systems should always honor the time-to-live (TTL) associated with the CNAME record. Caching the value of the CNAME record beyond the TTL could lead to unintended behavior.

Migrate a storage account

The following table summarizes and points to guidance on how to move, upgrade, or migrate a storage account:

[] [Expand table](#)

Migration scenario	Details
Move a storage account to a different subscription	Azure Resource Manager provides options for moving a resource to a different subscription. For more information, see Move resources to a new resource group or subscription .

Migration scenario	Details
Move a storage account to a different resource group	Azure Resource Manager provides options for moving a resource to a different resource group. For more information, see Move resources to a new resource group or subscription .
Move a storage account to a different region	To move a storage account, create a copy of your storage account in another region. Then, move your data to that account by using AzCopy, or another tool of your choice. For more information, see Move an Azure Storage account to another region .
Upgrade to a general-purpose v2 storage account	You can upgrade a general-purpose v1 storage account or Blob Storage account to a general-purpose v2 account. Note that this action can't be undone. For more information, see Upgrade to a general-purpose v2 storage account .
Migrate a classic storage account to Azure Resource Manager	The Azure Resource Manager deployment model is superior to the classic deployment model in terms of functionality, scalability, and security. For more information about migrating a classic storage account to Azure Resource Manager, see the "Migration of storage accounts" section of Platform-supported migration of IaaS resources from classic to Azure Resource Manager .

Transfer data into a storage account

Microsoft provides services and utilities for importing your data from on-premises storage devices or third-party cloud storage providers. Which solution you use depends on the quantity of data you're transferring. For more information, see [Azure Storage migration overview](#).

Storage account encryption

All data in your storage account is automatically encrypted on the service side. For more information about encryption and key management, see [Azure Storage encryption for data at rest](#).

Storage account billing

Azure Storage bills based on your storage account usage. All objects in a storage account are billed together as a group. Storage costs are calculated according to the following factors:

- **Region** refers to the geographical region in which your account is based.

- **Account type** refers to the type of storage account you're using.
- **Access tier** refers to the data usage pattern you've specified for your general-purpose v2 or Blob Storage account.
- **Capacity** refers to how much of your storage account allotment you're using to store data.
- **Redundancy** determines how many copies of your data are maintained at one time, and in what locations.
- **Transactions** refer to all read and write operations to Azure Storage.
- **Data egress** refers to any data transferred out of an Azure region. When the data in your storage account is accessed by an application that isn't running in the same region, you're charged for data egress. For information about using resource groups to group your data and services in the same region to limit egress charges, see [What is an Azure resource group?](#).

The [Azure Storage pricing page](#) provides detailed pricing information based on account type, storage capacity, replication, and transactions. The [Data Transfers pricing details](#) provides detailed pricing information for data egress. You can use the [Azure Storage pricing calculator](#) to help estimate your costs.

Azure services cost money. Azure Cost Management helps you set budgets and configure alerts to keep spending under control. Analyze, manage, and optimize your Azure costs with Cost Management. To learn more, see the [quickstart on analyzing your costs](#).

Legacy storage account types

The following table describes the legacy storage account types. These account types aren't recommended by Microsoft, but may be used in certain scenarios:

Expand table

Type of legacy storage account	Supported storage services	Redundancy options	Deployment model	Usage
Standard general-purpose v1	Blob Storage, Queue Storage, Table Storage, and Azure Files	LRS/GRS/RA-GRS	Resource Manager, classic ¹	<p>General-purpose v1 accounts may not have the latest features or the lowest per-gigabyte pricing.</p> <p>Consider using it for these scenarios:</p> <ul style="list-style-type: none"> • Your applications require the Azure classic deployment

Type of legacy storage account	Supported storage services	Redundancy options	Deployment model	Usage model ¹ .
				<ul style="list-style-type: none"> • Your applications are transaction-intensive or use significant geo-replication bandwidth, but don't require large capacity. In this case, a general-purpose v1 account may be the most economical choice. • You use a version of the Azure Storage REST API that is earlier than February 14, 2014, or a client library with a version lower than 4.x, and you can't upgrade your application. • You're selecting a storage account to use as a cache for Azure Site Recovery. Because Site Recovery is transaction-intensive, a general-purpose v1 account may be more cost-effective. For more information, see Support matrix for Azure VM disaster recovery between Azure regions.
Blob Storage	Blob Storage (block blobs and append blobs only)	LRS/GRS/RA-GRS	Resource Manager	Microsoft recommends using standard general-purpose v2 accounts instead when possible.

¹ Beginning August 1, 2022, you'll no longer be able to create new storage accounts with the classic deployment model. Resources created prior to that date will continue to be supported through August 31, 2024. For more information, see [Azure classic storage accounts will be retired on 31 August 2024](#).

Scalability targets for standard storage accounts

The following table describes default limits for Azure general-purpose v2 (GPv2), general-purpose v1 (GPv1), and Blob storage accounts. The *ingress* limit refers to all data that is sent to a storage account. The *egress* limit refers to all data that is received from a storage account.

Microsoft recommends that you use a GPv2 storage account for most scenarios. You can easily upgrade a GPv1 or a Blob storage account to a GPv2 account with no downtime and without the need to copy data. For more information, see [Upgrade to a GPv2 storage account](#).

 **Note**

You can request higher capacity and ingress limits. To request an increase, contact [Azure Support](#).

 [Expand table](#)

Resource	Limit
Maximum number of storage accounts with standard endpoints per region per subscription, including standard and premium storage accounts.	250 by default, 500 by request ¹
Maximum number of storage accounts with Azure DNS zone endpoints (preview) per region per subscription, including standard and premium storage accounts.	5000 (preview)
Default maximum storage account capacity	5 PiB ²
Maximum number of blob containers, blobs, directories and subdirectories (if Hierarchical Namespace is enabled), file shares, tables, queues, entities, or messages per storage account.	No limit
Default maximum request rate per storage account	20,000 requests per second ²
Default maximum ingress per general-purpose v2 and Blob storage account in the following regions:	60 Gbps ²
<ul style="list-style-type: none">• East Asia• Southeast Asia• Australia East• Brazil South• Canada Central• China East 2• China North 3• North Europe	

Resource	Limit
<ul style="list-style-type: none"> • West Europe • France Central • Germany West Central • Central India • Japan East • Jio India West • Korea Central • Norway East • South Africa North • Sweden Central • UAE North • UK South • Central US • East US • East US 2 • USGov Virginia • USGov Arizona • North Central US • South Central US • West US • West US 2 • West US 3 	
Default maximum ingress per general-purpose v2 and Blob storage account in regions that aren't listed in the previous row.	25 Gbps ²
Default maximum ingress for general-purpose v1 storage accounts (all regions)	10 Gbps ²
Default maximum egress for general-purpose v2 and Blob storage accounts in the following regions: <ul style="list-style-type: none"> • East Asia • Southeast Asia • Australia East • Brazil South • Canada Central • China East 2 • China North 3 • North Europe • West Europe • France Central • Germany West Central • Central India • Japan East • Jio India West • Korea Central • Norway East 	120 Gbps ²

Resource	Limit
<ul style="list-style-type: none"> • South Africa North • Sweden Central • UAE North • UK South • Central US • East US • East US 2 • USGov Virginia • USGov Arizona • North Central US • South Central US • West US • West US 2 • West US 3 	
Default maximum egress for general-purpose v2 and Blob storage accounts in regions that aren't listed in the previous row.	50 Gbps ²
Maximum egress for general-purpose v1 storage accounts (US regions)	20 Gbps if RA-GRS/GRS is enabled, 30 Gbps for LRS/ZRS
Maximum egress for general-purpose v1 storage accounts (non-US regions)	10 Gbps if RA-GRS/GRS is enabled, 15 Gbps for LRS/ZRS
Maximum number of IP address rules per storage account	200
Maximum number of virtual network rules per storage account	200
Maximum number of resource instance rules per storage account	200
Maximum number of private endpoints per storage account	200

¹ With a quota increase, you can create up to 500 storage accounts with standard endpoints per region. For more information, see [Increase Azure Storage account quotas](#).

² Azure Storage standard accounts support higher capacity limits and higher limits for ingress and egress by request. To request an increase in account limits, contact [Azure Support](#).

Next steps

- [Create a storage account](#)
- [Upgrade to a general-purpose v2 storage account](#)
- [Recover a deleted storage account](#)

Azure Storage encryption for data at rest

Article • 02/13/2023

Azure Storage uses service-side encryption (SSE) to automatically encrypt your data when it is persisted to the cloud. Azure Storage encryption protects your data and to help you to meet your organizational security and compliance commitments.

Microsoft recommends using service-side encryption to protect your data for most scenarios. However, the Azure Storage client libraries for Blob Storage and Queue Storage also provide client-side encryption for customers who need to encrypt data on the client. For more information, see [Client-side encryption for blobs and queues](#).

About Azure Storage service-side encryption

Data in Azure Storage is encrypted and decrypted transparently using 256-bit [AES encryption](#), one of the strongest block ciphers available, and is FIPS 140-2 compliant. Azure Storage encryption is similar to BitLocker encryption on Windows.

Azure Storage encryption is enabled for all storage accounts, including both Resource Manager and classic storage accounts. Azure Storage encryption cannot be disabled. Because your data is secured by default, you don't need to modify your code or applications to take advantage of Azure Storage encryption.

Data in a storage account is encrypted regardless of performance tier (standard or premium), access tier (hot or cool), or deployment model (Azure Resource Manager or classic). All new and existing block blobs, append blobs, and page blobs are encrypted, including blobs in the archive tier. All Azure Storage redundancy options support encryption, and all data in both the primary and secondary regions is encrypted when geo-replication is enabled. All Azure Storage resources are encrypted, including blobs, disks, files, queues, and tables. All object metadata is also encrypted.

There is no additional cost for Azure Storage encryption.

For more information about the cryptographic modules underlying Azure Storage encryption, see [Cryptography API: Next Generation](#).

For information about encryption and key management for Azure managed disks, see [Server-side encryption of Azure managed disks](#).

About encryption key management

Data in a new storage account is encrypted with Microsoft-managed keys by default. You can continue to rely on Microsoft-managed keys for the encryption of your data, or you can manage encryption with your own keys. If you choose to manage encryption with your own keys, you have two options. You can use either type of key management, or both:

- You can specify a *customer-managed key* to use for encrypting and decrypting data in Blob Storage and in Azure Files.^{1,2} Customer-managed keys must be stored in Azure Key Vault or Azure Key Vault Managed Hardware Security Model (HSM). For more information about customer-managed keys, see [Use customer-managed keys for Azure Storage encryption](#).
- You can specify a *customer-provided key* on Blob Storage operations. A client making a read or write request against Blob Storage can include an encryption key on the request for granular control over how blob data is encrypted and decrypted. For more information about customer-provided keys, see [Provide an encryption key on a request to Blob Storage](#).

By default, a storage account is encrypted with a key that is scoped to the entire storage account. Encryption scopes enable you to manage encryption with a key that is scoped to a container or an individual blob. You can use encryption scopes to create secure boundaries between data that resides in the same storage account but belongs to different customers. Encryption scopes can use either Microsoft-managed keys or customer-managed keys. For more information about encryption scopes, see [Encryption scopes for Blob storage](#).

The following table compares key management options for Azure Storage encryption.

Key management parameter	Microsoft-managed keys	Customer-managed keys	Customer-provided keys
Encryption/decryption operations	Azure	Azure	Azure
Azure Storage services supported	All	Blob Storage, Azure Files ^{1,2}	Blob Storage
Key storage	Microsoft key store	Azure Key Vault or Key Vault HSM	Customer's own key store
Key rotation responsibility	Microsoft	Customer	Customer
Key control	Microsoft	Customer	Customer

Key management parameter	Microsoft-managed keys	Customer-managed keys	Customer-provided keys
Key scope	Account (default), container, or blob	Account (default), container, or blob	N/A

¹ For information about creating an account that supports using customer-managed keys with Queue storage, see [Create an account that supports customer-managed keys for queues](#).

² For information about creating an account that supports using customer-managed keys with Table storage, see [Create an account that supports customer-managed keys for tables](#).

 **Note**

Microsoft-managed keys are rotated appropriately per compliance requirements. If you have specific key rotation requirements, Microsoft recommends that you move to customer-managed keys so that you can manage and audit the rotation yourself.

Doubly encrypt data with infrastructure encryption

Customers who require high levels of assurance that their data is secure can also enable 256-bit AES encryption at the Azure Storage infrastructure level. When infrastructure encryption is enabled, data in a storage account is encrypted twice — once at the service level and once at the infrastructure level — with two different encryption algorithms and two different keys. Double encryption of Azure Storage data protects against a scenario where one of the encryption algorithms or keys may be compromised. In this scenario, the additional layer of encryption continues to protect your data.

Service-level encryption supports the use of either Microsoft-managed keys or customer-managed keys with Azure Key Vault. Infrastructure-level encryption relies on Microsoft-managed keys and always uses a separate key.

For more information about how to create a storage account that enables infrastructure encryption, see [Create a storage account with infrastructure encryption enabled for double encryption of data](#).

Client-side encryption for blobs and queues

The Azure Blob Storage client libraries for .NET, Java, and Python support encrypting data within client applications before uploading to Azure Storage, and decrypting data while downloading to the client. The Queue Storage client libraries for .NET and Python also support client-side encryption.

 **Note**

Consider using the service-side encryption features provided by Azure Storage to protect your data, instead of client-side encryption.

The Blob Storage and Queue Storage client libraries uses [AES](#) in order to encrypt user data. There are two versions of client-side encryption available in the client libraries:

- Version 2 uses [Galois/Counter Mode \(GCM\)](#) mode with AES. The Blob Storage and Queue Storage SDKs support client-side encryption with v2.
- Version 1 uses [Cipher Block Chaining \(CBC\)](#) mode with AES. The Blob Storage, Queue Storage, and Table Storage SDKs support client-side encryption with v1.

 **Warning**

Using client-side encryption v1 is no longer recommended due to a security vulnerability in the client library's implementation of CBC mode. For more information about this security vulnerability, see [Azure Storage updating client-side encryption in SDK to address security vulnerability](#). If you are currently using v1, we recommend that you update your application to use client-side encryption v2 and migrate your data.

The Azure Table Storage SDK supports only client-side encryption v1. Using client-side encryption with Table Storage is not recommended.

The following table shows which client libraries support which versions of client-side encryption and provides guidelines for migrating to client-side encryption v2.

Client library	Version of client-side encryption supported	Recommended migration	Additional guidance

Client library	Version of client-side encryption supported	Recommended migration	Additional guidance
Blob Storage client libraries for .NET (version 12.13.0 and above), Java (version 12.18.0 and above), and Python (version 12.13.0 and above)	2.0 1.0 (for backward compatibility only)	Update your code to use client-side encryption v2. Download any encrypted data to decrypt it, then reencrypt it with client-side encryption v2.	Client-side encryption for blobs
Blob Storage client library for .NET (version 12.12.0 and below), Java (version 12.17.0 and below), and Python (version 12.12.0 and below)	1.0 (not recommended)	Update your application to use a version of the Blob Storage SDK that supports client-side encryption v2. See SDK support matrix for client-side encryption for details.	Client-side encryption for blobs
		Update your code to use client-side encryption v2. Download any encrypted data to decrypt it, then reencrypt it with client-side encryption v2.	
Queue Storage client library for .NET (version 12.11.0 and above) and Python (version 12.4 and above)	2.0 1.0 (for backward compatibility only)	Update your code to use client-side encryption v2.	Client-side encryption for queues
Queue Storage client library for .NET (version 12.10.0 and below) and Python (version 12.3.0 and below)	1.0 (not recommended)	Update your application to use a version of the Queue Storage SDK version that supports client-side encryption v2. See SDK support matrix for client-side encryption	Client-side encryption for queues
Table Storage client library for .NET, Java, and Python	1.0 (not recommended)	Update your code to use client-side encryption v2. Not available.	N/A

Next steps

- What is Azure Key Vault?
- Customer-managed keys for Azure Storage encryption
- Encryption scopes for Blob Storage
- Provide an encryption key on a request to Blob Storage

Azure Policy Regulatory Compliance controls for Azure Storage

Article • 01/02/2024

[Regulatory Compliance in Azure Policy](#) provides Microsoft created and managed initiative definitions, known as *built-ins*, for the **compliance domains** and **security controls** related to different compliance standards. This page lists the **compliance domains** and **security controls** for Azure Storage. You can assign the built-ins for a **security control** individually to help make your Azure resources compliant with the specific standard.

The title of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Policy Version** column to view the source on the [Azure Policy GitHub repo](#).

Important

Each control is associated with one or more [Azure Policy](#) definitions. These policies might help you [assess compliance](#) with the control. However, there often isn't a one-to-one or complete match between a control and one or more policies. As such, **Compliant** in Azure Policy refers only to the policies themselves. This doesn't ensure that you're fully compliant with all requirements of a control. In addition, the compliance standard includes controls that aren't addressed by any Azure Policy definitions at this time. Therefore, compliance in Azure Policy is only a partial view of your overall compliance status. The associations between controls and Azure Policy Regulatory Compliance definitions for these compliance standards can change over time.

Australian Government ISM PROTECTED

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Australian Government ISM PROTECTED](#). For more information about this compliance standard, see [Australian Government ISM PROTECTED](#).

 [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for Networking - Network design and configuration	520	Network access controls - 520	Storage accounts should restrict network access ↗	1.1.1 ↗
Guidelines for Networking - Network design and configuration	1182	Network access controls - 1182	Storage accounts should restrict network access ↗	1.1.1 ↗
Guidelines for Database Systems - Database servers	1277	Communications between database servers and web servers - 1277	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Guidelines for System Hardening - Authentication hardening	1546	Authenticating to systems - 1546	Storage accounts should restrict network access ↗	1.1.1 ↗

Canada Federal PBMM

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Canada Federal PBMM](#). For more information about this compliance standard, see [Canada Federal PBMM ↗](#).

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-17(1)	Remote Access Automated Monitoring / Control	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	SC-8(1)	Transmission Confidentiality and Integrity Cryptographic or Alternate Physical Protection	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

CIS Microsoft Azure Foundations Benchmark

1.1.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CIS Microsoft Azure Foundations Benchmark 1.1.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark](#).

 [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
3 Storage Accounts	3.1	Ensure that 'Secure transfer required' is set to 'Enabled'	Secure transfer to storage accounts should be enabled	2.0.0
3 Storage Accounts	3.6	Ensure that 'Public access level' is set to Private for blob containers	[Preview]: Storage account public access should be disallowed	3.1.0-preview
3 Storage Accounts	3.7	Ensure default network access rule for Storage Accounts is set to deny	Storage accounts should restrict network access	1.1.1
3 Storage Accounts	3.8	Ensure 'Trusted Microsoft Services' is enabled for Storage Account access	Storage accounts should allow access from trusted Microsoft services	1.0.0
5 Logging and Monitoring	5.1.5	Ensure the storage container storing the activity logs is not publicly accessible	[Preview]: Storage account public access should be disallowed	3.1.0-preview
5 Logging and Monitoring	5.1.6	Ensure the storage account containing the container with activity logs is encrypted with BYOK (Use Your Own Key)	Storage account containing the container with activity logs must be encrypted with BYOK	1.0.0

CIS Microsoft Azure Foundations Benchmark

1.3.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CIS Microsoft Azure Foundations Benchmark 1.3.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark](#).

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
3 Storage Accounts	3.1	Ensure that 'Secure transfer required' is set to 'Enabled'	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
3 Storage Accounts	3.5	Ensure that 'Public access level' is set to Private for blob containers	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
3 Storage Accounts	3.6	Ensure default network access rule for Storage Accounts is set to deny	Storage accounts should restrict network access ↗	1.1.1 ↗
3 Storage Accounts	3.6	Ensure default network access rule for Storage Accounts is set to deny	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
3 Storage Accounts	3.7	Ensure 'Trusted Microsoft Services' is enabled for Storage Account access	Storage accounts should allow access from trusted Microsoft services ↗	1.0.0 ↗
3 Storage Accounts	3.9	Ensure storage for critical data are encrypted with Customer Managed Key	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
5 Logging and Monitoring	5.1.3	Ensure the storage container storing the activity logs is not publicly accessible	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
5 Logging and Monitoring	5.1.4	Ensure the storage account containing the container with activity logs is encrypted with BYOK (Use Your Own Key)	Storage account containing the container with activity logs must be encrypted with BYOK ↗	1.0.0 ↗

CIS Microsoft Azure Foundations Benchmark

1.4.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for CIS v1.4.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark](#) ↗.

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
3 Storage Accounts	3.1	Ensure that 'Secure transfer required' is set to 'Enabled'	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
3 Storage Accounts	3.5	Ensure that 'Public access level' is set to Private for blob containers	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
3 Storage Accounts	3.6	Ensure Default Network Access Rule for Storage Accounts is Set to Deny	Storage accounts should restrict network access ↗	1.1.1 ↗
3 Storage Accounts	3.6	Ensure Default Network Access Rule for Storage Accounts is Set to Deny	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
3 Storage Accounts	3.7	Ensure 'Trusted Microsoft Services' are Enabled for Storage Account Access	Storage accounts should allow access from trusted Microsoft services ↗	1.0.0 ↗
3 Storage Accounts	3.9	Ensure Storage for Critical Data are Encrypted with Customer Managed Keys	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
5 Logging and Monitoring	5.1.3	Ensure the storage container storing the activity logs is not publicly accessible	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
5 Logging and Monitoring	5.1.4	Ensure the storage account containing the container with activity logs is encrypted with BYOK (Use Your Own Key)	Storage account containing the container with activity logs must be encrypted with BYOK ↗	1.0.0 ↗

CIS Microsoft Azure Foundations Benchmark

2.0.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for CIS v2.0.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark ↗](#).

expand Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
3	3.1	Ensure that 'Secure transfer required' is set to 'Enabled'	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
3	3.10	Ensure Private Endpoints are used to access Storage Accounts	Storage accounts should use private link ↗	2.0.0 ↗
3	3.12	Ensure Storage for Critical Data are Encrypted with Customer Managed Keys	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
3	3.15	Ensure the "Minimum TLS version" for storage accounts is set to "Version 1.2"	Storage accounts should have the specified minimum TLS version ↗	1.0.0 ↗
3	3.2	Ensure that 'Enable Infrastructure Encryption' for Each Storage Account in Azure Storage is Set to 'enabled'	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗
3	3.7	Ensure that 'Public access level' is disabled for storage accounts with blob containers	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
3	3.8	Ensure Default Network Access Rule for Storage Accounts is Set to Deny	Storage accounts should restrict network access ↗	1.1.1 ↗
3	3.8	Ensure Default Network Access Rule for Storage Accounts is Set to Deny	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
3	3.9	Ensure 'Allow Azure services on the trusted services list to access this storage account' is Enabled for Storage Account Access	Storage accounts should allow access from trusted Microsoft services ↗	1.0.0 ↗
5.1	5.1.3	Ensure the Storage Container Storing the Activity Logs is not Publicly Accessible	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
5.1	5.1.4	Ensure the storage account containing the container with activity logs is encrypted with Customer Managed Key	Storage account containing the container with activity logs must be encrypted with BYOK ↗	1.0.0 ↗

CMMC Level 3

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CMMC Level 3](#). For more information about this compliance standard, see [Cybersecurity Maturity Model Certification \(CMMC\)](#).

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	[Preview]: Storage account public access should be disallowed	3.1.0-preview
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Storage accounts should allow access from trusted Microsoft services	1.0.0
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Storage accounts should restrict network access	1.1.1
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	[Preview]: Storage account public access should be disallowed	3.1.0-preview
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	Secure transfer to storage accounts should be enabled	2.0.0
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	Storage accounts should allow access from trusted Microsoft services	1.0.0
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	Storage accounts should restrict network access	1.1.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC.2.013	Monitor and control remote access sessions.	Storage accounts should restrict network access	1.1.1 ↗
Access Control	AC.2.016	Control the flow of CUI in accordance with approved authorizations.	[Preview]: Storage account public access should be disallowed	3.1.0-preview ↗
Access Control	AC.2.016	Control the flow of CUI in accordance with approved authorizations.	Storage accounts should restrict network access	1.1.1 ↗
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	[Preview]: Storage account public access should be disallowed	3.1.0-preview ↗
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Storage accounts should restrict network access	1.1.1 ↗
Incident Response	IR.2.093	Detect and report events.	Deploy Defender for Storage (Classic) on storage accounts	1.0.1 ↗
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	[Preview]: Storage account public access should be disallowed	3.1.0-preview ↗
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Secure transfer to storage accounts should be enabled	2.0.0 ↗
System and Communications	SC.1.175	Monitor, control, and protect communications (i.e.,	Storage accounts should restrict	1.1.1 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection		information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	network access ↗	
System and Communications Protection	SC.1.176	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	SC.3.177	Employ FIPS-validated cryptography when used to protect the confidentiality of CUI.	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗
System and Communications Protection	SC.3.177	Employ FIPS-validated cryptography when used to protect the confidentiality of CUI.	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	[Preview]: Storage account public access should be disallowed ↗	3.1.0- preview ↗
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Storage accounts should allow access from trusted Microsoft services ↗	1.0.0 ↗
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	SC.3.185	Implement cryptographic mechanisms to prevent unauthorized disclosure of CUI during transmission unless otherwise protected by alternative physical safeguards.	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	SC.3.185	Implement cryptographic mechanisms to prevent unauthorized disclosure of CUI during transmission unless otherwise protected by alternative physical safeguards.	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	SC.3.191	Protect the confidentiality of CUI at rest.	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗
System and Communications Protection	SC.3.191	Protect the confidentiality of CUI at rest.	Storage accounts should restrict network access ↗	1.1.1 ↗

FedRAMP High

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - FedRAMP High](#). For more information about this compliance standard, see [FedRAMP High \[↗\]\(#\)](#).

[↗](#) Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-3	Access Enforcement	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Access Control	AC-4	Information Flow Enforcement	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-4	Information Flow Enforcement	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17	Remote Access	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17	Remote Access	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17 (1)	Automated Monitoring / Control	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17 (1)	Automated Monitoring / Control	Storage accounts should use private link ↗	2.0.0 ↗
Contingency Planning	CP-6	Alternate Storage Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
Contingency Planning	CP-6 (1)	Separation From Primary Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
System And Communications Protection	SC-7	Boundary Protection	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access ↗	1.1.1 ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should use private link ↗	2.0.0 ↗
System And Communications Protection	SC-7 (3)	Access Points	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System And Communications Protection	SC-7 (3)	Access Points	Storage accounts should restrict network access ↗	1.1.1 ↗
System And Communications	SC-7 (3)	Access Points	Storage accounts should restrict network access using	1.0.1 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection				virtual network rules
System And Communications Protection	SC-7 (3)	Access Points	Storage accounts should use private link	2.0.0
System And Communications Protection	SC-8	Transmission Confidentiality And Integrity	Secure transfer to storage accounts should be enabled	2.0.0
System And Communications Protection	SC-8 (1)	Cryptographic Or Alternate Physical Protection	Secure transfer to storage accounts should be enabled	2.0.0
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Storage account encryption scopes should use customer-managed keys to encrypt data at rest	1.0.0
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Storage accounts should use customer-managed key for encryption	1.0.3
System And Communications Protection	SC-28	Protection Of Information At Rest	Storage accounts should have infrastructure encryption	1.0.0
System And Communications Protection	SC-28 (1)	Cryptographic Protection	Storage accounts should have infrastructure encryption	1.0.0

FedRAMP Moderate

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - FedRAMP Moderate](#). For more information about this compliance standard, see [FedRAMP Moderate](#).

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-3	Access Enforcement	Storage accounts should be migrated to new Azure	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			Resource Manager resources ↗	
Access Control	AC-4	Information Flow Enforcement	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17	Remote Access	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17	Remote Access	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17 (1)	Automated Monitoring / Control	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17 (1)	Automated Monitoring / Control	Storage accounts should use private link ↗	2.0.0 ↗
Contingency Planning	CP-6	Alternate Storage Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
Contingency Planning	CP-6 (1)	Separation From Primary Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
System And Communications Protection	SC-7	Boundary Protection	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access ↗	1.1.1 ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should use private link ↗	2.0.0 ↗
System And Communications Protection	SC-7 (3)	Access Points	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System And Communications Protection	SC-7 (3)	Access Points	Storage accounts should restrict network access ↗	1.1.1 ↗
System And Communications Protection	SC-7 (3)	Access Points	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
System And Communications Protection	SC-7 (3)	Access Points	Storage accounts should use private link ↗	2.0.0 ↗
System And Communications Protection	SC-8	Transmission Confidentiality And Integrity	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
System And Communications Protection	SC-8 (1)	Cryptographic Or Alternate Physical Protection	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Storage account encryption scopes should use customer-managed keys to encrypt data at rest ↗	1.0.0 ↗
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
System And Communications Protection	SC-28	Protection Of Information At Rest	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗
System And Communications Protection	SC-28 (1)	Cryptographic Protection	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗

HIPAA HITRUST 9.2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - HIPAA HITRUST 9.2](#). For more information about this compliance standard, see [HIPAA HITRUST 9.2](#).

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification of Risks Related to External Parties	1401.05i1Organizational.1239 - 05.i	Access to the organizations information and systems by external parties is not permitted until due diligence has been conducted, the appropriate controls have been implemented, and a contract/agreement reflecting the security requirements is signed acknowledging they understand and accept their obligations.	Secure transfer to storage accounts should be enabled	2.0.0
08 Network Protection	0805.01m1Organizational.12-01.m	0805.01m1Organizational.12-01.m 01.04 Network Access Control	Storage Accounts should use a virtual network service endpoint	1.0.0
08 Network Protection	0806.01m2Organizational.12356-01.m	0806.01m2Organizational.12356-01.m 01.04 Network Access Control	Storage Accounts should use a virtual network service endpoint	1.0.0
08 Network Protection	0809.01n2Organizational.1234-01.n	0809.01n2Organizational.1234-01.n 01.04 Network Access Control	Secure transfer to storage accounts should be enabled	2.0.0
08 Network Protection	0810.01n2Organizational.5-01.n	0810.01n2Organizational.5-01.n 01.04 Network Access Control	Secure transfer to storage accounts	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			should be enabled ↗	
08 Network Protection	0811.01n2Organizational.6-01.n	0811.01n2Organizational.6-01.n 01.04 Network Access Control	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
08 Network Protection	0812.01n2Organizational.8-01.n	0812.01n2Organizational.8-01.n 01.04 Network Access Control	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
08 Network Protection	0814.01n1Organizational.12-01.n	0814.01n1Organizational.12-01.n 01.04 Network Access Control	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
08 Network Protection	0866.09m3Organizational.1516-09.m	0866.09m3Organizational.1516-09.m 09.06 Network Security Management	Storage accounts should restrict network access ↗	1.1.1 ↗
08 Network Protection	0894.01m2Organizational.7-01.m	0894.01m2Organizational.7-01.m 01.04 Network Access Control	Storage Accounts should use a virtual network service endpoint ↗	1.0.0 ↗
Network Controls	0867.09m3Organizational.17 - 09.m	Wireless access points are placed in secure areas and shut down when not in use (e.g. nights, weekends).	Storage Accounts should use a virtual network service endpoint ↗	1.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
09 Transmission Protection	0943.09y1Organizational.1-09.y	0943.09y1Organizational.1-09.y 09.09 Electronic Commerce Services	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

IRS 1075 September 2016

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - IRS 1075 September 2016](#). For more information about this compliance standard, see [IRS 1075 September 2016 ↗](#).

expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	9.3.1.12	Remote Access (AC-17)	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	9.3.16.5	Boundary Protection (SC-7)	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	9.3.16.6	Transmission Confidentiality and Integrity (SC-8)	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

ISO 27001:2013

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - ISO 27001:2013](#). For more information about this compliance standard, see [ISO 27001:2013 ↗](#).

expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Cryptography	10.1.1	Policy on the use of cryptographic controls	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Communications Security	13.1.1	Network controls	Storage accounts should restrict network access ↗	1.1.1 ↗
Communications Security	13.2.1	Information transfer policies and procedures	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Access Control	9.1.2	Access to networks and network services	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗

Microsoft Cloud for Sovereignty Baseline Confidential Policies

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for MCFS Sovereignty Baseline Confidential Policies](#). For more information about this compliance standard, see [Microsoft Cloud for Sovereignty Policy portfolio](#).

↔ [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
SO.3 - Customer-Managed Keys	SO.3	Azure products must be configured to use Customer-Managed Keys when possible.	Queue Storage should use customer-managed key for encryption ↗	1.0.0 ↗
SO.3 - Customer-Managed Keys	SO.3	Azure products must be configured to use Customer-Managed Keys when possible.	Storage account encryption scopes should use customer-managed keys to encrypt data at rest ↗	1.0.0 ↗
SO.3 - Customer-Managed Keys	SO.3	Azure products must be configured to use Customer-Managed Keys when possible.	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
SO.3 - Customer-Managed Keys	SO.3	Azure products must be configured to use Customer-Managed Keys when possible.	Table Storage should use customer-managed key for encryption ↗	1.0.0 ↗

Microsoft cloud security benchmark

The [Microsoft cloud security benchmark](#) provides recommendations on how you can secure your cloud solutions on Azure. To see how this service completely maps to the Microsoft cloud security benchmark, see the [Azure Security Benchmark mapping files \[↗\]\(#\)](#).

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Microsoft cloud security benchmark](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Network Security	NS-2	Secure cloud services with network controls	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
Network Security	NS-2	Secure cloud services with network controls	Storage accounts should restrict network access ↗	1.1.1 ↗
Network Security	NS-2	Secure cloud services with network controls	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
Network Security	NS-2	Secure cloud services with network controls	Storage accounts should use private link ↗	2.0.0 ↗
Identity Management	IM-1	Use centralized identity and authentication system	Storage accounts should prevent shared key access ↗	2.0.0 ↗
Data Protection	DP-3	Encrypt sensitive data in transit	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Data Protection	DP-5	Use customer-managed key option in data at rest encryption when required	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
Asset Management	AM-2	Use only approved services		
Storage accounts should be migrated to new Azure Resource Manager resources	1.0.0 ↗			

New Zealand ISM Restricted

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - New Zealand ISM Restricted](#). For more information about this compliance standard, see [New Zealand ISM Restricted ↗](#).

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Physical Security	PS-4	8.3.5 Network infrastructure in unsecure areas	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Infrastructure	INF-9	10.8.35 Security Architecture	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
Infrastructure	INF-9	10.8.35 Security Architecture	Storage accounts should use private link ↗	2.0.0 ↗
Cryptography	CR-3	17.1.46 Reducing storage and physical transfer requirements	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
Gateway security	GS-2	19.1.11 Using Gateways	Storage account keys should not be expired ↗	3.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Gateway security	GS-3	19.1.12 Configuration of Gateways	Storage accounts should restrict network access ↗	1.1.1 ↗

NIST SP 800-171 R2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-171 R2](#). For more information about this compliance standard, see [NIST SP 800-171 R2](#) [↗](#).

[Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	3.1.12	Monitor and control remote access sessions.	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	3.1.12	Monitor and control remote access sessions.	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	3.1.13	Employ cryptographic mechanisms to protect the confidentiality of remote access sessions.	Storage accounts should restrict network access ↗	1.1.1 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	3.1.13	Employ cryptographic mechanisms to protect the confidentiality of remote access sessions.	Storage accounts should use private link	2.0.0 ↗
Access Control	3.1.14	Route remote access via managed access control points.	Storage accounts should restrict network access	1.1.1 ↗
Access Control	3.1.14	Route remote access via managed access control points.	Storage accounts should use private link	2.0.0 ↗
Access Control	3.1.2	Limit system access to the types of transactions and functions that authorized users are permitted to execute.	Storage accounts should be migrated to new Azure Resource Manager resources	1.0.0 ↗
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	[Preview]: Storage account public access should be disallowed	3.1.0-preview ↗
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Storage accounts should restrict network access	1.1.1 ↗
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Storage accounts should restrict network access using virtual network rules	1.0.1 ↗
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Storage accounts should use private link	2.0.0 ↗
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	[Preview]: Storage account public access should be disallowed	3.1.0-preview ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Storage accounts should restrict network access	1.1.1 ↗
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Storage accounts should restrict network access using virtual network rules	1.0.1 ↗
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Storage accounts should use private link	2.0.0 ↗
System and Communications Protection	3.13.10	Establish and manage cryptographic keys for cryptography employed in organizational systems.	Storage account encryption scopes should use customer-managed keys to encrypt data at rest	1.0.0 ↗
System and Communications Protection	3.13.10	Establish and manage cryptographic keys for cryptography employed in organizational systems.	Storage accounts should use customer-managed key for encryption	1.0.3 ↗
System and Communications Protection	3.13.16	Protect the confidentiality of CUI at rest.	Storage accounts should have infrastructure encryption	1.0.0 ↗
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that	[Preview]: Storage account public access should be disallowed	3.1.0-preview ↗

Domain	Control ID	Control title Promote effective information security within organizational systems.	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Storage accounts should restrict network access	1.1.1 ↗
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Storage accounts should restrict network access using virtual network rules	1.0.1 ↗
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Storage accounts should use private link	2.0.0 ↗
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	[Preview]: Storage account public access should be disallowed	3.1.0-preview ↗
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Storage accounts should restrict network access	1.1.1 ↗
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Storage accounts should restrict network access using virtual network rules	1.0.1 ↗
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Storage accounts should use private link	2.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
System and Communications Protection	3.13.8	Implement cryptographic mechanisms to prevent unauthorized disclosure of CUI during transmission unless otherwise protected by alternative physical safeguards.	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

NIST SP 800-53 Rev. 4

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 4](#). For more information about this compliance standard, see [NIST SP 800-53 Rev. 4](#).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-3	Access Enforcement	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Access Control	AC-4	Information Flow Enforcement	[Preview]: Storage account public access should be	3.1.0-preview ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
disallowed ↗				
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17	Remote Access	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17	Remote Access	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17 (1)	Automated Monitoring / Control	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17 (1)	Automated Monitoring / Control	Storage accounts should use private link ↗	2.0.0 ↗
Contingency Planning	CP-6	Alternate Storage Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
Contingency Planning	CP-6 (1)	Separation From Primary Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
System And Communications Protection	SC-7	Boundary Protection	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access ↗	1.1.1 ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
System And Communications Protection	SC-7	Boundary Protection	Storage accounts should use private link ↗	2.0.0 ↗
System And Communications	SC-7 (3)	Access Points	[Preview]: Storage account public access should be	3.1.0-preview ↗

Domain	Control ID	Control title	Policy	Policy version
System And Communications Protection	SC-7 (3)	Access Points	(Azure portal) Storage accounts should restrict network access	1.1.1 (GitHub)
System And Communications Protection	SC-7 (3)	Access Points	Storage accounts should restrict network access using virtual network rules	1.0.1
System And Communications Protection	SC-7 (3)	Access Points	Storage accounts should use private link	2.0.0
System And Communications Protection	SC-8	Transmission Confidentiality And Integrity	Secure transfer to storage accounts should be enabled	2.0.0
System And Communications Protection	SC-8 (1)	Cryptographic Or Alternate Physical Protection	Secure transfer to storage accounts should be enabled	2.0.0
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Storage account encryption scopes should use customer-managed keys to encrypt data at rest	1.0.0
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Storage accounts should use customer-managed key for encryption	1.0.3
System And Communications Protection	SC-28	Protection Of Information At Rest	Storage accounts should have infrastructure encryption	1.0.0
System And Communications Protection	SC-28 (1)	Cryptographic Protection	Storage accounts should have infrastructure encryption	1.0.0

NIST SP 800-53 Rev. 5

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 5](#). For more information about this compliance standard, see [NIST SP 800-53 Rev. 5](#).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-3	Access Enforcement	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Access Control	AC-4	Information Flow Enforcement	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
Access Control	AC-4	Information Flow Enforcement	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17	Remote Access	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17	Remote Access	Storage accounts should use private link ↗	2.0.0 ↗
Access Control	AC-17 (1)	Monitoring and Control	Storage accounts should restrict network access ↗	1.1.1 ↗
Access Control	AC-17 (1)	Monitoring and Control	Storage accounts should use private link ↗	2.0.0 ↗
Contingency Planning	CP-6	Alternate Storage Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
Contingency Planning	CP-6 (1)	Separation from Primary Site	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
System and Communications Protection	SC-7	Boundary Protection	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System and Communications Protection	SC-7	Boundary Protection	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications	SC-7	Boundary Protection	Storage accounts should restrict network access using	1.0.1 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection		virtual network rules ↗		
System and Communications Protection	SC-7	Boundary Protection	Storage accounts should use private link ↗	2.0.0 ↗
System and Communications Protection	SC-7 (3)	Access Points	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
System and Communications Protection	SC-7 (3)	Access Points	Storage accounts should restrict network access ↗	1.1.1 ↗
System and Communications Protection	SC-7 (3)	Access Points	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
System and Communications Protection	SC-7 (3)	Access Points	Storage accounts should use private link ↗	2.0.0 ↗
System and Communications Protection	SC-8	Transmission Confidentiality and Integrity	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
System and Communications Protection	SC-8 (1)	Cryptographic Protection	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
System and Communications Protection	SC-12	Cryptographic Key Establishment and Management	Storage account encryption scopes should use customer-managed keys to encrypt data at rest ↗	1.0.0 ↗
System and Communications Protection	SC-12	Cryptographic Key Establishment and Management	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
System and Communications Protection	SC-28	Protection of Information at Rest	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗
System and Communications Protection	SC-28 (1)	Cryptographic Protection	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗

NL BIO Cloud Theme

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for NL BIO Cloud Theme](#). For more information about this compliance standard, see [Baseline Information Security Government Cybersecurity - Digital Government \(digitaleoverheid.nl\)](#).

 [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
B.09.1 Privacy and protection of personal data - Security aspects and stages	B.09.1	Availability, integrity and confidentiality measures have been taken.	Secure transfer to storage accounts should be enabled	2.0.0
U.05.1 Data protection - Cryptographic measures	U.05.1	Data transport is secured with cryptography where key management is carried out by the CSC itself if possible.	Secure transfer to storage accounts should be enabled	2.0.0
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	Storage account encryption scopes should use customer-managed keys to encrypt data at rest	1.0.0
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	Storage accounts should have infrastructure encryption	1.0.0
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	Storage accounts should use customer-managed key for encryption	1.0.3
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	[Preview]: Storage account public access should be disallowed	3.1.0-preview
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are	Storage accounts should restrict network access	1.1.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		realized in a controlled manner.		
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	Storage accounts should restrict network access using virtual network rules	1.0.1 ↗
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	Storage accounts should use private link	2.0.0 ↗
U.10.2 Access to IT services and data - Users	U.10.2	Under the responsibility of the CSP, access is granted to administrators.	Storage accounts should be migrated to new Azure Resource Manager resources	1.0.0 ↗
U.10.3 Access to IT services and data - Users	U.10.3	Only users with authenticated equipment can access IT services and data.	Storage accounts should be migrated to new Azure Resource Manager resources	1.0.0 ↗
U.10.5 Access to IT services and data - Competent	U.10.5	Access to IT services and data is limited by technical measures and has been implemented.	Storage accounts should be migrated to new Azure Resource Manager resources	1.0.0 ↗
U.11.1 Cryptoservices - Policy	U.11.1	In the cryptography policy, at least the subjects in accordance with BIO have been elaborated.	Secure transfer to storage accounts should be enabled	2.0.0 ↗
U.11.2 Cryptoservices - Cryptographic measures	U.11.2	In case of PKoverheid certificates use PKoverheid requirements for key management. In other situations use ISO11770.	Secure transfer to storage accounts should be enabled	2.0.0 ↗
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	Storage account encryption scopes should use customer-managed keys to encrypt data at rest	1.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	Storage accounts should have infrastructure encryption	1.0.0 ↗
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	Storage accounts should use customer-managed key for encryption	1.0.3 ↗
U.12.1 Interfaces - Network connections	U.12.1	In connection points with external or untrusted zones, measures are taken against attacks.	Storage accounts should restrict network access	1.1.1 ↗
U.12.2 Interfaces - Network connections	U.12.2	Network components are such that network connections between trusted and untrusted networks are limited.	Storage accounts should restrict network access	1.1.1 ↗

NZ ISM Restricted v3.5

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NZ ISM Restricted v3.5](#). For more information about this compliance standard, see [NZ ISM Restricted v3.5](#).

 [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control and Passwords	AC-19	16.6.12 Event log protection	Storage account containing the container with activity logs must be encrypted with BYOK	1.0.0 ↗
Cryptography	CR-3	17.1.53 Reducing storage and physical transfer requirements	Storage accounts should use customer-managed key for encryption	1.0.3 ↗
Gateway security	GS-2	19.1.11 Using Gateways	Storage account keys should not be expired	3.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Gateway security	GS-3	19.1.12 Configuration of Gateways	Storage accounts should restrict network access ↗	1.1.1 ↗
Infrastructure	INF-9	10.8.35 Security Architecture	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗
Infrastructure	INF-9	10.8.35 Security Architecture	Storage accounts should use private link ↗	2.0.0 ↗
Physical Security	PS-4	8.3.5 Network infrastructure in unsecure areas	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

PCI DSS 3.2.1

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [PCI DSS 3.2.1](#). For more information about this compliance standard, see [PCI DSS 3.2.1](#) [↗](#).

[\[\]](#) [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 1	1.3.2	PCI DSS requirement 1.3.2	Storage accounts should restrict network access ↗	1.1.1 ↗
Requirement 1	1.3.4	PCI DSS requirement 1.3.4	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Requirement 1	1.3.4	PCI DSS requirement 1.3.4	Storage accounts should restrict network access ↗	1.1.1 ↗
Requirement 10	10.5.4	PCI DSS requirement 10.5.4	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Requirement 3	3.4	PCI DSS requirement 3.4	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Requirement 4	4.1	PCI DSS requirement 4.1	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 6	6.5.3	PCI DSS requirement 6.5.3	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

PCI DSS v4.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for PCI DSS v4.0](#). For more information about this compliance standard, see [PCI DSS v4.0 ↗](#).

 [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 01: Install and Maintain Network Security Controls	1.3.2	Network access to and from the cardholder data environment is restricted	Storage accounts should restrict network access ↗	1.1.1 ↗
Requirement 01: Install and Maintain Network Security Controls	1.4.2	Network connections between trusted and untrusted networks are controlled	Storage accounts should restrict network access ↗	1.1.1 ↗
Requirement 10: Log and Monitor All Access to System Components and Cardholder Data	10.2.2	Audit logs are implemented to support the detection of anomalies and suspicious activity, and the forensic analysis of events	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Requirement 10: Log and Monitor All Access to System Components and Cardholder Data	10.3.3	Audit logs are protected from destruction and unauthorized modifications	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Requirement 03: Protect Stored Account Data	3.5.1	Primary account number (PAN) is secured wherever it is stored	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Requirement 06: Develop and Maintain Secure Systems and Software	6.2.4	Bespoke and custom software are developed securely	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

Reserve Bank of India - IT Framework for NBFC

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Reserve Bank of India - IT Framework for NBFC](#). For more information about this compliance standard, see [Reserve Bank of India - IT Framework for NBFC](#).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Information and Cyber Security	3.1.g	Trails-3.1	Storage account containing the container with activity logs must be encrypted with BYOK	1.0.0
Information and Cyber Security	3.1.h	Public Key Infrastructure (PKI)-3.1	Secure transfer to storage accounts should be enabled	2.0.0
Information and Cyber Security	3.1.h	Public Key Infrastructure (PKI)-3.1	Storage account encryption scopes should use customer-managed keys to encrypt data at rest	1.0.0
Information and Cyber Security	3.1.h	Public Key Infrastructure (PKI)-3.1	Storage account encryption scopes should use double encryption for data at rest	1.0.0
Information and Cyber Security	3.1.h	Public Key Infrastructure (PKI)-3.1	Storage accounts should have infrastructure encryption	1.0.0
Information and Cyber Security	3.1.h	Public Key Infrastructure (PKI)-3.1	Storage accounts should use customer-managed key for encryption	1.0.3

Reserve Bank of India IT Framework for Banks v2016

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - RBI ITF Banks v2016](#). For more information about this compliance standard, see [RBI ITF Banks v2016 \(PDF\)](#).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Patch/Vulnerability & Change Management	Patch/Vulnerability & Change Management-7.7	Patch/Vulnerability & Change Management-7.7	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
Patch/Vulnerability & Change Management	Patch/Vulnerability & Change Management-7.7	Patch/Vulnerability & Change Management-7.7	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
Anti-Phishing		Anti-Phishing-14.1	[Preview]: Storage account public access should be disallowed ↗	3.1.0-preview ↗
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.4	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Secure Mail And Messaging Systems		Secure Mail And Messaging Systems-10.2	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Secure Mail And Messaging Systems		Secure Mail And Messaging Systems-10.1	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	Storage accounts should be migrated to new Azure Resource Manager resources ↗	1.0.0 ↗
Data Leak Prevention Strategy		Data Leak Prevention Strategy-15.2	Storage accounts should disable public network access ↗	1.0.1 ↗
Data Leak Prevention Strategy		Data Leak Prevention Strategy-15.2	Storage accounts should disable public network access ↗	1.0.1 ↗
Anti-Phishing		Anti-Phishing-14.1	Storage accounts should restrict network access ↗	1.1.1 ↗
Patch/Vulnerability & Change Management		Patch/Vulnerability & Change Management-7.7	Storage accounts should restrict network access ↗	1.1.1 ↗
Data Leak Prevention Strategy		Data Leak Prevention Strategy-15.1	Storage accounts should restrict network access ↗	1.1.1 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Data Leak Prevention Strategy	Data Leak Prevention Strategy-15.1	Storage accounts should restrict network access ↗	1.1.1 ↗	
Patch/Vulnerability & Change Management	Patch/Vulnerability & Change Management-7.7	Storage accounts should restrict network access ↗	1.1.1 ↗	
Patch/Vulnerability & Change Management	Patch/Vulnerability & Change Management-7.7	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗	
Data Leak Prevention Strategy	Data Leak Prevention Strategy-15.2	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗	
Data Leak Prevention Strategy	Data Leak Prevention Strategy-15.2	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗	
Patch/Vulnerability & Change Management	Patch/Vulnerability & Change Management-7.7	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗	
Anti-Phishing	Anti-Phishing-14.1	Storage accounts should restrict network access using virtual network rules ↗	1.0.1 ↗	
Metrics	Metrics-21.1	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗	
Advanced Real-Timethreat Defenceand Management	Advanced Real-Timethreat Defenceand Management-13.4	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗	
Metrics	Metrics-21.1	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗	

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Patch/Vulnerability & Change Management		Patch/Vulnerability & Change Management-7.7	Storage accounts should use private link ↗	2.0.0 ↗
Anti-Phishing		Anti-Phishing-14.1	Storage accounts should use private link ↗	2.0.0 ↗
Patch/Vulnerability & Change Management		Patch/Vulnerability & Change Management-7.7	Storage accounts should use private link ↗	2.0.0 ↗

RMIT Malaysia

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - RMIT Malaysia](#). For more information about this compliance standard, see [RMIT Malaysia \[↗\]\(#\)](#).

[\[\]](#) Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Cryptography	10.16	Cryptography - 10.16	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
Cryptography	10.16	Cryptography - 10.16	Storage accounts should have infrastructure encryption ↗	1.0.0 ↗
Network Resilience	10.39	Network Resilience - 10.39	Storage Accounts should use a virtual network service endpoint ↗	1.0.0 ↗
Cloud Services	10.51	Cloud Services - 10.51	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
Cloud Services	10.53	Cloud Services - 10.53	Storage account containing the container with activity logs must be encrypted with BYOK ↗	1.0.0 ↗
Cloud Services	10.53	Cloud Services - 10.53	Storage accounts should use customer-managed key for encryption ↗	1.0.3 ↗
Access Control	10.55	Access Control - 10.55	Storage accounts should allow access from trusted Microsoft	1.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			services ↗	
Cybersecurity Operations	11.5	Cybersecurity Operations - 11.5	Deploy Defender for Storage (Classic) on storage accounts ↗	1.0.1 ↗

SWIFT CSP-CSCF v2021

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for SWIFT CSP-CSCF v2021](#). For more information about this compliance standard, see [SWIFT CSP CSCF v2021](#) [↗](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
SWIFT Environment Protection	1.1	SWIFT Environment Protection	Storage accounts should restrict network access ↗	1.1.1 ↗
SWIFT Environment Protection	1.1	SWIFT Environment Protection	Storage Accounts should use a virtual network service endpoint ↗	1.0.0 ↗
Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗

SWIFT CSP-CSCF v2022

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for SWIFT CSP-CSCF v2022](#). For more information about this compliance standard, see [SWIFT CSP CSCF v2022](#) [↗](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	Storage accounts should restrict network access ↗	1.1.1 ↗
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	Storage Accounts should use a virtual network service endpoint ↗	1.0.0 ↗
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	Storage accounts should restrict network access ↗	1.1.1 ↗
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	Storage Accounts should use a virtual network service endpoint ↗	1.0.0 ↗
2. Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Geo-redundant storage should be enabled for Storage Accounts ↗	1.0.0 ↗
2. Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Secure transfer to storage accounts should be enabled ↗	2.0.0 ↗
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Storage account containing the container with activity logs must be encrypted with BYOK ↗	1.0.0 ↗

UK OFFICIAL and UK NHS

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - UK OFFICIAL](#) and [UK NHS](#). For more information about this compliance standard, see [UK OFFICIAL](#).

 [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Data in transit protection	1	Data in transit protection	Secure transfer to storage accounts should be enabled	2.0.0
Identity and authentication	10	Identity and authentication	Storage accounts should be migrated to new Azure Resource Manager resources	1.0.0
External interface protection	11	External interface protection	Storage accounts should restrict network access	1.1.1
Operational security	5.3	Protective Monitoring	Storage accounts should restrict network access	1.1.1

Next steps

- Learn more about [Azure Policy Regulatory Compliance](#).
- See the built-ins on the [Azure Policy GitHub repo](#).

Use private endpoints for Azure Storage

Article • 06/22/2023

You can use [private endpoints](#) for your Azure Storage accounts to allow clients on a virtual network (VNet) to securely access data over a [Private Link](#). The private endpoint uses a separate IP address from the VNet address space for each storage account service. Network traffic between the clients on the VNet and the storage account traverses over the VNet and a private link on the Microsoft backbone network, eliminating exposure from the public internet.

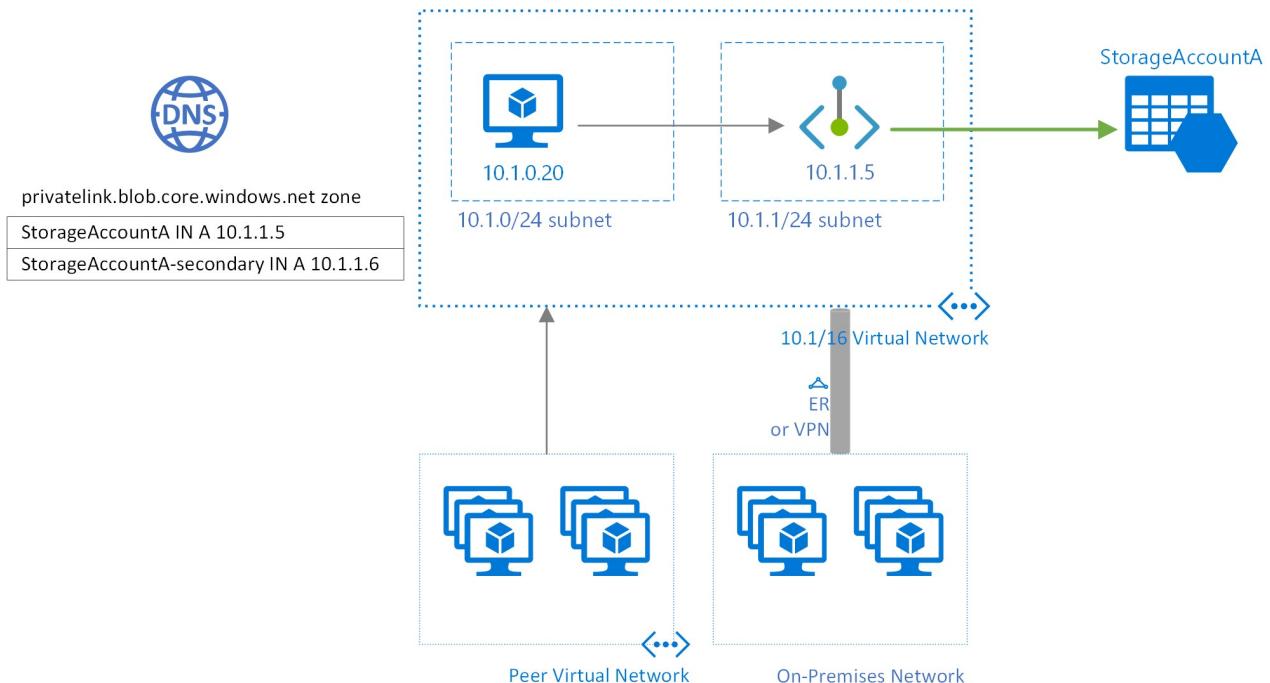
⚠ Note

Private endpoints are not available for general-purpose v1 storage accounts.

Using private endpoints for your storage account enables you to:

- Secure your storage account by configuring the storage firewall to block all connections on the public endpoint for the storage service.
- Increase security for the virtual network (VNet), by enabling you to block exfiltration of data from the VNet.
- Securely connect to storage accounts from on-premises networks that connect to the VNet using [VPN](#) or [ExpressRoutes](#) with private-peering.

Conceptual overview



A private endpoint is a special network interface for an Azure service in your [Virtual Network](#) (VNet). When you create a private endpoint for your storage account, it provides secure connectivity between clients on your VNet and your storage. The private endpoint is assigned an IP

address from the IP address range of your VNet. The connection between the private endpoint and the storage service uses a secure private link.

Applications in the VNet can connect to the storage service over the private endpoint seamlessly, **using the same connection strings and authorization mechanisms that they would use otherwise**. Private endpoints can be used with all protocols supported by the storage account, including REST and SMB.

Private endpoints can be created in subnets that use [Service Endpoints](#). Clients in a subnet can thus connect to one storage account using private endpoint, while using service endpoints to access others.

When you create a private endpoint for a storage service in your VNet, a consent request is sent for approval to the storage account owner. If the user requesting the creation of the private endpoint is also an owner of the storage account, this consent request is automatically approved.

Storage account owners can manage consent requests and the private endpoints through the '*Private endpoints*' tab for the storage account in the [Azure portal](#).

💡 Tip

If you want to restrict access to your storage account through the private endpoint only, configure the storage firewall to deny or control access through the public endpoint.

You can secure your storage account to only accept connections from your VNet by [configuring the storage firewall](#) to deny access through its public endpoint by default. You don't need a firewall rule to allow traffic from a VNet that has a private endpoint, since the storage firewall only controls access through the public endpoint. Private endpoints instead rely on the consent flow for granting subnets access to the storage service.

ⓘ Note

When copying blobs between storage accounts, your client must have network access to both accounts. So if you choose to use a private link for only one account (either the source or the destination), make sure that your client has network access to the other account. To learn about other ways to configure network access, see [Configure Azure Storage firewalls and virtual networks](#).

Creating a private endpoint

To create a private endpoint by using the Azure Portal, see [Connect privately to a storage account from the Storage Account experience in the Azure portal](#).

To create a private endpoint by using PowerShell or the Azure CLI, see either of these articles. Both of them feature an Azure web app as the target service, but the steps to create a private link are the same for an Azure Storage account.

- [Create a private endpoint using Azure CLI](#)
- [Create a private endpoint using Azure PowerShell](#)

When you create a private endpoint, you must specify the storage account and the storage service to which it connects.

You need a separate private endpoint for each storage resource that you need to access, namely [Blobs](#), [Data Lake Storage Gen2](#), [Files](#), [Queues](#), [Tables](#), or [Static Websites](#). On the private endpoint, these storage services are defined as the **target sub-resource** of the associated storage account.

If you create a private endpoint for the Data Lake Storage Gen2 storage resource, then you should also create one for the Blob Storage resource. That's because operations that target the Data Lake Storage Gen2 endpoint might be redirected to the Blob endpoint. Similarly, if you add a private endpoint for Blob Storage only, and not for Data Lake Storage Gen2, some operations (such as Manage ACL, Create Directory, Delete Directory, etc.) will fail since the Gen2 APIs require a DFS private endpoint. By creating a private endpoint for both resources, you ensure that all operations can complete successfully.

Tip

Create a separate private endpoint for the secondary instance of the storage service for better read performance on RA-GRS accounts. Make sure to create a general-purpose v2(Standard or Premium) storage account.

For read access to the secondary region with a storage account configured for geo-redundant storage, you need separate private endpoints for both the primary and secondary instances of the service. You don't need to create a private endpoint for the secondary instance for **failover**. The private endpoint will automatically connect to the new primary instance after failover. For more information about storage redundancy options, see [Azure Storage redundancy](#).

Connecting to a private endpoint

Clients on a VNet using the private endpoint should use the same connection string for the storage account as clients connecting to the public endpoint. We rely upon DNS resolution to automatically route the connections from the VNet to the storage account over a private link.

Important

Use the same connection string to connect to the storage account using private endpoints as you'd use otherwise. Please don't connect to the storage account using its `privatelink` subdomain URL.

By default, We create a [private DNS zone](#) attached to the VNet with the necessary updates for the private endpoints. However, if you're using your own DNS server, you may need to make additional

changes to your DNS configuration. The section on [DNS changes](#) below describes the updates required for private endpoints.

DNS changes for private endpoints

Note

For details on how to configure your DNS settings for private endpoints, see [Azure Private Endpoint DNS configuration](#).

When you create a private endpoint, the DNS CNAME resource record for the storage account is updated to an alias in a subdomain with the prefix `privatelink`. By default, we also create a [private DNS zone](#), corresponding to the `privatelink` subdomain, with the DNS A resource records for the private endpoints.

When you resolve the storage endpoint URL from outside the VNet with the private endpoint, it resolves to the public endpoint of the storage service. When resolved from the VNet hosting the private endpoint, the storage endpoint URL resolves to the private endpoint's IP address.

For the illustrated example above, the DNS resource records for the storage account 'StorageAccountA', when resolved from outside the VNet hosting the private endpoint, will be:

Name	Type	Value
<code>StorageAccountA.blob.core.windows.net</code>	CNAME	<code>StorageAccountA.privatelink.blob.core.windows.net</code>
<code>StorageAccountA.privatelink.blob.core.windows.net</code>	CNAME	<storage service public endpoint>
<storage service public endpoint>	A	<storage service public IP address>

As previously mentioned, you can deny or control access for clients outside the VNet through the public endpoint using the storage firewall.

The DNS resource records for StorageAccountA, when resolved by a client in the VNet hosting the private endpoint, will be:

Name	Type	Value
<code>StorageAccountA.blob.core.windows.net</code>	CNAME	<code>StorageAccountA.privatelink.blob.core.windows.net</code>
<code>StorageAccountA.privatelink.blob.core.windows.net</code>	A	10.1.1.5

This approach enables access to the storage account **using the same connection string** for clients on the VNet hosting the private endpoints, as well as clients outside the VNet.

If you are using a custom DNS server on your network, clients must be able to resolve the FQDN for the storage account endpoint to the private endpoint IP address. You should configure your DNS server to delegate your private link subdomain to the private DNS zone for the VNet, or configure

the A records for `StorageAccountA.privatelink.blob.core.windows.net` with the private endpoint IP address.

💡 Tip

When using a custom or on-premises DNS server, you should configure your DNS server to resolve the storage account name in the `privatelink` subdomain to the private endpoint IP address. You can do this by delegating the `privatelink` subdomain to the private DNS zone of the VNet or by configuring the DNS zone on your DNS server and adding the DNS A records.

The recommended DNS zone names for private endpoints for storage services, and the associated endpoint target sub-resources, are:

Storage service	Target sub-resource	Zone name
Blob service	blob	<code>privatelink.blob.core.windows.net</code>
Data Lake Storage Gen2	dfs	<code>privatelink.dfs.core.windows.net</code>
File service	file	<code>privatelink.file.core.windows.net</code>
Queue service	queue	<code>privatelink.queue.core.windows.net</code>
Table service	table	<code>privatelink.table.core.windows.net</code>
Static Websites	web	<code>privatelink.web.core.windows.net</code>

For more information on configuring your own DNS server to support private endpoints, refer to the following articles:

- [Name resolution for resources in Azure virtual networks](#)
- [DNS configuration for private endpoints](#)

Pricing

For pricing details, see [Azure Private Link pricing](#).

Known Issues

Keep in mind the following known issues about private endpoints for Azure Storage.

Storage access constraints for clients in VNets with private endpoints

Clients in VNets with existing private endpoints face constraints when accessing other storage accounts that have private endpoints. For example, suppose a VNet N1 has a private endpoint for a storage account A1 for Blob storage. If storage account A2 has a private endpoint in a VNet N2 for

Blob storage, then clients in VNet N1 must also access Blob storage in account A2 using a private endpoint. If storage account A2 does not have any private endpoints for Blob storage, then clients in VNet N1 can access Blob storage in that account without a private endpoint.

This constraint is a result of the DNS changes made when account A2 creates a private endpoint.

Copying blobs between storage accounts

You can copy blobs between storage accounts by using private endpoints only if you use the Azure REST API, or tools that use the REST API. These tools include AzCopy, Storage Explorer, Azure PowerShell, Azure CLI, and the Azure Blob Storage SDKs.

Only private endpoints that target the `blob` storage resource endpoint are supported. This includes REST API calls against Data Lake Storage Gen2 accounts in which the `blob` resource endpoint is referenced explicitly or implicitly. Private endpoints that target the Data Lake Storage Gen2 `dfs` endpoint or the `file` resource endpoint are not yet supported. Copying between storage accounts by using the Network File System (NFS) protocol is not yet supported.

Next steps

- [Azure Private Endpoint DNS configuration](#)
- [Configure Azure Storage firewalls and virtual networks](#)
- [Security recommendations for Blob storage](#)

Azure security baseline for Storage

Article • 09/20/2023

This security baseline applies guidance from the [Microsoft cloud security benchmark version 1.0](#) to Storage. The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the security controls defined by the Microsoft cloud security benchmark and the related guidance applicable to Storage.

You can monitor this security baseline and its recommendations using Microsoft Defender for Cloud. Azure Policy definitions will be listed in the Regulatory Compliance section of the Microsoft Defender for Cloud portal page.

When a feature has relevant Azure Policy Definitions, they are listed in this baseline to help you measure compliance with the Microsoft cloud security benchmark controls and recommendations. Some recommendations may require a paid Microsoft Defender plan to enable certain security scenarios.

ⓘ Note

Features not applicable to Storage have been excluded. To see how Storage completely maps to the Microsoft cloud security benchmark, see the [full Storage security baseline mapping file](#).

Security profile

The security profile summarizes high-impact behaviors of Storage, which may result in increased security considerations.

Service Behavior Attribute	Value
Product Category	Storage
Customer can access HOST / OS	No Access
Service can be deployed into customer's virtual network	True
Stores customer content at rest	True

Network security

For more information, see the [Microsoft cloud security benchmark: Network security](#).

NS-1: Establish network segmentation boundaries

Features

Virtual Network Integration

Description: Service supports deployment into customer's private Virtual Network (VNet). [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

NS-2: Secure cloud services with network controls

Features

Azure Private Link

Description: Service native IP filtering capability for filtering network traffic (not to be confused with NSG or Azure Firewall). [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Deploy private endpoints for Azure Storage to establish a private access point for the resources.

Reference: [Use private endpoints for Azure Storage](#)

Disable Public Network Access

Description: Service supports disabling public network access either through using service-level IP ACL filtering rule (not NSG or Azure Firewall) or using a 'Disable Public Network Access' toggle switch. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Disable public network access by either using Azure Storage service-level IP ACL filtering or a toggling switch for public network access.

Reference: [Change the default network access rule](#)

Identity management

For more information, see the [Microsoft cloud security benchmark: Identity management](#).

IM-1: Use centralized identity and authentication system

Features

Azure AD Authentication Required for Data Plane Access

Description: Service supports using Azure AD authentication for data plane access.

[Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: Storage offers multiple ways to authorize to the data plane. Azure provides Azure role-based access control (Azure RBAC) for fine-grained control over a client's access to resources in a storage account. Use Azure AD credentials when possible as a security best practice, rather than using the account key, which can be more easily compromised. When your application design requires shared access signatures for access to Blob storage, use Azure AD credentials to create user delegation shared access signatures (SAS) when possible for superior security.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Authorize access to data in Azure Storage](#)

Local Authentication Methods for Data Plane Access

Description: Local authentications methods supported for data plane access, such as a local username and password. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: Restrict the use of local authentication methods for data plane access. Instead, use Azure Active Directory (Azure AD) as the default authentication method to control your data plane access.

Reference: [SFTP permission model](#)

IM-3: Manage application identities securely and automatically

Features

Managed Identities

Description: Data plane actions support authentication using managed identities. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure managed identities instead of service principals when possible, which can authenticate to Azure services and resources that support Azure Active Directory (Azure AD) authentication. Managed identity credentials are fully managed, rotated, and protected by the platform, avoiding hard-coded credentials in source code or configuration files.

Reference: [Authorize access to blob data with managed identities for Azure resources](#)

Service Principals

Description: Data plane supports authentication using service principals. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Additional Guidance: With Azure AD, you can use Azure role-based access control (Azure RBAC) to grant permissions to a security principal, which may be a user, group, or application service principal. The security principal is authenticated by Azure AD to return an OAuth 2.0 token. The token can then be used to authorize a request against the Blob service.

Reference: [Authorize access to blobs using Azure Active Directory](#)

IM-7: Restrict resource access based on conditions

Features

Conditional Access for Data Plane

Description: Data plane access can be controlled using Azure AD Conditional Access Policies. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Define the applicable conditions and criteria for Azure Active Directory (Azure AD) conditional access in the workload. Consider common use cases such as blocking or granting access from specific locations, blocking risky sign-in behavior, or requiring organization-managed devices for specific applications.

Reference: [Disallow Shared Key authorization to use Azure AD Conditional Access](#)

IM-8: Restrict the exposure of credential and secrets

Features

Service Credential and Secrets Support Integration and Storage in Azure Key Vault

Description: Data plane supports native use of Azure Key Vault for credential and secrets store. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Ensure that secrets and credentials are stored in secure locations such as Azure Key Vault, instead of embedding them into code or configuration files.

Reference: [Manage storage account keys with Key Vault and the Azure CLI](#)

Privileged access

For more information, see the [Microsoft cloud security benchmark: Privileged access](#).

PA-1: Separate and limit highly privileged/administrative users

Features

Local Admin Accounts

Description: Service has the concept of a local administrative account. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

PA-7: Follow just enough administration (least privilege) principle

Features

Azure RBAC for Data Plane

Description: Azure Role-Based Access Control (Azure RBAC) can be used to manage access to service's data plane actions. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Storage supports using Azure Active Directory (Azure AD) to authorize requests to blob data. With Azure AD, you can use Azure role-based access control (Azure RBAC) to grant permissions to a security principal, which may be a user, group, or application service principal.

Authorizing requests against Azure Storage with Azure AD provides superior security and ease of use over Shared Key authorization. Microsoft recommends using Azure AD authorization with your blob applications when possible to assure access with minimum required privileges.

Reference: [Authorize access to blobs using Azure Active Directory](#)

PA-8: Determine access process for cloud provider support

Features

Customer Lockbox

Description: Customer Lockbox can be used for Microsoft support access. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: In support scenarios where Microsoft needs to access your data, use Customer Lockbox to review, then approve or reject each of Microsoft's data access requests.

Reference: [Customer Lockbox](#)

Data protection

For more information, see the [Microsoft cloud security benchmark: Data protection](#).

DP-1: Discover, classify, and label sensitive data

Features

Sensitive Data Discovery and Classification

Description: Tools (such as Azure Purview or Azure Information Protection) can be used for data discovery and classification in the service. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Storage integration with Azure purview is currently in private preview.

Configuration Guidance: Use Azure Purview to scan, classify and label any sensitive data that resides in Azure Storage.

Reference: [Connect to Azure Blob storage in Microsoft Purview](#)

DP-2: Monitor anomalies and threats targeting sensitive data

Features

Data Leakage/Loss Prevention

Description: Service supports DLP solution to monitor sensitive data movement (in customer's content). [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Defender for Storage continually analyzes the telemetry stream generated by the Azure Blob Storage and Azure Files services. When potentially malicious activities are detected, security alerts are generated. These alerts are displayed in Microsoft Defender for Cloud, together with the details of the suspicious activity along with the relevant investigation steps, remediation actions, and security recommendations.

Microsoft Defender for Storage is built into Microsoft Defender for Cloud. When you enable Microsoft Defender for Cloud's enhanced security features on your subscription, Microsoft Defender for Storage is automatically enabled for all of your storage accounts.

You may enable or disable Defender for Storage for individual storage accounts under a specific subscription.

Reference: [Configure Microsoft Defender for Storage](#)

DP-3: Encrypt sensitive data in transit

Features

Data in Transit Encryption

Description: Service supports data in-transit encryption for data plane. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Enforce a minimum required version of Transport Layer Security \(TLS\) for requests to a storage account](#)

DP-4: Enable data at rest encryption by default

Features

Data at Rest Encryption Using Platform Keys

Description: Data at-rest encryption using platform keys is supported, any customer content at rest is encrypted with these Microsoft managed keys. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Azure Storage encryption for data at rest](#)

DP-5: Use customer-managed key option in data at rest encryption when required

Features

Data at Rest Encryption Using CMK

Description: Data at-rest encryption using customer-managed keys is supported for customer content stored by the service. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: If required for regulatory compliance, define the use case and service scope where encryption using customer-managed keys are needed. Enable and implement data at rest encryption for the in-scope data using customer-managed key for Azure Storage

Reference: [Customer-managed keys for Azure Storage encryption](#)

DP-6: Use a secure key management process

Features

Key Management in Azure Key Vault

Description: The service supports Azure Key Vault integration for any customer keys, secrets, or certificates. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Key Vault to create and control the life cycle of your encryption keys, including key generation, distribution, and storage. Rotate and revoke your keys in Azure Key Vault and your service based on a defined schedule or when there is a key retirement or compromise. When there is a need to use customer-managed key (CMK) in the workload, service, or application level, ensure you follow the best practices for key management: Use a key hierarchy to generate a separate data encryption key (DEK) with your key encryption key (KEK) in your key vault. Ensure keys

are registered with Azure Key Vault and referenced via key IDs from the service or application. If you need to bring your own key (BYOK) to the service (such as importing HSM-protected keys from your on-premises HSMs into Azure Key Vault), follow recommended guidelines to perform initial key generation and key transfer.

Reference: [Manage storage account keys with Key Vault and the Azure CLI](#)

Asset management

For more information, see the [Microsoft cloud security benchmark: Asset management](#).

AM-2: Use only approved services

Features

Azure Policy Support

Description: Service configurations can be monitored and enforced via Azure Policy.
[Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Define and implement standard security configurations for network resources associated with your Azure Storage Account with Azure Policy. Use Azure Policy aliases in the "Microsoft.Storage" and "Microsoft.Network" namespaces to create custom policies to audit or enforce the network configuration of your Storage account resources.

You may also make use of built-in policy definitions related to Storage account, such as: Storage Accounts should use a virtual network service endpoint

Reference: [Azure Policy built-in definitions for Azure Storage](#)

Logging and threat detection

For more information, see the [Microsoft cloud security benchmark: Logging and threat detection](#).

LT-1: Enable threat detection capabilities

Features

Microsoft Defender for Service / Product Offering

Description: Service has an offering-specific Microsoft Defender solution to monitor and alert on security issues. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Microsoft Defender for Storage to provide an additional layer of security intelligence that detects unusual and potentially harmful attempts to access or exploit storage accounts. It uses advanced threat detection capabilities and Microsoft Threat Intelligence data to provide contextual security alerts. Those alerts also include steps to mitigate the detected threats and prevent future attacks.

Reference: [Introduction to Microsoft Defender for Storage](#)

LT-4: Enable logging for security investigation

Features

Azure Resource Logs

Description: Service produces resource logs that can provide enhanced service-specific metrics and logging. The customer can configure these resource logs and send them to their own data sink like a storage account or log analytics workspace. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Ingest logs via Azure Monitor to aggregate security data generated by endpoints devices, network resources, and other security systems. Within Azure Monitor, use Log Analytics Workspace(s) to query and perform analytics, and use Azure Storage Accounts for long-term/archival storage, optionally with security features such as immutable storage and enforced retention holds.

Backup and recovery

For more information, see the [Microsoft cloud security benchmark: Backup and recovery](#).

BR-1: Ensure regular automated backups

Features

Azure Backup

Description: The service can be backed up by the Azure Backup service. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Azure Backup is currently only supported for Azure Blob storage. Queue and table data can be backed up by using the AzCopy command line tool.

Configuration Guidance: Enable Azure Backup and configure the backup source on a desired frequency and with a desired retention period. Azure Backup lets you easily configure operational backup for protecting block blobs in your storage accounts. Backup of blobs is configured at the storage account level. So, all blobs in the storage account are protected with operational backup.

You can configure backup for multiple storage accounts using the Backup Center. You can also configure backup for a storage account using the storage account's Data Protection properties.

Reference: [Overview of operational backup for Azure Blobs](#)

Service Native Backup Capability

Description: Service supports its own native backup capability (if not using Azure Backup). [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Additional Guidance: Operational backup of blobs is a local backup solution. So the backup data isn't transferred to the Backup vault, but is stored in the source storage account itself. However, the Backup vault still serves as the unit of managing backups. Also, this is a continuous backup solution, which means that you don't need to schedule any backups and all changes will be retained and restorable from the state at a selected point in time.

Reference: [Overview of operational backup for Azure Blobs](#)

Next steps

- See the [Microsoft cloud security benchmark overview](#)
- Learn more about [Azure security baselines](#)

Azure Storage redundancy

Article • 01/08/2024

Azure Storage always stores multiple copies of your data so that it's protected from planned and unplanned events, including transient hardware failures, network or power outages, and massive natural disasters. Redundancy ensures that your storage account meets its availability and durability targets even in the face of failures.

When deciding which redundancy option is best for your scenario, consider the tradeoffs between lower costs and higher availability. The factors that help determine which redundancy option you should choose include:

- How your data is replicated within the primary region.
- Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters (geo-replication).
- Whether your application requires read access to the replicated data in the secondary region if the primary region becomes unavailable for any reason (geo-replication with read access).

ⓘ Note

The features and regional availability described in this article are also available to accounts that have a hierarchical namespace (Azure Blob storage).

The services that comprise Azure Storage are managed through a common Azure resource called a *storage account*. The storage account represents a shared pool of storage that can be used to deploy storage resources such as blob containers (Blob Storage), file shares (Azure Files), tables (Table Storage), or queues (Queue Storage). For more information about Azure Storage accounts, see [Storage account overview](#).

The redundancy setting for a storage account is shared for all storage services exposed by that account. All storage resources deployed in the same storage account have the same redundancy setting. You may want to isolate different types of resources in separate storage accounts if they have different redundancy requirements.

Redundancy in the primary region

Data in an Azure Storage account is always replicated three times in the primary region. Azure Storage offers two options for how your data is replicated in the primary region:

- **Locally redundant storage (LRS)** copies your data synchronously three times within a single physical location in the primary region. LRS is the least expensive replication option, but isn't recommended for applications requiring high availability or durability.
- **Zone-redundant storage (ZRS)** copies your data synchronously across three Azure availability zones in the primary region. For applications requiring high availability, Microsoft recommends using ZRS in the primary region, and also replicating to a secondary region.

 **Note**

Microsoft recommends using ZRS in the primary region for Azure Data Lake Storage Gen2 workloads.

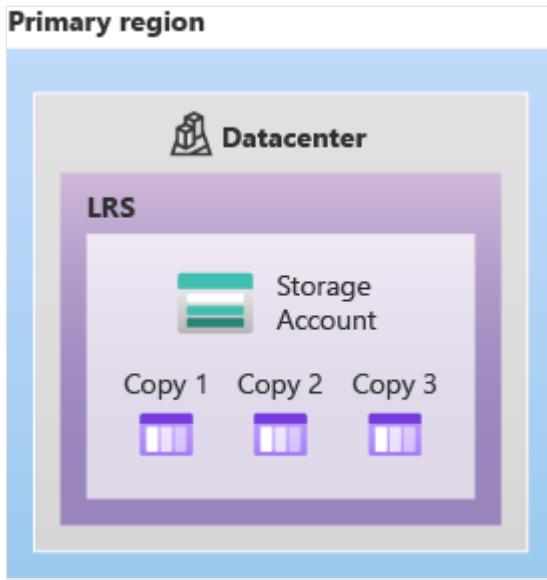
Locally redundant storage

Locally redundant storage (LRS) replicates your storage account three times within a single data center in the primary region. LRS provides at least 99.99999999% (11 nines) durability of objects over a given year.

LRS is the lowest-cost redundancy option and offers the least durability compared to other options. LRS protects your data against server rack and drive failures. However, if a disaster such as fire or flooding occurs within the data center, all replicas of a storage account using LRS may be lost or unrecoverable. To mitigate this risk, Microsoft recommends using [zone-redundant storage \(ZRS\)](#), [geo-redundant storage \(GRS\)](#), or [geo-zone-redundant storage \(GZRS\)](#).

A write request to a storage account that is using LRS happens synchronously. The write operation returns successfully only after the data is written to all three replicas.

The following diagram shows how your data is replicated within a single data center with LRS:



LRS is a good choice for the following scenarios:

- If your application stores data that can be easily reconstructed if data loss occurs, you may opt for LRS.
- If your application is restricted to replicating data only within a country or region due to data governance requirements, you may opt for LRS. In some cases, the paired regions across which the data is geo-replicated may be in another country or region. For more information on paired regions, see [Azure regions](#).
- If your scenario is using Azure unmanaged disks, you may opt for LRS. While it's possible to create a storage account for Azure unmanaged disks that uses GRS, it isn't recommended due to potential issues with consistency over asynchronous geo-replication.

Zone-redundant storage

Zone-redundant storage (ZRS) replicates your storage account synchronously across three Azure availability zones in the primary region. Each availability zone is a separate physical location with independent power, cooling, and networking. ZRS offers durability for storage resources of at least 99.9999999999% (12 9's) over a given year.

With ZRS, your data is still accessible for both read and write operations even if a zone becomes unavailable. If a zone becomes unavailable, Azure undertakes networking updates, such as DNS repointing. These updates may affect your application if you access data before the updates have completed. When designing applications for ZRS, follow practices for transient fault handling, including implementing retry policies with exponential back-off.

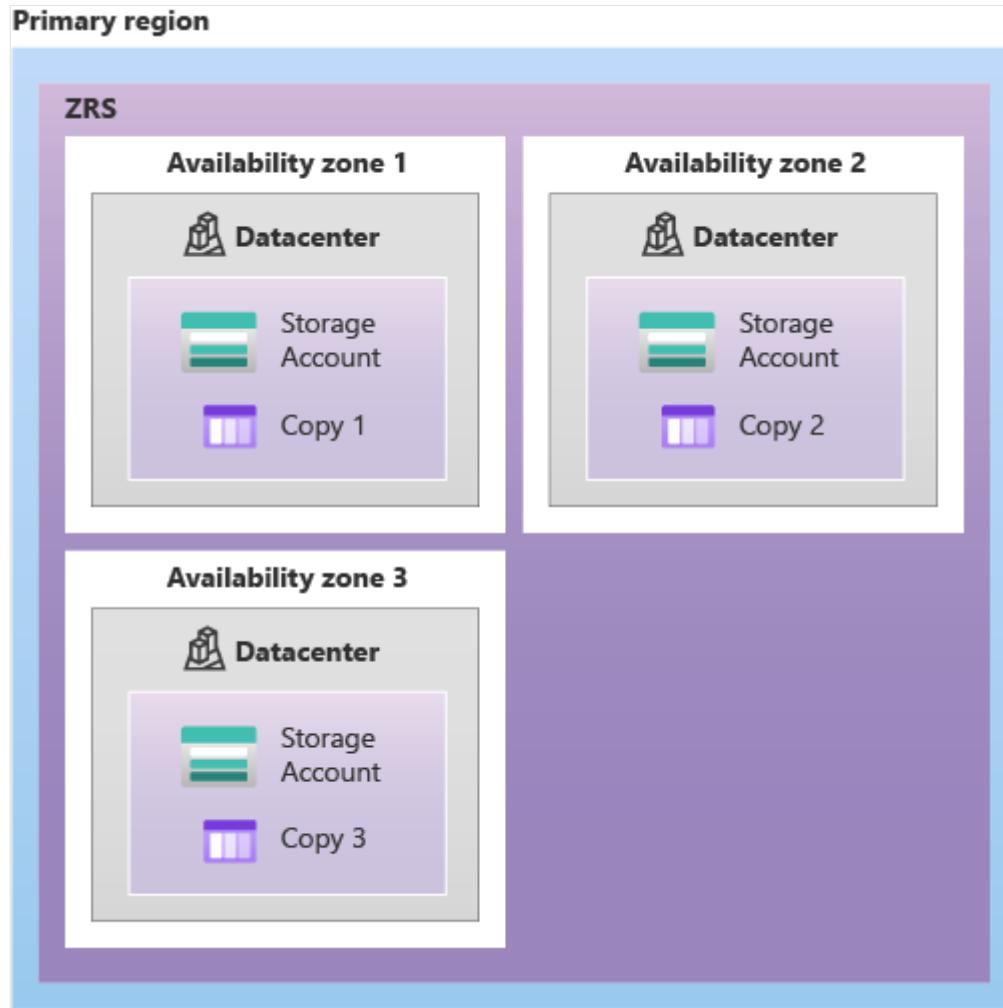
A write request to a storage account that is using ZRS happens synchronously. The write operation returns successfully only after the data is written to all replicas across the

three availability zones. If an availability zone is temporarily unavailable, the operation returns successfully after the data is written to all available zones.

Microsoft recommends using ZRS in the primary region for scenarios that require high availability. ZRS is also recommended for restricting replication of data to a particular country or region to meet data governance requirements.

Microsoft recommends using ZRS for Azure Files workloads. If a zone becomes unavailable, no remounting of Azure file shares from the connected clients is required.

The following diagram shows how your data is replicated across availability zones in the primary region with ZRS:



ZRS provides excellent performance, low latency, and resiliency for your data if it becomes temporarily unavailable. However, ZRS by itself may not protect your data against a regional disaster where multiple zones are permanently affected. For protection against regional disasters, Microsoft recommends using [geo-zone-redundant storage](#) (GZRS), which uses ZRS in the primary region and also geo-replicates your data to a secondary region.

The archive tier for Blob Storage isn't currently supported for ZRS, GZRS, or RA-GZRS accounts. Unmanaged disks don't support ZRS or GZRS.

For more information about which regions support ZRS, see [Azure regions with availability zones](#).

Standard storage accounts

ZRS is supported for all Azure Storage services through standard general-purpose v2 storage accounts, including:

- Azure Blob storage (hot and cool block blobs and append blobs, non-disk page blobs)
- Azure Files (all standard tiers: transaction optimized, hot, and cool)
- Azure Table storage
- Azure Queue storage

For a list of regions that support zone-redundant storage (ZRS) for standard accounts, see [Azure regions that support zone-redundant storage \(ZRS\) for standard storage accounts](#).

Premium block blob accounts

ZRS is supported for premium block blobs accounts. For more information about premium block blobs, see [Premium block blob storage accounts](#).

For a list of regions that support zone-redundant storage (ZRS) for premium block blobs accounts, see [Azure regions that support zone-redundant storage \(ZRS\) for premium block blob accounts](#).

Premium file share accounts

ZRS is supported for premium file shares (Azure Files) through the `FileStorage` storage account kind.

For a list of regions that support zone-redundant storage (ZRS) for premium file share accounts, see [Azure Files zone-redundant storage for premium file shares](#).

Managed disks

ZRS is supported for managed disks with the following [limitations](#).

For a list of regions that support zone-redundant storage (ZRS) for managed disks, see [regional availability](#).

Redundancy in a secondary region

For applications requiring high durability, you can choose to additionally copy the data in your storage account to a secondary region that is hundreds of miles away from the primary region. If your storage account is copied to a secondary region, then your data is durable even in the case of a complete regional outage or a disaster in which the primary region isn't recoverable.

When you create a storage account, you select the primary region for the account. The paired secondary region is determined based on the primary region, and can't be changed. For more information about regions supported by Azure, see [Azure regions](#).

Azure Storage offers two options for copying your data to a secondary region:

- **Geo-redundant storage (GRS)** copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.
- **Geo-zone-redundant storage (GZRS)** copies your data synchronously across three Azure availability zones in the primary region using ZRS. It then copies your data asynchronously to a single physical location in the secondary region. Within the secondary region, your data is copied synchronously three times using LRS.

ⓘ Note

The primary difference between GRS and GZRS is how data is replicated in the primary region. Within the secondary region, data is always replicated synchronously three times using LRS. LRS in the secondary region protects your data against hardware failures.

With GRS or GZRS, the data in the secondary region isn't available for read or write access unless there's a failover to the primary region. For read access to the secondary region, configure your storage account to use read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS). For more information, see [Read access to data in the secondary region](#).

If the primary region becomes unavailable, you can choose to fail over to the secondary region. After the failover has completed, the secondary region becomes the primary region, and you can again read and write data. For more information on disaster recovery and to learn how to fail over to the secondary region, see [Disaster recovery and storage account failover](#).

ⓘ Important

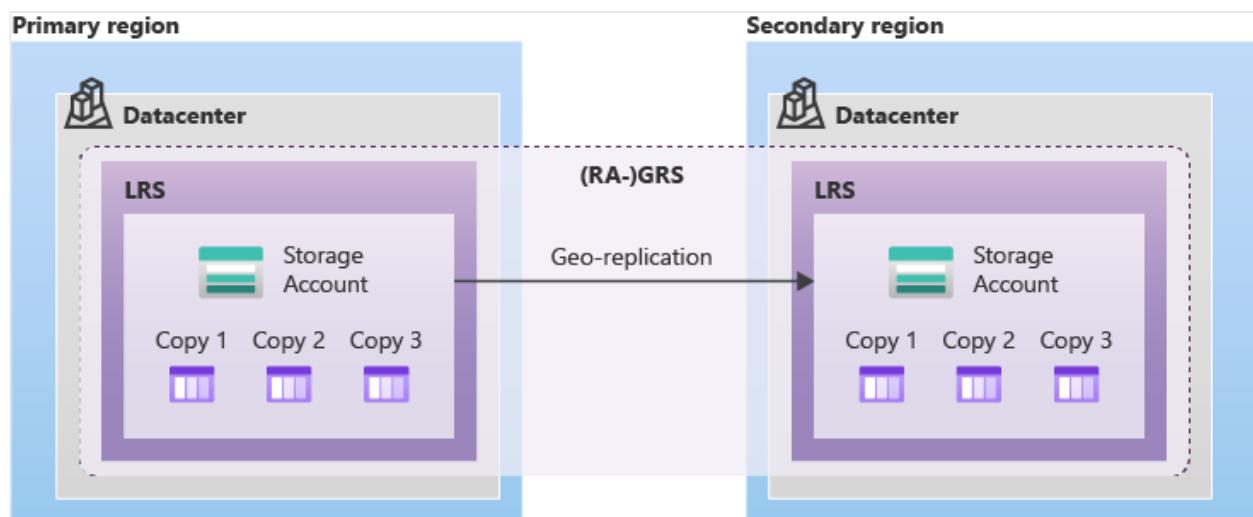
Because data is replicated to the secondary region asynchronously, a failure that affects the primary region may result in data loss if the primary region cannot be recovered. The interval between the most recent writes to the primary region and the last write to the secondary region is known as the recovery point objective (RPO). The RPO indicates the point in time to which data can be recovered. The Azure Storage platform typically has an RPO of less than 15 minutes, although there's currently no SLA on how long it takes to replicate data to the secondary region.

Geo-redundant storage

Geo-redundant storage (GRS) copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in a secondary region that is hundreds of miles away from the primary region. GRS offers durability for storage resources of at least 99.9999999999999% (16 9's) over a given year.

A write operation is first committed to the primary location and replicated using LRS. The update is then replicated asynchronously to the secondary region. When data is written to the secondary location, it's also replicated within that location using LRS.

The following diagram shows how your data is replicated with GRS or RA-GRS:



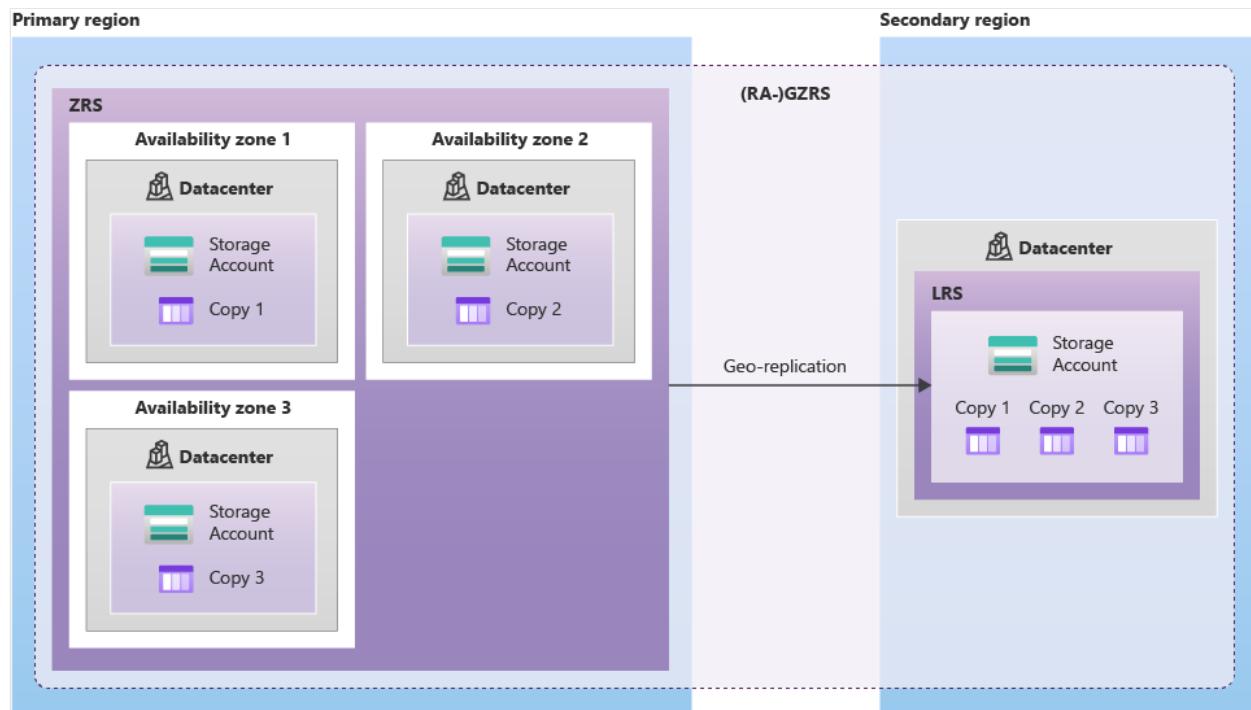
Geo-zone-redundant storage

Geo-zone-redundant storage (GZRS) combines the high availability provided by redundancy across availability zones with protection from regional outages provided by

geo-replication. Data in a GZRS storage account is copied across three [Azure availability zones](#) in the primary region and is also replicated to a secondary geographic region for protection from regional disasters. Microsoft recommends using GZRS for applications requiring maximum consistency, durability, and availability, excellent performance, and resilience for disaster recovery.

With a GZRS storage account, you can continue to read and write data if an availability zone becomes unavailable or is unrecoverable. Additionally, your data is also durable in the case of a complete regional outage or a disaster in which the primary region isn't recoverable. GZRS is designed to provide at least 99.9999999999999% (16 9's) durability of objects over a given year.

The following diagram shows how your data is replicated with GZRS or RA-GZRS:



Only standard general-purpose v2 storage accounts support GZRS. GZRS is supported by all of the Azure Storage services, including:

- Azure Blob storage (hot and cool block blobs, non-disk page blobs)
- Azure Files (all standard tiers: transaction optimized, hot, and cool)
- Azure Table storage
- Azure Queue storage

For a list of regions that support geo-zone-redundant storage (GZRS), see [Azure regions that support geo-zone-redundant storage \(GZRS\)](#).

Read access to data in the secondary region

Geo-redundant storage (with GRS or GZRS) replicates your data to another physical location in the secondary region to protect against regional outages. With an account configured for GRS or GZRS, data in the secondary region is not directly accessible to users or applications, unless a failover occurs. The failover process updates the DNS entry provided by Azure Storage so that the secondary endpoint becomes the new primary endpoint for your storage account. During the failover process, your data is inaccessible. After the failover is complete, you can read and write data to the new primary region. For more information, see [How customer-managed storage account failover works](#).

If your applications require high availability, then you can configure your storage account for read access to the secondary region. When you enable read access to the secondary region, then your data is always available to be read from the secondary, including in a situation where the primary region becomes unavailable. Read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS) configurations permit read access to the secondary region.

 **Note**

Azure Files does not support read-access geo-redundant storage (RA-GRS) or read-access geo-zone-redundant storage (RA-GZRS).

Design your applications for read access to the secondary

If your storage account is configured for read access to the secondary region, then you can design your applications to seamlessly shift to reading data from the secondary region if the primary region becomes unavailable for any reason.

The secondary region is available for read access after you enable RA-GRS or RA-GZRS, so that you can test your application in advance to make sure that it will properly read from the secondary in the event of an outage. For more information about how to design your applications to take advantage of geo-redundancy, see [Use geo-redundancy to design highly available applications](#).

When read access to the secondary is enabled, your application can be read from the secondary endpoint as well as from the primary endpoint. The secondary endpoint appends the suffix `-secondary` to the account name. For example, if your primary endpoint for Blob storage is `myaccount.blob.core.windows.net`, then the secondary endpoint is `myaccount-secondary.blob.core.windows.net`. The account access keys for your storage account are the same for both the primary and secondary endpoints.

Plan for data loss

Because data is replicated asynchronously from the primary to the secondary region, the secondary region is typically behind the primary region in terms of write operations. If a disaster were to strike the primary region, it's likely that some data would be lost and that files within a directory or container would not be consistent. For more information about how to plan for potential data loss, see [Data loss and inconsistencies](#).

Summary of redundancy options

The tables in the following sections summarize the redundancy options available for Azure Storage.

Durability and availability parameters

The following table describes key parameters for each redundancy option:

Expand table

Parameter	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
Percent durability of objects over a given year	at least 99.999999999% (11 9's)	at least 99.999999999% (12 9's)	at least 99.9999999999999% (16 9's)	at least 99.9999999999999% (16 9's)
Availability for read requests	At least 99.9% (99% for cool or archive access tiers)	At least 99.9% (99% for cool or archive access tier)	At least 99.9% (99% for cool or archive access tiers) for GRS	At least 99.9% (99% for cool access tier) for GZRS
			At least 99.99% (99.9% for cool or archive access tiers) for RA-GRS	At least 99.99% (99.9% for cool access tier) for RA-GZRS
Availability for write requests	At least 99.9% (99% for cool or archive access tiers)	At least 99.9% (99% for cool or archive access tier)	At least 99.9% (99% for cool or archive access tiers)	At least 99.9% (99% for cool access tier)
Number of copies of data maintained	Three copies within a single region	Three copies across separate availability zones within a single region	Six copies total, including three in the primary region and three in the secondary region	Six copies total, including three across separate availability zones in the primary region

Parameter	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
separate nodes				and three locally redundant copies in the secondary region

For more information, see the [SLA for Storage Accounts](#) .

Durability and availability by outage scenario

The following table indicates whether your data is durable and available in a given scenario, depending on which type of redundancy is in effect for your storage account:

 [Expand table](#)

Outage scenario	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
A node within a data center becomes unavailable	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes ¹	Yes
A region-wide outage occurs in the primary region	No	No	Yes ¹	Yes ¹
Read access to the secondary region is available if the primary region becomes unavailable	No	No	Yes (with RA-GRS)	Yes (with RA-GZRS)

¹ Account failover is required to restore write availability if the primary region becomes unavailable. For more information, see [Disaster recovery and storage account failover](#).

Supported Azure Storage services

The following table shows which redundancy options are supported by each Azure Storage service.

 [Expand table](#)

Service	LRS	ZRS	GRS	RA-GRS	GZRS	RA-GZRS
Blob storage (including Data Lake Storage)	✓	✓	✓	✓	✓	✓
Queue storage	✓	✓	✓	✓	✓	✓
Table storage	✓	✓	✓	✓	✓	✓

Service	LRS	ZRS	GRS	RA-GRS	GZRS	RA-GZRS
Azure Files	✓ 1,2	✓ 1,2	✓ 1		✓ 1	
Azure managed disks	✓	✓ 3				
Azure Elastic SAN	✓	✓				

¹ Standard file shares are supported on LRS and ZRS. Standard file shares are supported on GRS and GZRS as long as they're less than or equal to 5 TiB in size.

² Premium file shares are supported on LRS and ZRS.

³ ZRS managed disks have certain limitations. See the [Limitations](#) section of the redundancy options for managed disks article for details.

Supported storage account types

The following table shows which redundancy options are supported for each type of storage account. For information for storage account types, see [Storage account overview](#).

Expand table

Storage account types	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
Recommended	Standard general-purpose v2 <code>(StorageV2)</code> ¹ Premium block blobs <code>(BlockBlobStorage)</code> ¹ Premium file shares <code>(FileStorage)</code> Premium page blobs <code>(StorageV2)</code>	Standard general-purpose v2 <code>(StorageV2)</code> ¹ Premium block blobs <code>(BlockBlobStorage)</code> ¹ Premium file shares <code>(FileStorage)</code>	Standard general-purpose v2 <code>(StorageV2)</code> ¹	Standard general-purpose v2 <code>(StorageV2)</code> ¹
Legacy	Standard general-purpose v1 <code>(Storage)</code> Legacy blob <code>(BlobStorage)</code>	N/A	Standard general-purpose v1 <code>(Storage)</code>	N/A

¹ Accounts of this type with a hierarchical namespace enabled also support the specified redundancy option.

All data for all storage accounts is copied from the primary to the secondary according to the redundancy option for the storage account. Objects including block blobs, append blobs, page blobs, queues, tables, and files are copied.

Data in all tiers, including the archive tier, is always copied from the primary to the secondary during geo-replication. The archive tier for Blob Storage is currently supported for LRS, GRS, and RA-GRS accounts, but not for ZRS, GZRS, or RA-GZRS accounts. For more information about blob tiers, see [Access tiers for blob data](#).

Unmanaged disks don't support ZRS or GZRS.

For pricing information for each redundancy option, see [Azure Storage pricing](#).

 **Note**

Block blob storage accounts support locally redundant storage (LRS) and zone redundant storage (ZRS) in certain regions.

Data integrity

Azure Storage regularly verifies the integrity of data stored using cyclic redundancy checks (CRCs). If data corruption is detected, it's repaired using redundant data. Azure Storage also calculates checksums on all network traffic to detect corruption of data packets when storing or retrieving data.

See also

- [Change the redundancy option for a storage account](#)
- [Geo replication \(GRS/GZRS/RA-GRS/RA-GZRS\)](#)
 - [Check the Last Sync Time property for a storage account](#)
 - [Disaster recovery and storage account failover](#)
- [Pricing](#)
 - [Blob Storage](#)
 - [Azure Files](#)
 - [Table Storage](#)
 - [Queue Storage](#)
 - [Azure Disks](#)

Azure storage disaster recovery planning and failover

Article • 01/11/2024

Microsoft strives to ensure that Azure services are always available. However, unplanned service outages may occur. Key components of a good disaster recovery plan include strategies for:

- [Data protection](#)
- [Backup and restore](#)
- [Data redundancy](#)
- [Failover](#)
- [Designing applications for high availability](#)

This article focuses on failover for globally redundant storage accounts (GRS, GZRS, and RA-GZRS), and how to design your applications to be highly available if there's an outage and subsequent failover.

Choose the right redundancy option

Azure Storage maintains multiple copies of your storage account to ensure durability and high availability. Which redundancy option you choose for your account depends on the degree of resiliency you need for your applications.

With locally redundant storage (LRS), three copies of your storage account are automatically stored and replicated within a single datacenter. With zone-redundant storage (ZRS), a copy is stored and replicated in each of three separate availability zones within the same region. For more information about availability zones, see [Azure availability zones](#).

Recovery of a single copy of a storage account occurs automatically with LRS and ZRS.

Globally redundant storage and failover

With globally redundant storage (GRS, GZRS, and RA-GZRS), Azure copies your data asynchronously to a secondary geographic region at least hundreds of miles away. This allows you to recover your data if there's an outage in the primary region. A feature that distinguishes globally redundant storage from LRS and ZRS is the ability to fail over to the secondary region if there's an outage in the primary region. The process of failing over updates the DNS entries for your storage account service endpoints such that the

endpoints for the secondary region become the new primary endpoints for your storage account. Once the failover is complete, clients can begin writing to the new primary endpoints.

RA-GRS and RA-GZRS redundancy configurations provide geo-redundant storage with the added benefit of read access to the secondary endpoint if there is an outage in the primary region. If an outage occurs in the primary endpoint, applications configured for read access to the secondary region and designed for high availability can continue to read from the secondary endpoint. Microsoft recommends RA-GZRS for maximum availability and durability of your storage accounts.

For more information about redundancy in Azure Storage, see [Azure Storage redundancy](#).

Plan for storage account failover

Azure Storage accounts support two types of failover:

- **Customer-managed failover** - Customers can manage storage account failover if there's an unexpected service outage.
- **Microsoft-managed failover** - Potentially initiated by Microsoft only in the case of a severe disaster in the primary region.^{1,2}

¹Microsoft-managed failover can't be initiated for individual storage accounts, subscriptions, or tenants. For more details see [Microsoft-managed failover](#).

² Your disaster recovery plan should be based on customer-managed failover. **Do not** rely on Microsoft-managed failover, which would only be used in extreme circumstances.

Each type of failover has a unique set of use cases, corresponding expectations for data loss, and support for accounts with a hierarchical namespace enabled (Azure Data Lake Storage Gen2). This table summarizes those aspects of each type of failover :

Expand table

Type	Failover Scope	Use case	Expected data loss	HNS supported
Customer-managed	Storage account	<p>The storage service endpoints for the primary region become unavailable, but the secondary region is available.</p> <p>You received an Azure Advisory in which Microsoft advises you to perform</p>	Yes	Yes (<i>In preview</i>)

Type	Failover Scope	Use case	Expected data loss	HNS supported
		a failover operation of storage accounts potentially affected by an outage.		
Microsoft-managed	Entire region or scale unit	The primary region becomes completely unavailable due to a significant disaster, but the secondary region is available.	Yes	Yes

Customer-managed failover

If the data endpoints for the storage services in your storage account become unavailable in the primary region, you can fail over to the secondary region. After the failover is complete, the secondary region becomes the new primary and users can proceed to access data in the new primary region.

To fully understand the impact that customer-managed account failover would have on your users and applications, it is helpful to know what happens during every step of the failover and fallback process. For details about how the process works, see [How customer-managed storage account failover works](#).

Microsoft-managed failover

In extreme circumstances where the original primary region is deemed unrecoverable within a reasonable amount of time due to a major disaster, Microsoft **may** initiate a regional failover. In this case, no action on your part is required. Until the Microsoft-managed failover has completed, you won't have write access to your storage account. Your applications can read from the secondary region if your storage account is configured for RA-GRS or RA-GZRS.

ⓘ Important

Your disaster recovery plan should be based on customer-managed failover. **Do not** rely on Microsoft-managed failover, which might only be used in extreme circumstances. A Microsoft-managed failover would be initiated for an entire physical unit, such as a region or scale unit. It can't be initiated for individual storage accounts, subscriptions, or tenants. For the ability to selectively failover your individual storage accounts, use [customer-managed account failover](#).

Anticipate data loss and inconsistencies

✖ Caution

Storage account failover usually involves some data loss, and potentially file and data inconsistencies. In your disaster recovery plan, it's important to consider the impact that an account failover would have on your data before initiating one.

Because data is written asynchronously from the primary region to the secondary region, there's always a delay before a write to the primary region is copied to the secondary. If the primary region becomes unavailable, the most recent writes may not yet have been copied to the secondary.

When a failover occurs, all data in the primary region is lost as the secondary region becomes the new primary. All data already copied to the secondary is maintained when the failover happens. However, any data written to the primary that hasn't also been copied to the secondary region is lost permanently.

The new primary region is configured to be locally redundant (LRS) after the failover.

You also might experience file or data inconsistencies if your storage accounts have one or more of the following enabled:

- [Hierarchical namespace \(Azure Data Lake Storage Gen2\)](#)
- [Change feed](#)
- [Point-in-time restore for block blobs](#)

Last sync time

The **Last Sync Time** property indicates the most recent time that data from the primary region is guaranteed to have been written to the secondary region. For accounts that have a hierarchical namespace, the same **Last Sync Time** property also applies to the metadata managed by the hierarchical namespace, including ACLs. All data and metadata written prior to the last sync time is available on the secondary, while data and metadata written after the last sync time may not have been written to the secondary, and may be lost. Use this property if there's an outage to estimate the amount of data loss you may incur by initiating an account failover.

As a best practice, design your application so that you can use the last sync time to evaluate expected data loss. For example, if you're logging all write operations, then you can compare the time of your last write operations to the last sync time to determine which writes haven't been synced to the secondary.

For more information about checking the **Last Sync Time** property, see [Check the Last Sync Time property for a storage account](#).

File consistency for Azure Data Lake Storage Gen2

Replication for storage accounts with a [hierarchical namespace enabled \(Azure Data Lake Storage Gen2\)](#) occurs at the file level. This means if an outage in the primary region occurs, it is possible that only some of the files in a container or directory might have successfully replicated to the secondary region. Consistency for all files in a container or directory after a storage account failover is not guaranteed.

Change feed and blob data inconsistencies

Storage account failover of geo-redundant storage accounts with [change feed](#) enabled may result in inconsistencies between the change feed logs and the blob data and/or metadata. Such inconsistencies can result from the asynchronous nature of both updates to the change logs and the replication of blob data from the primary to the secondary region. The only situation in which inconsistencies would not be expected is when all of the current log records have been successfully flushed to the log files, and all of the storage data has been successfully replicated from the primary to the secondary region.

For information about how change feed works see [How the change feed works](#).

Keep in mind that other storage account features require the change feed to be enabled such as [operational backup of Azure Blob Storage](#), [Object replication](#) and [Point-in-time restore for block blobs](#).

Point-in-time restore inconsistencies

Customer-managed failover is supported for general-purpose v2 standard tier storage accounts that include block blobs. However, performing a customer-managed failover on a storage account resets the earliest possible restore point for the account. Data for [Point-in-time restore for block blobs](#) is only consistent up to the failover completion time. As a result, you can only restore block blobs to a point in time no earlier than the failover completion time. You can check the failover completion time in the redundancy tab of your storage account in the Azure Portal.

For example, suppose you have set the retention period to 30 days. If more than 30 days have elapsed since the failover, then you can restore to any point within that 30 days. However, if fewer than 30 days have elapsed since the failover, then you can't restore to

a point prior to the failover, regardless of the retention period. For example, if it's been 10 days since the failover, then the earliest possible restore point is 10 days in the past, not 30 days in the past.

The time and cost of failing over

The time it takes for failover to complete after being initiated can vary, although it typically takes less than one hour.

A customer-managed failover loses its geo-redundancy after a failover (and failback). Your storage account is automatically converted to locally redundant storage (LRS) in the new primary region during a failover, and the storage account in the original primary region is deleted.

You can re-enable geo-redundant storage (GRS) or read-access geo-redundant storage (RA-GRS) for the account, but note that converting from LRS to GRS or RA-GRS incurs an additional cost. The cost is due to the network egress charges to re-replicate the data to the new secondary region. Also, all archived blobs need to be rehydrated to an online tier before the account can be configured for geo-redundancy, which will incur a cost.

For more information about pricing, see:

- [Bandwidth Pricing Details](#)
- [Azure Storage pricing](#)

After you re-enable GRS for your storage account, Microsoft begins replicating the data in your account to the new secondary region. Replication time depends on many factors, which include:

- The number and size of the objects in the storage account. Replicating many small objects can take longer than replicating fewer and larger objects.
- The available resources for background replication, such as CPU, memory, disk, and WAN capacity. Live traffic takes priority over geo replication.
- If your storage account contains blobs, the number of snapshots per blob.
- If your storage account contains tables, the [data partitioning strategy](#). The replication process can't scale beyond the number of partition keys that you use.

Supported storage account types

All geo-redundant offerings support Microsoft-managed failover. In addition, some account types support customer-managed account failover, as shown in the following table:

Type of failover	GRS/RA-GRS	GZRS/RA-GZRS
Customer-managed failover	General-purpose v2 accounts General-purpose v1 accounts Legacy Blob Storage accounts	General-purpose v2 accounts
Microsoft-managed failover	All account types	General-purpose v2 accounts

Classic storage accounts

ⓘ Important

Customer-managed account failover is only supported for storage accounts deployed using the Azure Resource Manager (ARM) deployment model. The Azure Service Manager (ASM) deployment model, also known as *classic*, isn't supported. To make classic storage accounts eligible for customer-managed account failover, they must first be [migrated to the ARM model](#). Your storage account must be accessible to perform the upgrade, so the primary region can't currently be in a failed state.

If there's a disaster that affects the primary region, Microsoft will manage the failover for classic storage accounts. For more information, see [Microsoft-managed failover](#).

Azure Data Lake Storage Gen2

ⓘ Important

Customer-managed account failover for accounts that have a hierarchical namespace (Azure Data Lake Storage Gen2) is currently in PREVIEW and only supported in the following regions:

- (Asia Pacific) Central India
- (Asia Pacific) South East Asia
- (Europe) North Europe
- (Europe) Switzerland North
- (Europe) Switzerland West
- (Europe) West Europe

- (North America) Canada Central
- (North America) East US 2
- (North America) South Central US

To opt in to the preview, see [Set up preview features in Azure subscription](#) and specify `AllowHNSAccountFailover` as the feature name.

See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

if there's a significant disaster that affects the primary region, Microsoft will manage the failover for accounts with a hierarchical namespace. For more information, see [Microsoft-managed failover](#).

Unsupported features and services

The following features and services aren't supported for account failover:

- Azure File Sync doesn't support storage account failover. Storage accounts containing Azure file shares being used as cloud endpoints in Azure File Sync shouldn't be failed over. Doing so will cause sync to stop working and may also cause unexpected data loss in the case of newly tiered files.
- A storage account containing premium block blobs can't be failed over. Storage accounts that support premium block blobs don't currently support geo-redundancy.
- Customer-managed failover isn't supported for either the source or the destination account in an [object replication policy](#).
- To failover an account with SSH File Transfer Protocol (SFTP) enabled, you must first [disable SFTP for the account](#). If you want to resume using SFTP after the failover is complete, simply [re-enable it](#).
- Network File System (NFS) 3.0 (NFSv3) isn't supported for storage account failover. You can't create a storage account configured for global-redundancy with NFSv3 enabled.

Failover is not for account migration

Storage account failover shouldn't be used as part of your data migration strategy. Failover is a temporary solution to a service outage. For information about how to migrate your storage accounts, see [Azure Storage migration overview](#).

Storage accounts containing archived blobs

Storage accounts containing archived blobs support account failover. However, after a [customer-managed failover](#) is complete, all archived blobs need to be rehydrated to an online tier before the account can be configured for geo-redundancy.

Storage resource provider

Microsoft provides two REST APIs for working with Azure Storage resources. These APIs form the basis of all actions you can perform against Azure Storage. The Azure Storage REST API enables you to work with data in your storage account, including blob, queue, file, and table data. The Azure Storage resource provider REST API enables you to manage the storage account and related resources.

After a failover is complete, clients can again read and write Azure Storage data in the new primary region. However, the Azure Storage resource provider does not fail over, so resource management operations must still take place in the primary region. If the primary region is unavailable, you will not be able to perform management operations on the storage account.

Because the Azure Storage resource provider does not fail over, the [Location](#) property will return the original primary location after the failover is complete.

Azure virtual machines

Azure virtual machines (VMs) don't fail over as part of an account failover. If the primary region becomes unavailable, and you fail over to the secondary region, then you will need to recreate any VMs after the failover. Also, there's a potential data loss associated with the account failover. Microsoft recommends following the [high availability](#) and [disaster recovery](#) guidance specific to virtual machines in Azure.

Keep in mind that any data stored in a temporary disk is lost when the VM is shut down.

Azure unmanaged disks

As a best practice, Microsoft recommends converting unmanaged disks to managed disks. However, if you need to fail over an account that contains unmanaged disks attached to Azure VMs, you will need to shut down the VM before initiating the failover.

Unmanaged disks are stored as page blobs in Azure Storage. When a VM is running in Azure, any unmanaged disks attached to the VM are leased. An account failover can't proceed when there's a lease on a blob. To perform the failover, follow these steps:

1. Before you begin, note the names of any unmanaged disks, their logical unit numbers (LUN), and the VM to which they are attached. Doing so will make it easier to reattach the disks after the failover.
2. Shut down the VM.
3. Delete the VM, but retain the VHD files for the unmanaged disks. Note the time at which you deleted the VM.
4. Wait until the **Last Sync Time** has updated, and is later than the time at which you deleted the VM. This step is important, because if the secondary endpoint hasn't been fully updated with the VHD files when the failover occurs, then the VM may not function properly in the new primary region.
5. Initiate the account failover.
6. Wait until the account failover is complete and the secondary region has become the new primary region.
7. Create a VM in the new primary region and reattach the VHDs.
8. Start the new VM.

Keep in mind that any data stored in a temporary disk is lost when the VM is shut down.

Copying data as an alternative to failover

If your storage account is configured for read access to the secondary region, then you can design your application to read from the secondary endpoint. If you prefer not to fail over if there's an outage in the primary region, you can use tools such as [AzCopy](#) or [Azure PowerShell](#) to copy data from your storage account in the secondary region to another storage account in an unaffected region. You can then point your applications to that storage account for both read and write availability.

Design for high availability

It's important to design your application for high availability from the start. Refer to these Azure resources for guidance in designing your application and planning for disaster recovery:

- [Designing resilient applications for Azure](#): An overview of the key concepts for architecting highly available applications in Azure.
- [Resiliency checklist](#): A checklist for verifying that your application implements the best design practices for high availability.
- [Use geo-redundancy to design highly available applications](#): Design guidance for building applications to take advantage of geo-redundant storage.
- [Tutorial: Build a highly available application with Blob storage](#): A tutorial that shows how to build a highly available application that automatically switches

between endpoints as failures and recoveries are simulated.

Keep in mind these best practices for maintaining high availability for your Azure Storage data:

- **Disks:** Use [Azure Backup](#) to back up the VM disks used by your Azure virtual machines. Also consider using [Azure Site Recovery](#) to protect your VMs if there's a regional disaster.
- **Block blobs:** Turn on [soft delete](#) to protect against object-level deletions and overwrites, or copy block blobs to another storage account in a different region using [AzCopy](#), [Azure PowerShell](#), or the [Azure Data Movement library](#).
- **Files:** Use [Azure Backup](#) to back up your file shares. Also enable [soft delete](#) to protect against accidental file share deletions. For geo-redundancy when GRS isn't available, use [AzCopy](#) or [Azure PowerShell](#) to copy your files to another storage account in a different region.
- **Tables:** use [AzCopy](#) to export table data to another storage account in a different region.

Track outages

Customers may subscribe to the [Azure Service Health Dashboard](#) to track the health and status of Azure Storage and other Azure services.

Microsoft also recommends that you design your application to prepare for the possibility of write failures. Your application should expose write failures in a way that alerts you to the possibility of an outage in the primary region.

See also

- [Use geo-redundancy to design highly available applications](#)
- [Tutorial: Build a highly available application with Blob storage](#)
- [Azure Storage redundancy](#)
- [How customer-managed storage account failover works](#)

Azure Storage migration overview

Article • 05/09/2023

This article focuses on storage migrations to Azure and provides guidance on the following storage migration scenarios:

- Migration of unstructured data, such as files and objects
- Migration of block-based devices, such as disks and storage area networks (SANs)

Migration of unstructured data

Migration of unstructured data includes following scenarios:

- File migration from network attached storage (NAS) to one of the Azure file offerings:
 - [Azure Files](#)
 - [Azure NetApp Files](#)
 - [Independent software vendor \(ISV\) solutions](#).
- Object migration from object storage solutions to the Azure object storage platform:
 - [Azure Blob Storage](#)
 - [Azure Data Lake Storage](#).

Migration phases

A full migration consists of several different phases: discovery, assessment, and migration.

Discovery	Assessment	Migration
- Discover sources to be migrated	- Assess applicable target service - Technical vs. cost considerations	- Initial migration - Resync - Final switch over

Discovery phase

In the discovery phase, you determine all sources that need to be migrated like SMB shares, NFS exports, or object namespaces. You can do this phase manually, or use automated tools.

Assessment phase

The assessment phase is critical in understanding available options for the migration. To reduce the risk during migration, and to avoid common pitfalls follow these three steps:

Assessment phase steps	Options
Choose a target storage service	<ul style="list-style-type: none">- Azure Blob Storage and Data Lake Storage- Azure Files- Azure NetApp Files- ISV solutions
Select a migration method	<ul style="list-style-type: none">- Online- Offline- Combination of both
Choose the best migration tool for the job	<ul style="list-style-type: none">- Commercial tools (Azure and ISV)- Open source

There are several commercial (ISV) tools that can help with the assessment phase. See the [comparison matrix](#).

Choose a target storage service

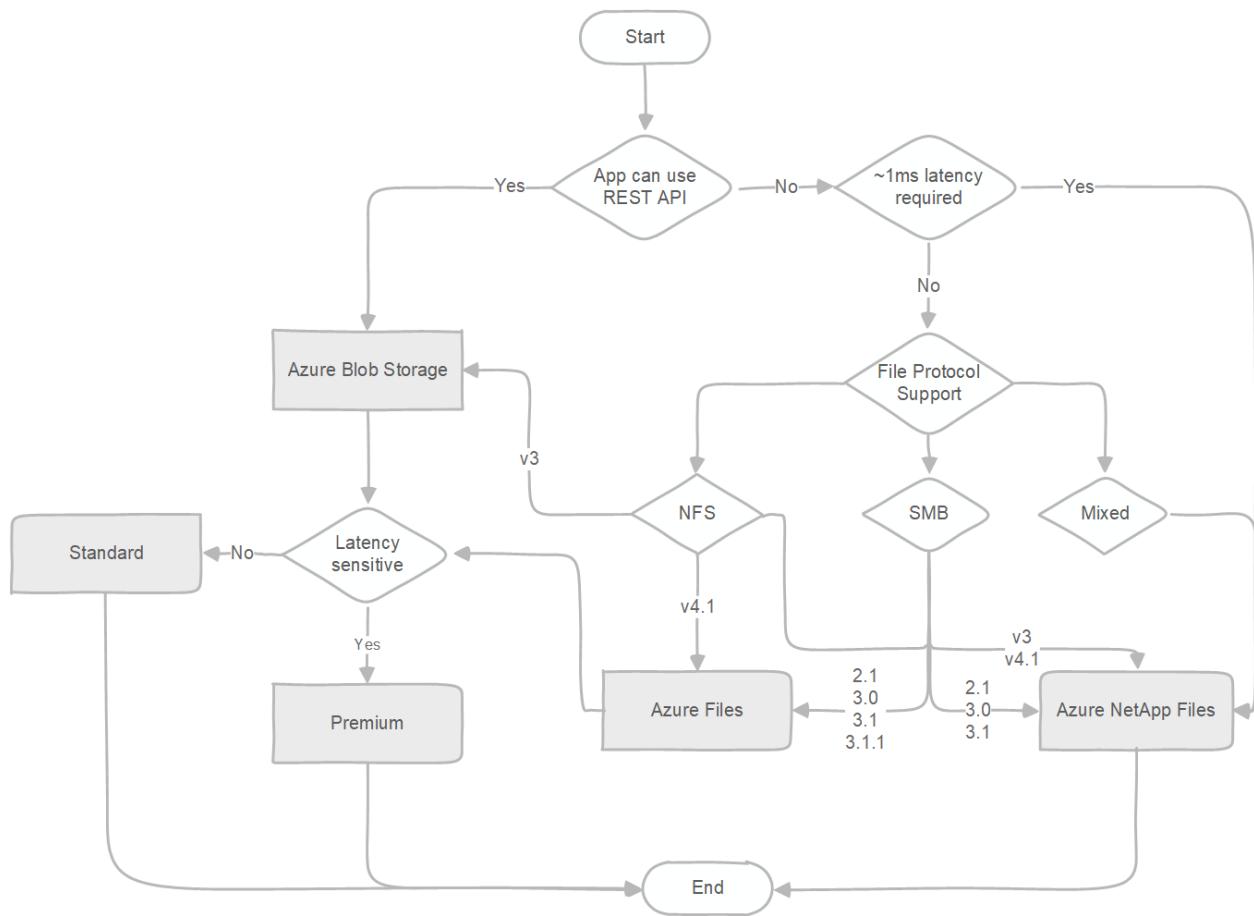
Choosing a target storage service depends on the application or users who access the data. The correct choice depends on both technical and financial aspects. First, do a technical assessment to assess possible targets and determine which services satisfy the requirements. Next, do a financial assessment to determine the best choice.

To help select the target storage service for the migration, evaluate the following aspects of each service:

- Protocol support
- Performance characteristics
- Limits of the target storage service

The following diagram is a simplified decision tree that helps guide you to the recommended Azure file service. If native Azure services do not satisfy requirements, a variety of [independent software vendor \(ISV\) solutions](#) will.

After you finish the technical assessment, and select the proper target, do a cost assessment to determine the most cost-effective option.



To keep the decision tree simple, limits of the target storage service aren't incorporated in the diagram. To find out more about current limits, and to determine whether you need to modify your choices based on them, see:

- [Storage account limits](#)
- [Blob Storage limits](#)
- [Azure Files scalability and performance targets](#)
- [Azure NetApp Files resource limits](#)

If any of the limits pose a blocker for using a service, Azure supports several storage vendors that offer their solutions on Azure Marketplace. For information about validated ISV partners that provide file services, see [Azure Storage partners for primary and secondary storage](#).

Select the migration method

There are two basic migration methods for storage migrations.

- **Online.** The online method uses the network for data migration. Either the public internet or [Azure ExpressRoute](#) can be used. If the service doesn't have a public endpoint, you must use a VPN with public internet.
- **Offline.** The offline method uses one of the [Azure Data Box](#) devices.

The decision to use an online method versus an offline method depends on the available network bandwidth. The online method is preferred in cases where there's sufficient network bandwidth to perform a migration within the needed timeline.

It's possible to use a combination of both methods, offline method for the initial bulk migration and an online method for incremental migration of changes. Using both methods simultaneously requires a high level of coordination and isn't recommended for this reason. If you choose to use both methods isolate the data sets that are migrated online from the data sets that are migrated offline.

For more information about the different migration methods and guidelines, see [Choose an Azure solution for data transfer](#) and [Migrate to Azure file shares](#).

Choose the best migration tool for the job

There are various migration tools that you can use to perform the migration. Some are open source like AzCopy, robocopy, xcopy, and rsync while others are commercial. List of available commercial tools and comparison between them is available on our [comparison matrix](#).

Open-source tools are well suited for small-scale migrations. For migration from Windows file servers to Azure Files, Microsoft recommends starting with Azure Files native capability and using [Azure File Sync](#). For more complex migrations consisting of different sources, large capacity, or special requirements like throttling or detailed reporting with audit capabilities, commercial tools are the best choice. These tools make the migration easier and reduce the risk significantly. Most commercial tools can also perform the discovery, which provides a valuable input for the assessment.

Migration phase

The migration phase is the final migration step that does data movement and migration. Typically, you'll run through the migration phase several times to accomplish an easier switchover. The migration phase consists of the following steps:

- 1. Initial migration.** The initial migration step migrates all the data from the source to the target. This step migrates the bulk of the data that needs to be migrated.
- 2. Resync.** A resync operation migrates any data that was changed after the initial migration step. You can repeat this step several times if there are numerous changes. The goal of running multiple resync operations is to reduce the time it takes for the final step. For inactive data and for data that has no changes (like backup or archive data), you can skip this step.

3. **Final switchover.** The final switchover step switches the active usage of the data from the source to the target and retires the source.

The duration of the migration for unstructured data depends on several aspects. Outside of the chosen method, the most critical factors are the total size of the data and file size distribution. The bigger the total data set, the longer the migration time. The smaller the average file size, the longer the migration time. If you have a large number of small files consider archiving them in larger files (like to a .tar or .zip file), if applicable, to reduce the total migration time.

Migration of block-based devices

Migration of block-based devices is typically done as part of virtual machine or physical host migration. It's a common misconception to delay block storage decisions until after the migration. Making these decisions ahead of time with appropriate considerations for workload requirements leads to a smoother migration to the cloud.

To explore workloads to migrate and approach to take, see the [Azure Disk Storage documentation](#), and resources on the [Disk Storage product page](#). You can learn about which disks fit your requirements, and the latest capabilities such as [disk bursting](#).

Migration of block based devices can be done in two ways:

- For migration of full virtual machines together with the underlying block-based devices, see the [Azure Migrate](#) documentation
- For migration of block based devices only, and more complexed use cases, use [Cirrus Migrate Cloud](#).

See also

- [Choose an Azure solution for data transfer](#)
- [Commercial migration tools comparison](#)
- [Migrate to Azure file shares](#)
- [Migrate to Data Lake Storage with WANdisco LiveData Platform for Azure](#)
- [Copy or move data to Azure Storage with AzCopy](#)
- [Migrate large datasets to Azure Blob Storage with AzReplicate](#)

Authorize access to blobs using Microsoft Entra ID

Article • 10/12/2023

Azure Storage supports using Microsoft Entra ID to authorize requests to blob data. With Microsoft Entra ID, you can use Azure role-based access control (Azure RBAC) to grant permissions to a security principal, which may be a user, group, or application service principal. The security principal is authenticated by Microsoft Entra ID to return an OAuth 2.0 token. The token can then be used to authorize a request against the Blob service.

Authorization with Microsoft Entra ID provides superior security and ease of use over Shared Key authorization. Microsoft recommends using Microsoft Entra authorization with your blob applications when possible to assure access with minimum required privileges.

Authorization with Microsoft Entra ID is available for all general-purpose and Blob storage accounts in all public regions and national clouds. Only storage accounts created with the Azure Resource Manager deployment model support Microsoft Entra authorization.

Blob storage additionally supports creating shared access signatures (SAS) that are signed with Microsoft Entra credentials. For more information, see [Grant limited access to data with shared access signatures](#).

Overview of Microsoft Entra ID for blobs

When a security principal (a user, group, or application) attempts to access a blob resource, the request must be authorized, unless it's a blob available for anonymous access. With Microsoft Entra ID, access to a resource is a two-step process:

1. First, the security principal's identity is authenticated and an OAuth 2.0 token is returned.

The authentication step requires that an application request an OAuth 2.0 access token at runtime. If an application is running from within an Azure entity such as an Azure VM, a virtual machine scale set, or an Azure Functions app, it can use a [managed identity](#) to access blob data.

2. Next, the token is passed as part of a request to the Blob service and used by the service to authorize access to the specified resource.

The authorization step requires that one or more Azure RBAC roles be assigned to the security principal making the request. For more information, see [Assign Azure roles for access rights](#).

Use a Microsoft Entra account with portal, PowerShell, or Azure CLI

To learn about how to access data in the Azure portal with a Microsoft Entra account, see [Data access from the Azure portal](#). To learn how to call Azure PowerShell or Azure CLI commands with a Microsoft Entra account, see [Data access from PowerShell or Azure CLI](#).

Use Microsoft Entra ID to authorize access in application code

To authorize access to Azure Storage with Microsoft Entra ID, you can use one of the following client libraries to acquire an OAuth 2.0 token:

- The Azure Identity client library is recommended for most development scenarios.
- The [Microsoft Authentication Library \(MSAL\)](#) may be suitable for certain advanced scenarios.

Azure Identity client library

The Azure Identity client library simplifies the process of getting an OAuth 2.0 access token for authorization with Microsoft Entra ID via the [Azure SDK](#). The latest versions of the Azure Storage client libraries for .NET, Java, Python, JavaScript, and Go integrate with the Azure Identity libraries for each of those languages to provide a simple and secure means to acquire an access token for authorization of Azure Storage requests.

An advantage of the Azure Identity client library is that it enables you to use the same code to acquire the access token whether your application is running in the development environment or in Azure. The Azure Identity client library returns an access token for a security principal. When your code is running in Azure, the security principal may be a managed identity for Azure resources, a service principal, or a user or group. In the development environment, the client library provides an access token for either a user or a service principal for testing purposes.

The access token returned by the Azure Identity client library is encapsulated in a token credential. You can then use the token credential to get a service client object to use in performing authorized operations against Azure Storage. A simple way to get the access

token and token credential is to use the **DefaultAzureCredential** class that is provided by the Azure Identity client library. **DefaultAzureCredential** attempts to get the token credential by sequentially trying several different credential types.

DefaultAzureCredential works in both the development environment and in Azure.

The following table points to additional information for authorizing access to data in various scenarios:

Language	.NET	Java	JavaScript	Python	Go
Overview of auth with Microsoft Entra ID	How to authenticate .NET applications with Azure services	Azure authentication with Java and Azure Identity	Authenticate apps to Azure using the Azure SDK	Authenticate Python apps to Azure using the Azure SDK	
Auth using developer service principals	Authenticate .NET apps to Azure services during local development using service principals	Azure authentication with service principal	Auth JS apps to Azure services with service principal	Authenticate Python apps to Azure services during local development using service principals	Azure SDK for Go authentication with a service principal
Auth using developer or user accounts	Authenticate .NET apps to Azure services during local development using developer accounts	Azure authentication with user credentials	Auth JS apps to Azure services with dev accounts	Authenticate Python apps to Azure services during local development using developer accounts	Azure authentication with the Azure SDK for Go
Auth from Azure-hosted apps	Authenticating Azure-hosted apps to Azure resources with the Azure SDK for .NET	Authenticate Azure-hosted Java applications	Authenticating Azure-hosted JavaScript apps to Azure resources with the Azure SDK for JavaScript	Authenticating Azure-hosted apps to Azure resources with the Azure SDK for Python	Authentication with the Azure SDK for Go using a managed identity
Auth from on-premises apps	Authenticate to Azure resources from .NET apps		Authenticate on-premises JavaScript apps to Azure resources	Authenticate to Azure resources from Python apps	

Language	.NET	Java	JavaScript	Python	Go
	hosted on-premises				
Identity client library overview	Azure Identity client library for .NET	Azure Identity client library for Java	Azure Identity client library for JavaScript	Azure Identity client library for Python	Azure Identity client library for Go ↗

Microsoft Authentication Library (MSAL)

While Microsoft recommends using the Azure Identity client library when possible, the MSAL library may be appropriate to use in certain advanced scenarios. For more information, see [Learn about MSAL](#).

When you use MSAL to acquire an OAuth token for access to Azure Storage, you need to provide a Microsoft Entra resource ID. The Microsoft Entra resource ID indicates the audience for which a token that is issued can be used to provide access to an Azure resource. In the case of Azure Storage, the resource ID may be specific to a single storage account, or it may apply to any storage account.

When you provide a resource ID that is specific to a single storage account and service, the resource ID is used to acquire a token for authorizing requests to the specified account and service only. The following table lists the value to use for the resource ID, based on the cloud you're working with. Replace `<account-name>` with the name of your storage account.

Cloud	Resource ID
Azure Global	<a href="https://<account-name>.blob.core.windows.net">https://<account-name>.blob.core.windows.net
Azure Government	<a href="https://<account-name>.blob.core.usgovcloudapi.net">https://<account-name>.blob.core.usgovcloudapi.net
Azure China 21Vianet	<a href="https://<account-name>.blob.core.chinacloudapi.cn">https://<account-name>.blob.core.chinacloudapi.cn

You can also provide a resource ID that applies to any storage account, as shown in the following table. This resource ID is the same for all public and sovereign clouds, and is used to acquire a token for authorizing requests to any storage account.

Cloud	Resource ID
Azure Global	https://storage.azure.com/
Azure Government	
Azure China 21Vianet	

Assign Azure roles for access rights

Microsoft Entra authorizes access rights to secured resources through Azure RBAC. Azure Storage defines a set of built-in RBAC roles that encompass common sets of permissions used to access blob data. You can also define custom roles for access to blob data. To learn more about assigning Azure roles for blob access, see [Assign an Azure role for access to blob data](#).

A Microsoft Entra security principal may be a user, a group, an application service principal, or a [managed identity for Azure resources](#). The RBAC roles that are assigned to a security principal determine the permissions that the principal has for the specified resource. To learn more about assigning Azure roles for blob access, see [Assign an Azure role for access to blob data](#)

In some cases you may need to enable fine-grained access to blob resources or to simplify permissions when you have a large number of role assignments for a storage resource. You can use Azure attribute-based access control (Azure ABAC) to configure conditions on role assignments. You can use conditions with a [custom role](#) or select built-in roles. For more information about configuring conditions for Azure storage resources with ABAC, see [Authorize access to blobs using Azure role assignment conditions \(preview\)](#). For details about supported conditions for blob data operations, see [Actions and attributes for Azure role assignment conditions in Azure Storage \(preview\)](#).

ⓘ Note

When you create an Azure Storage account, you are not automatically assigned permissions to access data via Microsoft Entra ID. You must explicitly assign yourself an Azure role for access to Blob Storage. You can assign it at the level of your subscription, resource group, storage account, or container.

Resource scope

Before you assign an Azure RBAC role to a security principal, determine the scope of access that the security principal should have. Best practices dictate that it's always best to grant only the narrowest possible scope. Azure RBAC roles defined at a broader scope are inherited by the resources beneath them.

You can scope access to Azure blob resources at the following levels, beginning with the narrowest scope:

- **An individual container.** At this scope, a role assignment applies to all of the blobs in the container, and to the container properties and metadata.
- **The storage account.** At this scope, a role assignment applies to all containers and their blobs.
- **The resource group.** At this scope, a role assignment applies to all of the containers in all of the storage accounts in the resource group.
- **The subscription.** At this scope, a role assignment applies to all of the containers in all of the storage accounts in all of the resource groups in the subscription.
- **A management group.** At this scope, a role assignment applies to all of the containers in all of the storage accounts in all of the resource groups in all of the subscriptions in the management group.

For more information about scope for Azure RBAC role assignments, see [Understand scope for Azure RBAC](#).

Azure built-in roles for blobs

Azure RBAC provides several built-in roles for authorizing access to blob data using Microsoft Entra ID and OAuth. Some examples of roles that provide permissions to data resources in Azure Storage include:

- [Storage Blob Data Owner](#): Use to set ownership and manage POSIX access control for Azure Data Lake Storage Gen2. For more information, see [Access control in Azure Data Lake Storage Gen2](#).
- [Storage Blob Data Contributor](#): Use to grant read/write/delete permissions to Blob storage resources.
- [Storage Blob Data Reader](#): Use to grant read-only permissions to Blob storage resources.
- [Storage Blob Delegator](#): Get a user delegation key to use to create a shared access signature that is signed with Microsoft Entra credentials for a container or blob.

To learn how to assign an Azure built-in role to a security principal, see [Assign an Azure role for access to blob data](#). To learn how to list Azure RBAC roles and their permissions, see [List Azure role definitions](#).

For more information about how built-in roles are defined for Azure Storage, see [Understand role definitions](#). For information about creating Azure custom roles, see [Azure custom roles](#).

Only roles explicitly defined for data access permit a security principal to access blob data. Built-in roles such as **Owner**, **Contributor**, and **Storage Account Contributor** permit a security principal to manage a storage account, but don't provide access to the

blob data within that account via Microsoft Entra ID. However, if a role includes **Microsoft.Storage/storageAccounts/listKeys/action**, then a user to whom that role is assigned can access data in the storage account via Shared Key authorization with the account access keys. For more information, see [Choose how to authorize access to blob data in the Azure portal](#).

For detailed information about Azure built-in roles for Azure Storage for both the data services and the management service, see the **Storage** section in [Azure built-in roles for Azure RBAC](#). Additionally, for information about the different types of roles that provide permissions in Azure, see [Azure roles, Microsoft Entra roles, and classic subscription administrator roles](#).

 **Important**

Azure role assignments may take up to 30 minutes to propagate.

Access permissions for data operations

For details on the permissions required to call specific Blob service operations, see [Permissions for calling data operations](#).

Access data with a Microsoft Entra account

Access to blob data via the Azure portal, PowerShell, or Azure CLI can be authorized either by using the user's Microsoft Entra account or by using the account access keys (Shared Key authorization).

 **Caution**

Authorization with Shared Key is not recommended as it may be less secure. For optimal security, disable authorization via Shared Key for your storage account, as described in [Prevent Shared Key authorization for an Azure Storage account](#).

Use of access keys and connection strings should be limited to initial proof of concept apps or development prototypes that don't access production or sensitive data. Otherwise, the token-based authentication classes available in the Azure SDK should always be preferred when authenticating to Azure resources.

Microsoft recommends that clients use either Microsoft Entra ID or a shared access signature (SAS) to authorize access to data in Azure Storage. For more information, see [Authorize operations for data access](#).

Data access from the Azure portal

The Azure portal can use either your Microsoft Entra account or the account access keys to access blob data in an Azure storage account. Which authorization scheme the Azure portal uses depends on the Azure roles that are assigned to you.

When you attempt to access blob data, the Azure portal first checks whether you've been assigned an Azure role with `Microsoft.Storage/storageAccounts/listkeys/action`. If you've been assigned a role with this action, then the Azure portal uses the account key for accessing blob data via Shared Key authorization. If you haven't been assigned a role with this action, then the Azure portal attempts to access data using your Microsoft Entra account.

To access blob data from the Azure portal using your Microsoft Entra account, you need permissions to access blob data, and you also need permissions to navigate through the storage account resources in the Azure portal. The built-in roles provided by Azure Storage grant access to blob resources, but they don't grant permissions to storage account resources. For this reason, access to the portal also requires the assignment of an Azure Resource Manager role such as the [Reader](#) role, scoped to the level of the storage account or higher. The [Reader](#) role grants the most restricted permissions, but another Azure Resource Manager role that grants access to storage account management resources is also acceptable. To learn more about how to assign permissions to users for data access in the Azure portal with a Microsoft Entra account, see [Assign an Azure role for access to blob data](#).

The Azure portal indicates which authorization scheme is in use when you navigate to a container. For more information about data access in the portal, see [Choose how to authorize access to blob data in the Azure portal](#).

Data access from PowerShell or Azure CLI

Azure CLI and PowerShell support signing in with Microsoft Entra credentials. After you sign in, your session runs under those credentials. To learn more, see one of the following articles:

- [Choose how to authorize access to blob data with Azure CLI](#)
- [Run PowerShell commands with Microsoft Entra credentials to access blob data](#)

Feature support

Support for this feature might be impacted by enabling Data Lake Storage Gen2, Network File System (NFS) 3.0 protocol, or the SSH File Transfer Protocol (SFTP).

If you've enabled any of these capabilities, see [Blob Storage feature support in Azure Storage accounts](#) to assess support for this feature.

Authorizing blob data operations with Microsoft Entra ID is supported only for REST API versions 2017-11-09 and later. For more information, see [Versioning for the Azure Storage services](#).

Next steps

- [Authorize access to data in Azure Storage](#)
- [Assign an Azure role for access to blob data](#)

Authorize access to Azure Blob Storage using Azure role assignment conditions

Article • 11/15/2023

Attribute-based access control (ABAC) is an authorization strategy that defines access levels based on attributes associated with security principals, resources, the environment, and the requests themselves. With ABAC, you can grant a security principal access to a resource based on a condition expressed as a predicate using these attributes.

Azure ABAC builds on Azure role-based access control (Azure RBAC) by adding [conditions to Azure role assignments](#). It enables you to author role-assignment conditions based on principal, resource, request, and environment attributes.

Important

Currently, Azure attribute-based access control (Azure ABAC) is generally available (GA) for controlling access only to Azure Blob Storage, Azure Data Lake Storage Gen2, and Azure Queues using `request`, `resource`, and `principal` attributes in the standard storage account performance tier. It is either not available or in PREVIEW for other storage account performance tiers, resource types, and attributes. For complete feature status information of ABAC for Azure Storage, see [Status of condition features in Azure Storage](#).

See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Overview of conditions in Azure Storage

You can [use Microsoft Entra ID](#) (Microsoft Entra ID) to authorize requests to Azure storage resources using Azure RBAC. Azure RBAC helps you manage access to resources by defining who has access to resources and what they can do with those resources, using role definitions and role assignments. Azure Storage defines a set of Azure [built-in roles](#) that encompass common sets of permissions used to access Azure storage data. You can also define custom roles with select sets of permissions. Azure Storage supports role assignments for both storage accounts and blob containers.

Azure ABAC builds on Azure RBAC by adding [role assignment conditions](#) in the context of specific actions. A *role assignment condition* is an additional check that is evaluated when the action on the storage resource is being authorized. This condition is expressed as a predicate using attributes associated with any of the following:

- Security principal that is requesting authorization
- Resource to which access is being requested
- Parameters of the request
- Environment in which the request is made

The benefits of using role assignment conditions are:

- **Enable finer-grained access to resources** - For example, if you want to grant a user read access to blobs in your storage accounts only if the blobs are tagged as Project=Sierra, you can use conditions on the read action using tags as an attribute.
- **Reduce the number of role assignments you have to create and manage** - You can do this by using a generalized role assignment for a security group, and then restricting the access for individual members of the group using a condition that matches attributes of a principal with attributes of a specific resource being accessed (such as a blob or a container).
- **Express access control rules in terms of attributes with business meaning** - For example, you can express your conditions using attributes that represent a project name, business application, organization function, or classification level.

The trade-off of using conditions is that you need a structured and consistent taxonomy when using attributes across your organization. Attributes must be protected to prevent access from being compromised. Also, conditions must be carefully designed and reviewed for their effect.

Role-assignment conditions in Azure Storage are supported for Azure blob storage. You can also use conditions with accounts that have the [hierarchical namespace](#) (HNS) feature enabled on them (Data Lake Storage Gen2).

Supported attributes and operations

You can configure conditions on role assignments for [DataActions](#) to achieve these goals. You can use conditions with a [custom role](#) or select built-in roles. Note, conditions aren't supported for management [Actions](#) through the [Storage resource provider](#).

You can add conditions to built-in roles or custom roles. The built-in roles on which you can use role-assignment conditions include:

- Storage Blob Data Reader
- Storage Blob Data Contributor
- Storage Blob Data Owner

You can use conditions with custom roles as long as the role includes [actions that support conditions](#).

If you're working with conditions based on [blob index tags](#), you should use the *Storage Blob Data Owner* since permissions for tag operations are included in this role.

 **Note**

Blob index tags are not supported for Data Lake Storage Gen2 storage accounts, which use a hierarchical namespace. You should not author role-assignment conditions using index tags on storage accounts that have HNS enabled.

The [Azure role assignment condition format](#) allows the use of `@Principal`, `@Resource`, `@Request` or `@Environment` attributes in the conditions. A `@Principal` attribute is a custom security attribute on a principal, such as a user, enterprise application (service principal), or managed identity. A `@Resource` attribute refers to an existing attribute of a storage resource that is being accessed, such as a storage account, a container, or a blob. A `@Request` attribute refers to an attribute or parameter included in a storage operation request. An `@Environment` attribute refers to the network environment or the date and time of a request.

[Azure RBAC supports a limited number of role assignments per subscription](#). If you need to create thousands of Azure role assignments, you might encounter this limit. Managing hundreds or thousands of role assignments can be difficult. In some cases, you can use conditions to reduce the number of role assignments on your storage account and make them easier to manage. You can [scale the management of role assignments](#) using conditions and [Microsoft Entra custom security attributes](#) for principals.

Status of condition features in Azure Storage

Currently, Azure attribute-based access control (Azure ABAC) is generally available (GA) for controlling access only to Azure Blob Storage, Azure Data Lake Storage Gen2, and Azure Queues using `request` and `resource` attributes in the standard storage account performance tier. It's either not available or in PREVIEW for other storage account performance tiers, resource types, and attributes.

See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

The following table shows the current status of ABAC by storage account performance tier, storage resource type, and attribute type. Exceptions for specific attributes are also shown.

Performance tier	Resource types	Attribute types	Attributes	Availability
Standard	Blobs Data Lake Storage Gen2 Queues	request resource principal	All attributes except for the snapshot resource attribute for Data Lake Storage Gen2	GA
Standard	Data Lake Storage Gen2	resource	snapshot	Preview
Standard	Blobs Data Lake Storage Gen2 Queues	environment	All attributes	Preview
Premium	Blobs Data Lake Storage Gen2 Queues	environment principal request resource	All attributes	Preview

Next steps

- [Prerequisites for Azure role assignment conditions](#)
- [Tutorial: Add a role assignment condition to restrict access to blobs using the Azure portal](#)
- [Actions and attributes for Azure role assignment conditions in Azure Storage](#)
- [Example Azure role assignment conditions](#)
- [Troubleshoot Azure role assignment conditions](#)

See also

- [What is Azure attribute-based access control \(Azure ABAC\)?](#)
- [FAQ for Azure role assignment conditions](#)
- [Azure role assignment condition format and syntax](#)

- Scale the management of Azure role assignments by using conditions and custom security attributes
- Security considerations for Azure role assignment conditions in Azure Storage

Data protection overview

Article • 09/20/2022

Azure Storage provides data protection for Blob Storage and Azure Data Lake Storage Gen2 to help you to prepare for scenarios where you need to recover data that has been deleted or overwritten. It's important to think about how to best protect your data before an incident occurs that could compromise it. This guide can help you decide in advance which data protection features your scenario requires, and how to implement them. If you should need to recover data that has been deleted or overwritten, this overview also provides guidance on how to proceed, based on your scenario.

In the Azure Storage documentation, *data protection* refers to strategies for protecting the storage account and data within it from being deleted or modified, or for restoring data after it has been deleted or modified. Azure Storage also offers options for *disaster recovery*, including multiple levels of redundancy to protect your data from service outages due to hardware problems or natural disasters, and customer-managed failover in the event that the data center in the primary region becomes unavailable. For more information about how your data is protected from service outages, see [Disaster recovery](#).

Recommendations for basic data protection

If you're looking for basic data protection coverage for your storage account and the data that it contains, then Microsoft recommends taking the following steps to begin with:

- Configure an Azure Resource Manager lock on the storage account to protect the account from deletion or configuration changes. [Learn more...](#)
- Enable container soft delete for the storage account to recover a deleted container and its contents. [Learn more...](#)
- Save the state of a blob at regular intervals:
 - For Blob Storage workloads, enable blob versioning to automatically save the state of your data each time a blob is overwritten. [Learn more...](#)
 - For Azure Data Lake Storage workloads, take manual snapshots to save the state of your data at a particular point in time. [Learn more...](#)

These options, as well as other data protection options for other scenarios, are described in more detail in the following section.

For an overview of the costs involved with these features, see [Summary of cost considerations](#).

Overview of data protection options

The following table summarizes the options available in Azure Storage for common data protection scenarios. Choose the scenarios that are applicable to your situation to learn more about the options available to you. Not all features are available at this time for storage accounts with a hierarchical namespace enabled.

Scenario	Data protection option	Recommendations	Protection benefit	Available for Data Lake Storage
Prevent a storage account from being deleted or modified.	Azure Resource Manager lock Learn more...	Lock all of your storage accounts with an Azure Resource Manager lock to prevent deletion of the storage account.	Protects the storage account against deletion or configuration changes. Doesn't protect containers or blobs in the account from being deleted or overwritten.	Yes
Prevent a blob version from being deleted for an interval that you control.	Immutability policy on a blob version Learn more...	Set an immutability policy on an individual blob version to protect business-critical documents, for example, in order to meet legal or regulatory compliance requirements.	Protects a blob version from being deleted and its metadata from being overwritten. An overwrite operation creates a new version. If at least one container has version-level immutability enabled, the storage account is also protected from deletion. Container deletion fails if at least one blob exists in the container.	No

Scenario	Data protection option	Recommendations	Protection benefit	Available for Data Lake Storage
Prevent a container and its blobs from being deleted or modified for an interval that you control.	Immutability policy on a container Learn more...	Set an immutability policy on a container to protect business-critical documents, for example, in order to meet legal or regulatory compliance requirements.	Protects a container and its blobs from all deletes and overwrites. When a legal hold or a locked time-based retention policy is in effect, the storage account is also protected from deletion. Containers for which no immutability policy has been set aren't protected from deletion.	Yes
Restore a deleted container within a specified interval.	Container soft delete Learn more...	<p>Enable container soft delete for all storage accounts, with a minimum retention interval of seven days.</p> <p>Enable blob versioning and blob soft delete together with container soft delete to protect individual blobs in a container.</p> <p>Store containers that require different retention periods in separate storage accounts.</p>	<p>A deleted container and its contents may be restored within the retention period.</p> <p>Only container-level operations (for example, Delete Container) can be restored. Container soft delete doesn't enable you to restore an individual blob in the container if that blob is deleted.</p>	Yes

Scenario	Data protection option	Recommendations	Protection benefit	Available for Data Lake Storage
Automatically save the state of a blob in a previous version when it's overwritten.	<p>Blob versioning Learn more...</p>	<p>Enable blob versioning, together with container soft delete and blob soft delete, for storage accounts where you need optimal protection for blob data.</p> <p>Store blob data that doesn't require versioning in a separate account to limit costs.</p>	<p>Every blob write operation creates a new version. The current version of a blob may be restored from a previous version if the current version is deleted or overwritten.</p>	No
Restore a deleted blob or blob version within a specified interval.	<p>Blob soft delete Learn more...</p>	<p>Enable blob soft delete for all storage accounts, with a minimum retention interval of seven days.</p> <p>Enable blob versioning and container soft delete together with blob soft delete for optimal protection of blob data.</p> <p>Store blobs that require different retention periods in separate storage accounts.</p>	<p>A deleted blob or blob version may be restored within the retention period.</p>	Yes

Scenario	Data protection option	Recommendations	Protection benefit	Available for Data Lake Storage
Restore a set of block blobs to a previous point in time.	Point-in-time restore Learn more...	To use point-in-time restore to revert to an earlier state, design your application to delete individual block blobs rather than deleting containers.	A set of block blobs may be reverted to their state at a specific point in the past. Only operations performed on block blobs are reverted. Any operations performed on containers, page blobs, or append blobs aren't reverted.	No
Manually save the state of a blob at a given point in time.	Blob snapshot Learn more...	Recommended as an alternative to blob versioning when versioning isn't appropriate for your scenario, due to cost or other considerations, or when the storage account has a hierarchical namespace enabled.	A blob may be restored from a snapshot if the blob is overwritten. If the blob is deleted, snapshots are also deleted.	Yes, in preview
A blob can be deleted or overwritten, but the data is regularly copied to a second storage account.	Roll-your-own solution for copying data to a second account by using Azure Storage object replication or a tool like AzCopy or Azure Data Factory.	Recommended for peace-of-mind protection against unexpected intentional actions or unpredictable scenarios. Create the second storage account in the same region as the primary account to avoid incurring egress charges.	Data can be restored from the second storage account if the primary account is compromised in any way.	AzCopy and Azure Data Factory are supported. Object replication isn't supported.

Data protection by resource type

The following table summarizes the Azure Storage data protection options according to the resources they protect.

Data protection option	Protects an account from deletion	Protects a container from deletion	Protects an object from deletion	Protects an object from overwrites
Azure Resource Manager lock	Yes	No ¹	No	No
Immutability policy on a blob version	Yes ²	Yes ³	Yes	Yes ⁴
Immutability policy on a container	Yes ⁵	Yes	Yes	Yes
Container soft delete	No	Yes	No	No
Blob versioning ⁶	No	No	Yes	Yes
Blob soft delete	No	No	Yes	Yes
Point-in-time restore ⁶	No	No	Yes	Yes
Blob snapshot	No	No	No	Yes
Roll-your-own solution for copying data to a second account ⁷	No	Yes	Yes	Yes

¹ An Azure Resource Manager lock doesn't protect a container from deletion.

² Storage account deletion fails if there is at least one container with version-level immutable storage enabled.

³ Container deletion fails if at least one blob exists in the container, regardless of whether policy is locked or unlocked.

⁴ Overwriting the contents of the current version of the blob creates a new version. An immutability policy protects a version's metadata from being overwritten.

⁵ While a legal hold or a locked time-based retention policy is in effect at container scope, the storage account is also protected from deletion.

⁶ Not currently supported for Data Lake Storage workloads.

⁷ AzCopy and Azure Data Factory are options that are supported for both Blob Storage and Data Lake Storage workloads. Object replication is supported for Blob Storage workloads only.

Recover deleted or overwritten data

If you should need to recover data that has been overwritten or deleted, how you proceed depends on which data protection options you've enabled and which resource was affected. The following table describes the actions that you can take to recover data.

Deleted or overwritten resource	Possible recovery actions	Requirements for recovery
Storage account	Attempt to recover the deleted storage account Learn more...	The storage account was originally created with the Azure Resource Manager deployment model and was deleted within the past 14 days. A new storage account with the same name hasn't been created since the original account was deleted.
Container	Recover the soft-deleted container and its contents Learn more...	Container soft delete is enabled and the container soft delete retention period hasn't yet expired.
Containers and blobs	Restore data from a second storage account	All container and blob operations have been effectively replicated to a second storage account.
Blob (any type)	Restore a blob from a previous version ¹ Learn more...	Blob versioning is enabled and the blob has one or more previous versions.
Blob (any type)	Recover a soft-deleted blob Learn more...	Blob soft delete is enabled and the soft delete retention interval hasn't expired.
Blob (any type)	Restore a blob from a snapshot Learn more...	The blob has one or more snapshots.
Set of block blobs	Recover a set of block blobs to their state at an earlier point in time ¹ Learn more...	Point-in-time restore is enabled and the restore point is within the retention interval. The storage account hasn't been compromised or corrupted.
Blob version	Recover a soft-deleted version ¹ Learn more...	Blob soft delete is enabled and the soft delete retention interval hasn't expired.

¹ Not currently supported for Data Lake Storage workloads.

Summary of cost considerations

The following table summarizes the cost considerations for the various data protection options described in this guide.

Data protection option	Cost considerations
Azure Resource Manager lock for a storage account	No charge to configure a lock on a storage account.
Immutability policy on a blob version	No charge to enable version-level immutability on a container. Creating, modifying, or deleting a time-based retention policy or legal hold on a blob version results in a write transaction charge.
Immutability policy on a container	No charge to configure an immutability policy on a container.
Container soft delete	No charge to enable container soft delete for a storage account. Data in a soft-deleted container is billed at same rate as active data until the soft-deleted container is permanently deleted.
Blob versioning	No charge to enable blob versioning for a storage account. After blob versioning is enabled, every write or delete operation on a blob in the account creates a new version, which may lead to increased capacity costs.
	A blob version is billed based on unique blocks or pages. Costs therefore increase as the base blob diverges from a particular version. Changing a blob or blob version's tier may have a billing impact. For more information, see Pricing and billing .
	Use lifecycle management to delete older versions as needed to control costs. For more information, see Optimize costs by automating Azure Blob Storage access tiers .
Blob soft delete	No charge to enable blob soft delete for a storage account. Data in a soft-deleted blob is billed at same rate as active data until the soft-deleted blob is permanently deleted.

Data protection option	Cost considerations
Point-in-time restore	<p>No charge to enable point-in-time restore for a storage account; however, enabling point-in-time restore also enables blob versioning, soft delete, and change feed, each of which may result in other charges.</p> <p>You're billed for point-in-time restore when you perform a restore operation. The cost of a restore operation depends on the amount of data being restored. For more information, see Pricing and billing.</p>
Blob snapshots	<p>Data in a snapshot is billed based on unique blocks or pages. Costs therefore increase as the base blob diverges from the snapshot. Changing a blob or snapshot's tier may have a billing impact. For more information, see Pricing and billing.</p> <p>Use lifecycle management to delete older snapshots as needed to control costs. For more information, see Optimize costs by automating Azure Blob Storage access tiers.</p>
Copy data to a second storage account	<p>Maintaining data in a second storage account will incur capacity and transaction costs. If the second storage account is located in a different region than the source account, then copying data to that second account will additionally incur egress charges.</p>

Disaster recovery

Azure Storage always maintains multiple copies of your data so that it's protected from planned and unplanned events, including transient hardware failures, network or power outages, and massive natural disasters. Redundancy ensures that your storage account meets its availability and durability targets even in the face of failures. For more information about how to configure your storage account for high availability, see [Azure Storage redundancy](#).

If a failure occurs in a data center, if your storage account is redundant across two geographical regions (geo-redundant), then you have the option to fail over your account from the primary region to the secondary region. For more information, see [Disaster recovery and storage account failover](#).

Customer-managed failover isn't currently supported for storage accounts with a hierarchical namespace enabled. For more information, see [Blob storage features available in Azure Data Lake Storage Gen2](#).

Next steps

- Azure Storage redundancy
- Disaster recovery and storage account failover

Security recommendations for Blob storage

Article • 10/12/2023

This article contains security recommendations for Blob storage. Implementing these recommendations will help you fulfill your security obligations as described in our shared responsibility model. For more information on how Microsoft fulfills service provider responsibilities, see [Shared responsibility in the cloud](#).

Some of the recommendations included in this article can be automatically monitored by Microsoft Defender for Cloud, which is the first line of defense in protecting your resources in Azure. For information on Microsoft Defender for Cloud, see [What is Microsoft Defender for Cloud?](#)

Microsoft Defender for Cloud periodically analyzes the security state of your Azure resources to identify potential security vulnerabilities. It then provides you with recommendations on how to address them. For more information on Microsoft Defender for Cloud recommendations, see [Review your security recommendations](#).

Data protection

Recommendation	Comments	Defender for Cloud
Use the Azure Resource Manager deployment model	Create new storage accounts using the Azure Resource Manager deployment model for important security enhancements, including superior Azure role-based access control (Azure RBAC) and auditing, Resource Manager-based deployment and governance, access to managed identities, access to Azure Key Vault for secrets, and Microsoft Entra authentication and authorization for access to Azure Storage data and resources. If possible, migrate existing storage accounts that use the classic deployment model to use Azure Resource Manager. For more information about Azure Resource Manager, see Azure Resource Manager overview .	-
Enable Microsoft Defender for all of your storage accounts	Microsoft Defender for Storage provides an additional layer of security intelligence that detects unusual and potentially harmful attempts to access or exploit storage accounts. Security alerts are triggered in Microsoft Defender for Cloud when anomalies in activity occur and are also sent via email to subscription administrators, with	Yes

Recommendation	Comments	Defender for Cloud
	details of suspicious activity and recommendations on how to investigate and remediate threats. For more information, see Configure Microsoft Defender for Storage .	
Turn on soft delete for blobs	Soft delete for blobs enables you to recover blob data after it has been deleted. For more information on soft delete for blobs, see Soft delete for Azure Storage blobs .	-
Turn on soft delete for containers	Soft delete for containers enables you to recover a container after it has been deleted. For more information on soft delete for containers, see Soft delete for containers .	-
Lock storage account to prevent accidental or malicious deletion or configuration changes	Apply an Azure Resource Manager lock to your storage account to protect the account from accidental or malicious deletion or configuration change. Locking a storage account does not prevent data within that account from being deleted. It only prevents the account itself from being deleted. For more information, see Apply an Azure Resource Manager lock to a storage account .	
Store business-critical data in immutable blobs	Configure legal holds and time-based retention policies to store blob data in a WORM (Write Once, Read Many) state. Blobs stored immutably can be read, but cannot be modified or deleted for the duration of the retention interval. For more information, see Store business-critical blob data with immutable storage .	-
Require secure transfer (HTTPS) to the storage account	When you require secure transfer for a storage account, all requests to the storage account must be made over HTTPS. Any requests made over HTTP are rejected. Microsoft recommends that you always require secure transfer for all of your storage accounts. For more information, see Require secure transfer to ensure secure connections .	-
Limit shared access signature (SAS) tokens to HTTPS connections only	Requiring HTTPS when a client uses a SAS token to access blob data helps to minimize the risk of eavesdropping. For more information, see Grant limited access to Azure Storage resources using shared access signatures (SAS) .	-
Disallow cross-tenant object replication	By default, an authorized user is permitted to configure an object replication policy where the source account is in one Microsoft Entra tenant and the destination account is in a different tenant. Disallow cross-tenant object replication to require that the source and destination accounts participating in an object replication policy are in	-

Recommendation	Comments	Defender for Cloud
	the same tenant. For more information, see Prevent object replication across Microsoft Entra tenants .	

Identity and access management

Recommendation	Comments	Defender for Cloud
Use Microsoft Entra ID to authorize access to blob data	Microsoft Entra ID provides superior security and ease of use over Shared Key for authorizing requests to Blob storage. For more information, see Authorize access to data in Azure Storage .	-
Keep in mind the principle of least privilege when assigning permissions to a Microsoft Entra security principal via Azure RBAC	When assigning a role to a user, group, or application, grant that security principal only those permissions that are necessary for them to perform their tasks. Limiting access to resources helps prevent both unintentional and malicious misuse of your data.	-
Use a user delegation SAS to grant limited access to blob data to clients	A user delegation SAS is secured with Microsoft Entra credentials and also by the permissions specified for the SAS. A user delegation SAS is analogous to a service SAS in terms of its scope and function, but offers security benefits over the service SAS. For more information, see Grant limited access to Azure Storage resources using shared access signatures (SAS) .	-
Secure your account access keys with Azure Key Vault	Microsoft recommends using Microsoft Entra ID to authorize requests to Azure Storage. However, if you must use Shared Key authorization, then secure your account keys with Azure Key Vault. You can retrieve the keys from the key vault at runtime, instead of saving them with your application. For more information about Azure Key Vault, see Azure Key Vault overview .	-
Regenerate your account keys periodically	Rotating the account keys periodically reduces the risk of exposing your data to malicious actors.	-
Disallow Shared Key authorization	When you disallow Shared Key authorization for a storage account, Azure Storage rejects all subsequent requests to that account that are authorized with the account access keys. Only secured requests that are authorized with Microsoft Entra ID will succeed. For	-

Recommendation	Comments	Defender for Cloud
	more information, see Prevent Shared Key authorization for an Azure Storage account .	
Keep in mind the principle of least privilege when assigning permissions to a SAS	When creating a SAS, specify only those permissions that are required by the client to perform its function. Limiting access to resources helps prevent both unintentional and malicious misuse of your data.	-
Have a revocation plan in place for any SAS that you issue to clients	If a SAS is compromised, you will want to revoke that SAS as soon as possible. To revoke a user delegation SAS, revoke the user delegation key to quickly invalidate all signatures associated with that key. To revoke a service SAS that is associated with a stored access policy, you can delete the stored access policy, rename the policy, or change its expiry time to a time that is in the past. For more information, see Grant limited access to Azure Storage resources using shared access signatures (SAS) .	-
If a service SAS is not associated with a stored access policy, then set the expiry time to one hour or less	A service SAS that is not associated with a stored access policy cannot be revoked. For this reason, limiting the expiry time so that the SAS is valid for one hour or less is recommended.	-
Disable anonymous read access to containers and blobs	anonymous read access to a container and its blobs grants read-only access to those resources to any client. Avoid enabling anonymous read access unless your scenario requires it. To learn how to disable anonymous access for a storage account, see Overview: Remediating anonymous read access for blob data .	-

Networking

Recommendation	Comments	Defender for Cloud
Configure the minimum required version of Transport Layer Security (TLS) for a storage account.	Require that clients use a more secure version of TLS to make requests against an Azure Storage account by configuring the minimum version of TLS for that account. For more information, see Configure minimum required version of Transport Layer Security (TLS) for a storage account	-

Recommendation	Comments	Defender for Cloud
Enable the Secure transfer required option on all of your storage accounts	When you enable the Secure transfer required option, all requests made against the storage account must take place over secure connections. Any requests made over HTTP will fail. For more information, see Require secure transfer in Azure Storage .	Yes
Enable firewall rules	Configure firewall rules to limit access to your storage account to requests that originate from specified IP addresses or ranges, or from a list of subnets in an Azure Virtual Network (VNet). For more information about configuring firewall rules, see Configure Azure Storage firewalls and virtual networks .	-
Allow trusted Microsoft services to access the storage account	Turning on firewall rules for your storage account blocks incoming requests for data by default, unless the requests originate from a service operating within an Azure Virtual Network (VNet) or from allowed public IP addresses. Requests that are blocked include those from other Azure services, from the Azure portal, from logging and metrics services, and so on. You can permit requests from other Azure services by adding an exception to allow trusted Microsoft services to access the storage account. For more information about adding an exception for trusted Microsoft services, see Configure Azure Storage firewalls and virtual networks .	-
Use private endpoints	A private endpoint assigns a private IP address from your Azure Virtual Network (VNet) to the storage account. It secures all traffic between your VNet and the storage account over a private link. For more information about private endpoints, see Connect privately to a storage account using Azure Private Endpoint .	-
Use VNet service tags	A service tag represents a group of IP address prefixes from a given Azure service. Microsoft manages the address prefixes encompassed by the service tag and automatically updates the service tag as addresses change. For more information about service tags supported by Azure Storage, see Azure service tags overview . For a tutorial that shows how to use service tags to create outbound network rules, see Restrict access to PaaS resources .	-
Limit network access to specific networks	Limiting network access to networks hosting clients requiring access reduces the exposure of your resources to network attacks.	Yes

Recommendation	Comments	Defender for Cloud
Configure network routing preference	<p>You can configure network routing preference for your Azure storage account to specify how network traffic is routed to your account from clients over the Internet using the Microsoft global network or Internet routing. For more information, see Configure network routing preference for Azure Storage.</p>	-

Logging/Monitoring

Recommendation	Comments	Defender for Cloud
Track how requests are authorized	<p>Enable logging for Azure Storage to track how requests to the service are authorized. The logs indicate whether a request was made anonymously, by using an OAuth 2.0 token, by using Shared Key, or by using a shared access signature (SAS). For more information, see Monitoring Azure Blob Storage with Azure Monitor or Azure Storage analytics logging with Classic Monitoring.</p>	-
Set up alerts in Azure Monitor	<p>Configure log alerts to evaluate resources logs at a set frequency and fire an alert based on the results. For more information, see Log alerts in Azure Monitor.</p>	-

Next steps

- [Azure security documentation](#)
- [Secure development documentation](#).

Performance and scalability checklist for Blob storage

Article • 06/07/2023

Microsoft has developed a number of proven practices for developing high-performance applications with Blob storage. This checklist identifies key practices that developers can follow to optimize performance. Keep these practices in mind while you're designing your application and throughout the process.

Azure Storage has scalability and performance targets for capacity, transaction rate, and bandwidth. For more information about Azure Storage scalability targets, see [Scalability and performance targets for standard storage accounts](#) and [Scalability and performance targets for Blob storage](#).

Checklist

This article organizes proven practices for performance into a checklist you can follow while developing your Blob storage application.

Done	Category	Design consideration
	Scalability targets	Can you design your application to use no more than the maximum number of storage accounts?
	Scalability targets	Are you avoiding approaching capacity and transaction limits?
	Scalability targets	Are a large number of clients accessing a single blob concurrently?
	Scalability targets	Is your application staying within the scalability targets for a single blob?
	Partitioning	Is your naming convention designed to enable better load-balancing?
	Networking	Do client-side devices have sufficiently high bandwidth and low latency to achieve the performance needed?
	Networking	Do client-side devices have a high quality network link?
	Networking	Is the client application in the same region as the storage account?
	Direct client access	Are you using shared access signatures (SAS) and cross-origin resource sharing (CORS) to enable direct access to Azure Storage?

Done	Category	Design consideration
	Caching	Is your application caching data that is frequently accessed and rarely changed?
	Caching	Is your application batching updates by caching them on the client and then uploading them in larger sets?
	.NET configuration	Have you configured your client to use a sufficient number of concurrent connections?
	.NET configuration	For .NET applications, have you configured .NET to use a sufficient number of threads?
	Parallelism	Have you ensured that parallelism is bounded appropriately so that you don't overload your client's capabilities or approach the scalability targets?
	Tools	Are you using the latest versions of Microsoft-provided client libraries and tools?
	Retries	Are you using a retry policy with an exponential backoff for throttling errors and timeouts?
	Retries	Is your application avoiding retries for non-retryable errors?
	Copying blobs	Are you copying blobs in the most efficient manner?
	Copying blobs	Are you using the latest version of AzCopy for bulk copy operations?
	Copying blobs	Are you using the Azure Data Box family for importing large volumes of data?
	Content distribution	Are you using a CDN for content distribution?
	Use metadata	Are you storing frequently used metadata about blobs in their metadata?
	Performance tuning	Are you proactively tuning client library options to optimize data transfer performance?
	Uploading quickly	When trying to upload one blob quickly, are you uploading blocks in parallel?
	Uploading quickly	When trying to upload many blobs quickly, are you uploading blobs in parallel?
	Blob type	Are you using page blobs or block blobs when appropriate?

Scalability targets

If your application approaches or exceeds any of the scalability targets, it may encounter increased transaction latencies or throttling. When Azure Storage throttles your application, the service begins to return 503 (Server busy) or 500 (Operation timeout) error codes. Avoiding these errors by staying within the limits of the scalability targets is an important part of enhancing your application's performance.

For more information about scalability targets for the Queue service, see [Azure Storage scalability and performance targets](#).

Maximum number of storage accounts

If you're approaching the maximum number of storage accounts permitted for a particular subscription/region combination, evaluate your scenario and determine whether any of the following conditions apply:

- Are you using storage accounts to store unmanaged disks and adding those disks to your virtual machines (VMs)? For this scenario, Microsoft recommends using managed disks. Managed disks scale for you automatically and without the need to create and manage individual storage accounts. For more information, see [Introduction to Azure managed disks](#)
- Are you using one storage account per customer, for the purpose of data isolation? For this scenario, Microsoft recommends using a blob container for each customer, instead of an entire storage account. Azure Storage now allows you to assign Azure roles on a per-container basis. For more information, see [Assign an Azure role for access to blob data](#).
- Are you using multiple storage accounts to shard to increase ingress, egress, I/O operations per second (IOPS), or capacity? In this scenario, Microsoft recommends that you take advantage of increased limits for storage accounts to reduce the number of storage accounts required for your workload if possible. Contact [Azure Support](#) to request increased limits for your storage account.

Capacity and transaction targets

If your application is approaching the scalability targets for a single storage account, consider adopting one of the following approaches:

- If your application hits the transaction target, consider using block blob storage accounts, which are optimized for high transaction rates and low and consistent latency. For more information, see [Azure storage account overview](#).

- Reconsider the workload that causes your application to approach or exceed the scalability target. Can you design it differently to use less bandwidth or capacity, or fewer transactions?
- If your application must exceed one of the scalability targets, then create multiple storage accounts and partition your application data across those multiple storage accounts. If you use this pattern, then be sure to design your application so that you can add more storage accounts in the future for load balancing. Storage accounts themselves have no cost other than your usage in terms of data stored, transactions made, or data transferred.
- If your application is approaching the bandwidth targets, consider compressing data on the client side to reduce the bandwidth required to send the data to Azure Storage. While compressing data may save bandwidth and improve network performance, it can also have negative effects on performance. Evaluate the performance impact of the additional processing requirements for data compression and decompression on the client side. Keep in mind that storing compressed data can make troubleshooting more difficult because it may be more challenging to view the data using standard tools.
- If your application is approaching the scalability targets, then make sure that you're using an exponential backoff for retries. It's best to try to avoid reaching the scalability targets by implementing the recommendations described in this article. However, using an exponential backoff for retries prevents your application from retrying rapidly, which could make throttling worse. For more information, see the section titled [Timeout and Server Busy errors](#).

Multiple clients accessing a single blob concurrently

If you have a large number of clients accessing a single blob concurrently, you need to consider both per blob and per storage account scalability targets. The exact number of clients that can access a single blob varies depending on factors such as the number of clients requesting the blob simultaneously, the size of the blob, and network conditions.

If the blob can be distributed through a CDN such as images or videos served from a website, then you can use a CDN. For more information, see the section titled [Content distribution](#).

In other scenarios, such as scientific simulations where the data is confidential, you have two options. The first is to stagger your workload's access such that the blob is accessed over a period of time vs being accessed simultaneously. Alternatively, you can temporarily copy the blob to multiple storage accounts to increase the total IOPS per blob and across storage accounts. Results vary depending on your application's behavior, so be sure to test concurrency patterns during design.

Bandwidth and operations per blob

A single blob supports up to 500 requests per second. If you have multiple clients that need to read the same blob and you might exceed this limit, then consider using a block blob storage account. A block blob storage account provides a higher request rate, or I/O operations per second (IOPS).

You can also use a content delivery network (CDN) such as Azure CDN to distribute operations on the blob. For more information about Azure CDN, see [Azure CDN overview](#).

Partitioning

Understanding how Azure Storage partitions your blob data is useful for enhancing performance. Azure Storage can serve data in a single partition more quickly than data that spans multiple partitions. By naming your blobs appropriately, you can improve the efficiency of read requests.

Blob storage uses a range-based partitioning scheme for scaling and load balancing. Each blob has a partition key comprised of the full blob name (account+container+blob). The partition key is used to partition blob data into ranges. The ranges are then load-balanced across Blob storage.

Range-based partitioning means that naming conventions that use lexical ordering (for example, *mypayroll*, *myperformance*, *myemployees*, etc.) or timestamps (*log20160101*, *log20160102*, *log20160102*, etc.) are more likely to result in the partitions being co-located on the same partition server until increased load requires that they're split into smaller ranges. Co-locating blobs on the same partition server enhances performance, so an important part of performance enhancement involves naming blobs in a way that organizes them most effectively.

For example, all blobs within a container can be served by a single server until the load on these blobs requires further rebalancing of the partition ranges. Similarly, a group of lightly loaded accounts with their names arranged in lexical order may be served by a single server until the load on one or all of these accounts require them to be split across multiple partition servers.

Each load-balancing operation may impact the latency of storage calls during the operation. The service's ability to handle a sudden burst of traffic to a partition is limited by the scalability of a single partition server until the load-balancing operation kicks in and rebalances the partition key range.

You can follow some best practices to reduce the frequency of such operations.

- If possible, use blob or block sizes greater than 256 KiB for standard and premium storage accounts. Larger blob or block sizes automatically activate high-throughput block blobs. High-throughput block blobs provide high-performance ingest that isn't affected by partition naming.
- Examine the naming convention you use for accounts, containers, blobs, tables, and queues. Consider prefixing account, container, or blob names with a three-digit hash using a hashing function that best suits your needs.
- If you organize your data using timestamps or numerical identifiers, make sure that you aren't using an append-only (or prepend-only) traffic pattern. These patterns aren't suitable for a range-based partitioning system. These patterns may lead to all traffic going to a single partition and limiting the system from effectively load balancing.

For example, if you have daily operations that use a blob with a timestamp such as *yyyymmdd*, then all traffic for that daily operation is directed to a single blob, which is served by a single partition server. Consider whether the per-blob limits and per-partition limits meet your needs, and consider breaking this operation into multiple blobs if needed. Similarly, if you store time series data in your tables, all traffic may be directed to the last part of the key namespace. If you're using numerical IDs, prefix the ID with a three-digit hash. If you're using timestamps, prefix the timestamp with the seconds value, for example, *ssyyyymmdd*. If your application routinely performs listing and querying operations, choose a hashing function that limits your number of queries. In some cases, a random prefix may be sufficient.

- For more information on the partitioning scheme used in Azure Storage, see [Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency](#).

Networking

The physical network constraints of the application may have a significant impact on performance. The following sections describe some of limitations users may encounter.

Client network capability

Bandwidth and the quality of the network link play important roles in application performance, as described in the following sections.

Throughput

For bandwidth, the problem is often the capabilities of the client. Larger Azure instances have NICs with greater capacity, so you should consider using a larger instance or more VMs if you need higher network limits from a single machine. If you're accessing Azure Storage from an on premises application, then the same rule applies: understand the network capabilities of the client device and the network connectivity to the Azure Storage location and either improve them as needed or design your application to work within their capabilities.

Link quality

As with any network usage, keep in mind that network conditions resulting in errors and packet loss slows effective throughput. Using WireShark or NetMon may help in diagnosing this issue.

Location

In any distributed environment, placing the client near to the server delivers in the best performance. For accessing Azure Storage with the lowest latency, the best location for your client is within the same Azure region. For example, if you have an Azure web app that uses Azure Storage, then locate them both within a single region, such as US West or Asia Southeast. Co-locating resources reduces the latency and the cost, as bandwidth usage within a single region is free.

If client applications access Azure Storage but aren't hosted within Azure, such as mobile device apps or on premises enterprise services, then locating the storage account in a region near to those clients may reduce latency. If your clients are broadly distributed (for example, some in North America, and some in Europe), then consider using one storage account per region. This approach is easier to implement if the data the application stores is specific to individual users, and doesn't require replicating data between storage accounts.

For broad distribution of blob content, use a content deliver network such as Azure CDN. For more information about Azure CDN, see [Azure CDN](#).

SAS and CORS

Suppose that you need to authorize code such as JavaScript that is running in a user's web browser or in a mobile phone app to access data in Azure Storage. One approach is to build a service application that acts as a proxy. The user's device authenticates with

the service, which in turn authorizes access to Azure Storage resources. In this way, you can avoid exposing your storage account keys on insecure devices. However, this approach places a significant overhead on the service application, because all of the data transferred between the user's device and Azure Storage must pass through the service application.

You can avoid using a service application as a proxy for Azure Storage by using shared access signatures (SAS). Using SAS, you can enable your user's device to make requests directly to Azure Storage by using a limited access token. For example, if a user wants to upload a photo to your application, then your service application can generate a SAS and send it to the user's device. The SAS token can grant permission to write to an Azure Storage resource for a specified interval of time, after which the SAS token expires. For more information about SAS, see [Grant limited access to Azure Storage resources using shared access signatures \(SAS\)](#).

Typically, a web browser doesn't allow JavaScript in a page that is hosted by a website on one domain to perform certain operations, such as write operations, to another domain. Known as the same-origin policy, this policy prevents a malicious script on one page from obtaining access to data on another web page. However, the same-origin policy can be a limitation when building a solution in the cloud. Cross-origin resource sharing (CORS) is a browser feature that enables the target domain to communicate to the browser that it trusts requests originating in the source domain.

For example, suppose a web application running in Azure makes a request for a resource to an Azure Storage account. The web application is the source domain, and the storage account is the target domain. You can configure CORS for any of the Azure Storage services to communicate to the web browser that requests from the source domain are trusted by Azure Storage. For more information about CORS, see [Cross-origin resource sharing \(CORS\) support for Azure Storage](#).

Both SAS and CORS can help you avoid unnecessary load on your web application.

Caching

Caching plays an important role in performance. The following sections discuss caching best practices.

Reading data

In general, reading data once is preferable to reading it twice. Consider the example of a web application that has retrieved a 50 MiB blob from the Azure Storage to serve as

content to a user. Ideally, the application caches the blob locally to disk and then retrieves the cached version for subsequent user requests.

One way to avoid retrieving a blob if it hasn't been modified since it was cached is to qualify the GET operation with a conditional header for modification time. If the last modified time is after the time that the blob was cached, then the blob is retrieved and re-cached. Otherwise, the cached blob is retrieved for optimal performance.

You may also decide to design your application to assume that the blob remains unchanged for a short period after retrieving it. In this case, the application doesn't need to check whether the blob was modified during that interval.

Configuration data, lookup data, and other data that is frequently used by the application are good candidates for caching.

For more information about using conditional headers, see [Specifying conditional headers for Blob service operations](#).

Uploading data in batches

In some scenarios, you can aggregate data locally, and then periodically upload it in a batch instead of uploading each piece of data immediately. For example, suppose a web application keeps a log file of activities. The application can either upload details of every activity as it happens to a table (which requires many storage operations), or it can save activity details to a local log file and then periodically upload all activity details as a delimited file to a blob. If each log entry is 1 KB in size, you can upload thousands of entries in a single transaction. A single transaction supports uploading a blob of up to 64 MiB in size. The application developer must design for the possibility of client device or upload failures. If the activity data needs to be downloaded for an interval of time rather than for a single activity, then using Blob storage is recommended over Table storage.

.NET configuration

For projects using .NET Framework, this section lists some quick configuration settings that you can use to make significant performance improvements. If you're using a language other than .NET, check to see if similar concepts apply in your chosen language.

Increase default connection limit

(!) Note

This section applies to projects using .NET Framework, as connection pooling is controlled by the `ServicePointManager` class. .NET Core introduced a significant change around connection pool management, where connection pooling happens at the `HttpClient` level and the pool size is not limited by default. This means that HTTP connections are automatically scaled to satisfy your workload. Using the latest version of .NET is recommended, when possible, to take advantage of performance enhancements.

For projects using .NET Framework, you can use the following code to increase the default connection limit (which is usually two in a client environment or ten in a server environment) to 100. Typically, you should set the value to approximately the number of threads used by your application. Set the connection limit before opening any connections.

C#

```
ServicePointManager.DefaultConnectionLimit = 100; // (Or More)
```

To learn more about connection pool limits in .NET Framework, see [.NET Framework Connection Pool Limits and the new Azure SDK for .NET](#).

For other programming languages, see the documentation to determine how to set the connection limit.

Increase minimum number of threads

If you're using synchronous calls together with asynchronous tasks, you may want to increase the number of threads in the thread pool:

C#

```
ThreadPool.SetMinThreads(100,100); // (Determine the right number for your application)
```

For more information, see the [ThreadPool.SetMinThreads](#) method.

Unbounded parallelism

While parallelism can be great for performance, be careful about using unbounded parallelism, meaning that there's no limit enforced on the number of threads or parallel requests. Be sure to limit parallel requests to upload or download data, to access multiple partitions in the same storage account, or to access multiple items in the same partition. If parallelism is unbounded, your application can exceed the client device's capabilities or the storage account's scalability targets, resulting in longer latencies and throttling.

Client libraries and tools

For best performance, always use the latest client libraries and tools provided by Microsoft. Azure Storage client libraries are available for a variety of languages. Azure Storage also supports PowerShell and Azure CLI. Microsoft actively develops these client libraries and tools with performance in mind, keeps them up-to-date with the latest service versions, and ensures that they handle many of the proven performance practices internally.

Tip

The **ABFS driver** was designed to overcome the inherent deficiencies of WASB. Microsoft recommends using the ABFS driver over the WASB driver, as the ABFS driver is optimized specifically for big data analytics.

Handle service errors

Azure Storage returns an error when the service can't process a request. Understanding the errors that may be returned by Azure Storage in a given scenario is helpful for optimizing performance. For a list of common error codes, see [Common REST API error codes](#).

Timeout and Server Busy errors

Azure Storage may throttle your application if it approaches the scalability limits. In some cases, Azure Storage may be unable to handle a request due to some transient condition. In both cases, the service may return a 503 (Server Busy) or 500 (Timeout) error. These errors can also occur if the service is rebalancing data partitions to allow for higher throughput. The client application should typically retry the operation that causes one of these errors. However, if Azure Storage is throttling your application because it's exceeding scalability targets, or even if the service was unable to serve the request for

some other reason, aggressive retries may make the problem worse. Using an exponential back off retry policy is recommended, and the client libraries default to this behavior. For example, your application may retry after 2 seconds, then 4 seconds, then 10 seconds, then 30 seconds, and then give up completely. In this way, your application significantly reduces its load on the service, rather than exacerbating behavior that could lead to throttling.

Connectivity errors can be retried immediately, because they aren't the result of throttling and are expected to be transient.

Non-retryable errors

The client libraries handle retries with an awareness of which errors can be retried and which can't be retried. However, if you're calling the Azure Storage REST API directly, there are some errors that you shouldn't retry. For example, a 400 (Bad Request) error indicates that the client application sent a request that couldn't be processed because it wasn't in the expected form. Resending this request results the same response every time, so there's no point in retrying it. If you're calling the Azure Storage REST API directly, be aware of potential errors and whether they should be retried.

For more information on Azure Storage error codes, see [Status and error codes](#).

Copying and moving blobs

Azure Storage provides a number of solutions for copying and moving blobs within a storage account, between storage accounts, and between on-premises systems and the cloud. This section describes some of these options in terms of their effects on performance. For information about efficiently transferring data to or from Blob storage, see [Choose an Azure solution for data transfer](#).

Blob copy APIs

To copy blobs across storage accounts, use the [Put Block From URL](#) operation. This operation copies data synchronously from any URL source into a block blob. Using the [Put Block from URL](#) operation can significantly reduce required bandwidth when you're migrating data across storage accounts. Because the copy operation takes place on the service side, you don't need to download and re-upload the data.

To copy data within the same storage account, use the [Copy Blob](#) operation. Copying data within the same storage account is typically completed quickly.

Use AzCopy

The AzCopy command-line utility is a simple and efficient option for bulk transfer of blobs to, from, and across storage accounts. AzCopy is optimized for this scenario, and can achieve high transfer rates. AzCopy version 10 uses the [Put Block From URL](#) operation to copy blob data across storage accounts. For more information, see [Copy or move data to Azure Storage by using AzCopy v10](#).

Use Azure Data Box

For importing large volumes of data into Blob storage, consider using the Azure Data Box family for offline transfers. Microsoft-supplied Data Box devices are a good choice for moving large amounts of data to Azure when you're limited by time, network availability, or costs. For more information, see the [Azure DataBox Documentation](#).

Content distribution

Sometimes an application needs to serve the same content to many users (for example, a product demo video used in the home page of a website), located in either the same or multiple regions. In this scenario, use a Content Delivery Network (CDN) such as Azure Front Door. Azure Front Door is Microsoft's modern cloud CDN that provides fast, reliable, and secure access between your users and your applications' static and dynamic web content across the globe. Azure Front Door delivers your Blob Storage content using Microsoft's global edge network with hundreds of [global and local points of presence \(PoPs\)](#). A CDN can typically support much higher egress limits than a single storage account and offers improved latency for content delivery to other regions.

For more information about Azure Front Door, see [Azure Front Door](#).

Use metadata

The Blob service supports HEAD requests, which can include blob properties or metadata. For example, if your application needs the Exif (exchangeable image format) data from a photo, it can retrieve the photo and extract it. To save bandwidth and improve performance, your application can store the Exif data in the blob's metadata when the application uploads the photo. You can then retrieve the Exif data in metadata using only a HEAD request. Retrieving only metadata and not the full contents of the blob saves significant bandwidth and reduces the processing time required to extract the Exif data. Keep in mind that 8 KiB of metadata can be stored per blob.

Performance tuning for data transfers

When an application transfers data using the Azure Storage client library, there are several factors that can affect speed, memory usage, and even the success or failure of the request. To maximize performance and reliability for data transfers, it's important to be proactive in configuring client library transfer options based on the environment your app runs in. To learn more, see [Performance tuning for uploads and downloads](#).

Upload blobs quickly

To upload blobs quickly, first determine whether you're uploading one blob or many. Use the below guidance to determine the correct method to use depending on your scenario. If you're using the Azure Storage client library for data transfers, see [Performance tuning for data transfers](#) for further guidance.

Upload one large blob quickly

To upload a single large blob quickly, a client application can upload its blocks or pages in parallel, being mindful of the scalability targets for individual blobs and the storage account as a whole. The Azure Storage client libraries support uploading in parallel. Client libraries for other supported languages provide similar options.

Upload many blobs quickly

To upload many blobs quickly, upload blobs in parallel. Uploading in parallel is faster than uploading single blobs at a time with parallel block uploads because it spreads the upload across multiple partitions of the storage service. AzCopy performs uploads in parallel by default, and is recommended for this scenario. For more information, see [Get started with AzCopy](#).

Choose the correct type of blob

Azure Storage supports block blobs, append blobs, and page blobs. For a given usage scenario, your choice of blob type affects the performance and scalability of your solution.

Block blobs are appropriate when you want to upload large amounts of data efficiently. For example, a client application that uploads photos or video to Blob storage would target block blobs.

Append blobs are similar to block blobs in that they're composed of blocks. When you modify an append blob, blocks are added to the end of the blob only. Append blobs are useful for scenarios such as logging, when an application needs to add data to an existing blob.

Page blobs are appropriate if the application needs to perform random writes on the data. For example, Azure virtual machine disks are stored as page blobs. For more information, see [Understanding block blobs, append blobs, and page blobs](#).

Next steps

- [Scalability and performance targets for Blob storage](#)
- [Scalability and performance targets for standard storage accounts](#)
- [Status and error codes](#)

Best practices for using Azure Data Lake Storage Gen2

Article • 03/09/2023

This article provides best practice guidelines that help you optimize performance, reduce costs, and secure your Data Lake Storage Gen2 enabled Azure Storage account.

For general suggestions around structuring a data lake, see these articles:

- [Overview of Azure Data Lake Storage for the data management and analytics scenario](#)
- [Provision three Azure Data Lake Storage Gen2 accounts for each data landing zone](#)

Find documentation

Azure Data Lake Storage Gen2 isn't a dedicated service or account type. It's a set of capabilities that support high throughput analytic workloads. The Data Lake Storage Gen2 documentation provides best practices and guidance for using these capabilities. For all other aspects of account management such as setting up network security, designing for high availability, and disaster recovery, see the [Blob storage documentation](#) content.

Evaluate feature support and known issues

Use the following pattern as you configure your account to use Blob storage features.

1. Review the [Blob Storage feature support in Azure Storage accounts](#) article to determine whether a feature is fully supported in your account. Some features aren't yet supported or have partial support in Data Lake Storage Gen2 enabled accounts. Feature support is always expanding so make sure to periodically review this article for updates.
2. Review the [Known issues with Azure Data Lake Storage Gen2](#) article to see if there are any limitations or special guidance around the feature you intend to use.
3. Scan feature articles for any guidance that is specific to Data Lake Storage Gen2 enabled accounts.

Understand the terms used in documentation

As you move between content sets, you notice some slight terminology differences. For example, content featured in the [Blob storage documentation](#), will use the term *blob* instead of *file*. Technically, the files that you ingest to your storage account become blobs in your account. Therefore, the term is correct. However, the term *blob* can cause confusion if you're used to the term *file*. You'll also see the term *container* used to refer to a *file system*. Consider these terms as synonymous.

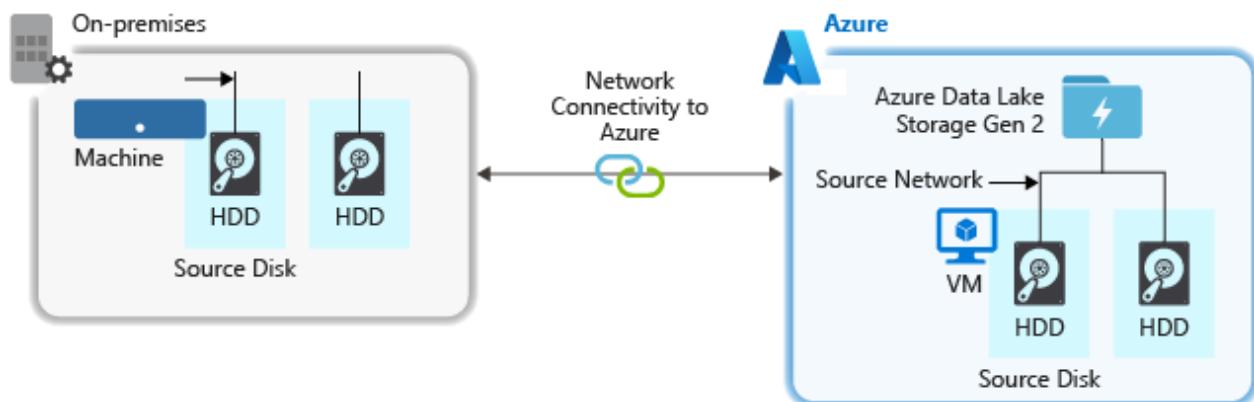
Consider premium

If your workloads require a low consistent latency and/or require a high number of input output operations per second (IOP), consider using a premium block blob storage account. This type of account makes data available via high-performance hardware. Data is stored on solid-state drives (SSDs) which are optimized for low latency. SSDs provide higher throughput compared to traditional hard drives. The storage costs of premium performance are higher, but transaction costs are lower. Therefore, if your workloads execute a large number of transactions, a premium performance block blob account can be economical.

If your storage account is going to be used for analytics, we highly recommend that you use Azure Data Lake Storage Gen2 along with a premium block blob storage account. This combination of using premium block blob storage accounts along with a Data Lake Storage enabled account is referred to as the [premium tier for Azure Data Lake Storage](#).

Optimize for data ingest

When ingesting data from a source system, the source hardware, source network hardware, or the network connectivity to your storage account can be a bottleneck.



Source hardware

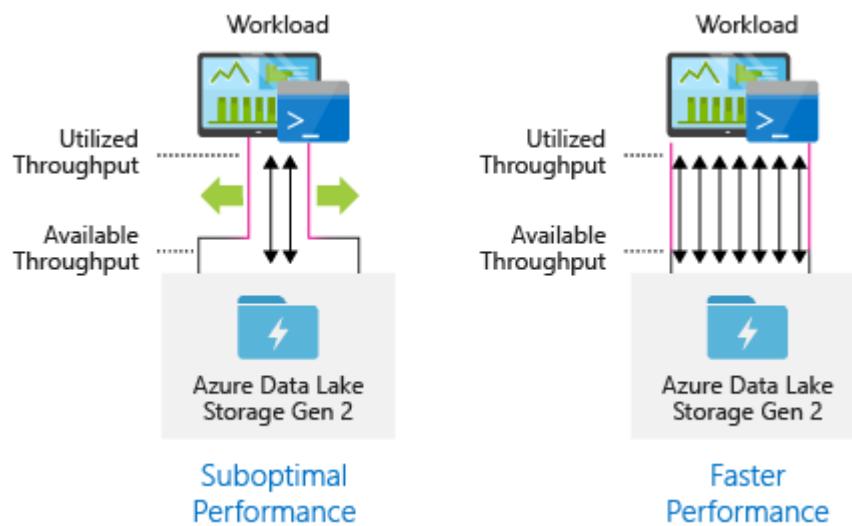
Whether you're using on-premises machines or Virtual Machines (VMs) in Azure, make sure to carefully select the appropriate hardware. For disk hardware, consider using Solid State Drives (SSD) and pick disk hardware that has faster spindles. For network hardware, use the fastest Network Interface Controllers (NIC) as possible. On Azure, we recommend Azure D14 VMs, which have the appropriately powerful disk and networking hardware.

Network connectivity to the storage account

The network connectivity between your source data and your storage account can sometimes be a bottleneck. When your source data is on premise, consider using a dedicated link with [Azure ExpressRoute](#). If your source data is in Azure, the performance is best when the data is in the same Azure region as your Data Lake Storage Gen2 enabled account.

Configure data ingestion tools for maximum parallelization

To achieve the best performance, use all available throughput by performing as many reads and writes in parallel as possible.



The following table summarizes the key settings for several popular ingestion tools.

Tool	Settings
DistCp	-m (mapper)
Azure Data Factory	parallelCopies
Sqoop	fs.azure.block.size, -m (mapper)

ⓘ Note

The overall performance of your ingest operations depend on other factors that are specific to the tool that you're using to ingest data. For the best up-to-date guidance, see the documentation for each tool that you intend to use.

Your account can scale to provide the necessary throughput for all analytics scenarios. By default, a Data Lake Storage Gen2 enabled account provides enough throughput in its default configuration to meet the needs of a broad category of use cases. If you run into the default limit, the account can be configured to provide more throughput by contacting [Azure Support](#).

Structure data sets

Consider pre-planning the structure of your data. File format, file size, and directory structure can all impact performance and cost.

File formats

Data can be ingested in various formats. Data can appear in human readable formats such as JSON, CSV, or XML or as compressed binary formats such as `.tar.gz`. Data can come in various sizes as well. Data can be composed of large files (a few terabytes) such as data from an export of a SQL table from your on-premises systems. Data can also come in the form of a large number of tiny files (a few kilobytes) such as data from real-time events from an Internet of things (IoT) solution. You can optimize efficiency and costs by choosing an appropriate file format and file size.

Hadoop supports a set of file formats that are optimized for storing and processing structured data. Some common formats are Avro, Parquet, and Optimized Row Columnar (ORC) format. All of these formats are machine-readable binary file formats. They're compressed to help you manage file size. They have a schema embedded in each file, which makes them self-describing. The difference between these formats is in how data is stored. Avro stores data in a row-based format and the Parquet and ORC formats store data in a columnar format.

Consider using the Avro file format in cases where your I/O patterns are more write heavy, or the query patterns favor retrieving multiple rows of records in their entirety. For example, the Avro format works well with a message bus such as Event Hubs or Kafka that write multiple events/messages in succession.

Consider Parquet and ORC file formats when the I/O patterns are more read heavy or when the query patterns are focused on a subset of columns in the records. Read transactions can be optimized to retrieve specific columns instead of reading the entire record.

Apache Parquet is an open source file format that is optimized for read heavy analytics pipelines. The columnar storage structure of Parquet lets you skip over non-relevant data. Your queries are much more efficient because they can narrowly scope which data to send from storage to the analytics engine. Also, because similar data types (for a column) are stored together, Parquet supports efficient data compression and encoding schemes that can lower data storage costs. Services such as [Azure Synapse Analytics](#), [Azure Databricks](#) and [Azure Data Factory](#) have native functionality that take advantage of Parquet file formats.

File size

Larger files lead to better performance and reduced costs.

Typically, analytics engines such as HDInsight have a per-file overhead that involves tasks such as listing, checking access, and performing various metadata operations. If you store your data as many small files, this can negatively affect performance. In general, organize your data into larger sized files for better performance (256 MB to 100 GB in size). Some engines and applications might have trouble efficiently processing files that are greater than 100 GB in size.

Increasing file size can also reduce transaction costs. Read and write operations are billed in 4 megabyte increments so you're charged for operation whether or not the file contains 4 megabytes or only a few kilobytes. For pricing information, see [Azure Data Lake Storage pricing](#).

Sometimes, data pipelines have limited control over the raw data, which has lots of small files. In general, we recommend that your system have some sort of process to aggregate small files into larger ones for use by downstream applications. If you're processing data in real time, you can use a real time streaming engine (such as [Azure Stream Analytics](#) or [Spark Streaming](#)) together with a message broker (such as [Event Hubs](#) or [Apache Kafka](#)) to store your data as larger files. As you aggregate small files into larger ones, consider saving them in a read-optimized format such as [Apache Parquet](#) for downstream processing.

Directory structure

Every workload has different requirements on how the data is consumed, but these are some common layouts to consider when working with Internet of Things (IoT), batch scenarios or when optimizing for time-series data.

IoT structure

In IoT workloads, there can be a great deal of data being ingested that spans across numerous products, devices, organizations, and customers. It's important to pre-plan the directory layout for organization, security, and efficient processing of the data for down-stream consumers. A general template to consider might be the following layout:

- *{Region}/{SubjectMatter(s)}/{yyyy}/{mm}/{dd}/{hh}/*

For example, landing telemetry for an airplane engine within the UK might look like the following structure:

- *UK/Planes/BA1293/Engine1/2017/08/11/12/*

In this example, by putting the date at the end of the directory structure, you can use ACLs to more easily secure regions and subject matters to specific users and groups. If you put the date structure at the beginning, it would be much more difficult to secure these regions and subject matters. For example, if you wanted to provide access only to UK data or certain planes, you'd need to apply a separate permission for numerous directories under every hour directory. This structure would also exponentially increase the number of directories as time went on.

Batch jobs structure

A commonly used approach in batch processing is to place data into an "in" directory. Then, once the data is processed, put the new data into an "out" directory for downstream processes to consume. This directory structure is sometimes used for jobs that require processing on individual files, and might not require massively parallel processing over large datasets. Like the IoT structure recommended above, a good directory structure has the parent-level directories for things such as region and subject matters (for example, organization, product, or producer). Consider date and time in the structure to allow better organization, filtered searches, security, and automation in the processing. The level of granularity for the date structure is determined by the interval on which the data is uploaded or processed, such as hourly, daily, or even monthly.

Sometimes file processing is unsuccessful due to data corruption or unexpected formats. In such cases, a directory structure might benefit from a **/bad** folder to move the files to for further inspection. The batch job might also handle the reporting or

notification of these *bad* files for manual intervention. Consider the following template structure:

- *{Region}/{SubjectMatter(s)}/In/{yyyy}/{mm}/{dd}/{hh}/*
- *{Region}/{SubjectMatter(s)}/Out/{yyyy}/{mm}/{dd}/{hh}/*
- *{Region}/{SubjectMatter(s)}/Bad/{yyyy}/{mm}/{dd}/{hh}/*

For example, a marketing firm receives daily data extracts of customer updates from their clients in North America. It might look like the following snippet before and after being processed:

- *NA/Extracts/ACMEPaperCo/In/2017/08/14/updates_08142017.csv*
- *NA/Extracts/ACMEPaperCo/Out/2017/08/14/processed_updates_08142017.csv*

In the common case of batch data being processed directly into databases such as Hive or traditional SQL databases, there isn't a need for an */in* or */out* directory because the output already goes into a separate folder for the Hive table or external database. For example, daily extracts from customers would land into their respective directories. Then, a service such as [Azure Data Factory](#), [Apache Oozie](#) ↗, or [Apache Airflow](#) ↗ would trigger a daily Hive or Spark job to process and write the data into a Hive table.

Time series data structure

For Hive workloads, partition pruning of time-series data can help some queries read only a subset of the data, which improves performance.

Those pipelines that ingest time-series data, often place their files with a structured naming for files and folders. Below is a common example we see for data that is structured by date:

/DataSet/YYYY/MM/DD/datafile_YYYY_MM_DD.tsv

Notice that the datetime information appears both as folders and in the filename.

For date and time, the following is a common pattern

/DataSet/YYYY/MM/DD/HH/mm/datafile_YYYY_MM_DD_HH_mm.tsv

Again, the choice you make with the folder and file organization should optimize for the larger file sizes and a reasonable number of files in each folder.

Set up security

Start by reviewing the recommendations in the [Security recommendations for Blob storage](#) article. You'll find best practice guidance about how to protect your data from accidental or malicious deletion, secure data behind a firewall, and use Azure Active Directory (Azure AD) as the basis of identity management.

Then, review the [Access control model in Azure Data Lake Storage Gen2](#) article for guidance that is specific to Data Lake Storage Gen2 enabled accounts. This article helps you understand how to use Azure role-based access control (Azure RBAC) roles together with access control lists (ACLs) to enforce security permissions on directories and files in your hierarchical file system.

Ingest, process, and analyze

There are many different sources of data and different ways in which that data can be ingested into a Data Lake Storage Gen2 enabled account.

For example, you can ingest large sets of data from HDInsight and Hadoop clusters or smaller sets of *ad hoc* data for prototyping applications. You can ingest streamed data that is generated by various sources such as applications, devices, and sensors. For this type of data, you can use tools to capture and process the data on an event-by-event basis in real time, and then write the events in batches into your account. You can also ingest web server logs, which contain information such as the history of page requests. For log data, consider writing custom scripts or applications to upload them so that you'll have the flexibility to include your data uploading component as part of your larger big data application.

Once the data is available in your account, you can run analysis on that data, create visualizations, and even download data to your local machine or to other repositories such as an Azure SQL database or SQL Server instance.

The following table recommends tools that you can use to ingest, analyze, visualize, and download data. Use the links in this table to find guidance about how to configure and use each tool.

Purpose	Tools & Tool guidance
Ingest ad hoc data	Azure portal, Azure PowerShell , Azure CLI , REST , Azure Storage Explorer , Apache DistCp , AzCopy
Ingest relational data	Azure Data Factory
Ingest web server logs	Azure PowerShell , Azure CLI , REST , Azure SDKs (.NET, Java, Python, and Node.js), Azure Data Factory

Purpose	Tools & Tool guidance
Ingest from HDInsight clusters	Azure Data Factory, Apache DistCp, AzCopy
Ingest from Hadoop clusters	Azure Data Factory, Apache DistCp, WANdisco LiveData Migrator for Azure, Azure Data Box
Ingest large data sets (several terabytes)	Azure ExpressRoute
Process & analyze data	Azure Synapse Analytics, Azure HDInsight, Databricks
Visualize data	Power BI, Azure Data Lake Storage query acceleration
Download data	Azure portal, PowerShell, Azure CLI, REST, Azure SDKs (.NET, Java, Python, and Node.js), Azure Storage Explorer, AzCopy, Azure Data Factory, Apache DistCp

 **Note**

This table doesn't reflect the complete list of Azure services that support Data Lake Storage Gen2. To see a list of supported Azure services, their level of support, see [Azure services that support Azure Data Lake Storage Gen2](#).

Monitor telemetry

Monitoring the use and performance is an important part of operationalizing your service. Examples include frequent operations, operations with high latency, or operations that cause service-side throttling.

All of the telemetry for your storage account is available through [Azure Storage logs in Azure Monitor](#). This feature integrates your storage account with Log Analytics and Event Hubs, while also enabling you to archive logs to another storage account. To see the full list of metrics and resources logs and their associated schema, see [Azure Storage monitoring data reference](#).

Where you choose to store your logs depends on how you plan to access them. For example, if you want to access your logs in near real time, and be able to correlate events in logs with other metrics from Azure Monitor, you can store your logs in a Log Analytics workspace. Then, query your logs by using KQL and author queries, which enumerate the `StorageBlobLogs` table in your workspace.

If you want to store your logs for both near real-time query and long term retention, you can configure your diagnostic settings to send logs to both a Log Analytics workspace and a storage account.

If you want to access your logs through another query engine such as Splunk, you can configure your diagnostic settings to send logs to an event hub and ingest logs from the event hub to your chosen destination.

Azure Storage logs in Azure Monitor can be enabled through the Azure portal, PowerShell, the Azure CLI, and Azure Resource Manager templates. For at-scale deployments, Azure Policy can be used with full support for remediation tasks. For more information, see [Azure/Community-Policy](#) and [ciphertxt/AzureStoragePolicy](#).

See also

- [Key considerations for Azure Data Lake Storage](#)
- [Access control model in Azure Data Lake Storage Gen2](#)
- [The hitchhiker's guide to the Data Lake](#)
- [Overview of Azure Data Lake Storage Gen2](#)

Azure Policy Regulatory Compliance controls for Azure Data Lake Storage Gen1

Article • 01/02/2024

[Regulatory Compliance in Azure Policy](#) provides Microsoft created and managed initiative definitions, known as *built-ins*, for the **compliance domains** and **security controls** related to different compliance standards. This page lists the **compliance domains** and **security controls** for Azure Data Lake Storage Gen1. You can assign the built-ins for a **security control** individually to help make your Azure resources compliant with the specific standard.

The title of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Policy Version** column to view the source on the [Azure Policy GitHub repo](#).

Important

Each control is associated with one or more [Azure Policy](#) definitions. These policies might help you [assess compliance](#) with the control. However, there often isn't a one-to-one or complete match between a control and one or more policies. As such, **Compliant** in Azure Policy refers only to the policies themselves. This doesn't ensure that you're fully compliant with all requirements of a control. In addition, the compliance standard includes controls that aren't addressed by any Azure Policy definitions at this time. Therefore, compliance in Azure Policy is only a partial view of your overall compliance status. The associations between controls and Azure Policy Regulatory Compliance definitions for these compliance standards can change over time.

CIS Microsoft Azure Foundations Benchmark 1.3.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CIS Microsoft Azure Foundations Benchmark 1.3.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark](#).

[+] [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
5 Logging and Monitoring	5.3	Ensure that Diagnostic Logs are enabled for all services which support it.	Resource logs in Azure Data Lake Store should be enabled	5.0.0

CIS Microsoft Azure Foundations Benchmark 1.4.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for CIS v1.4.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark](#).

[+] [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
5 Logging and Monitoring	5.3	Ensure that Diagnostic Logs Are Enabled for All Services that Support it.	Resource logs in Azure Data Lake Store should be enabled	5.0.0

CIS Microsoft Azure Foundations Benchmark 2.0.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for CIS v2.0.0](#). For more information about this compliance standard, see [CIS Microsoft Azure Foundations Benchmark](#).

[+] [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
5	5.4	Ensure that Azure Monitor Resource Logging is Enabled for All Services that Support it	Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗

CMMC Level 3

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - CMMC Level 3](#). For more information about this compliance standard, see [Cybersecurity Maturity Model Certification \(CMMC\)](#) [↗](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	SC.3.177	Employ FIPS-validated cryptography when used to protect the confidentiality of CUI.	Require encryption on Data Lake Store accounts	1.0.0 ↗
System and Communications Protection	SC.3.191	Protect the confidentiality of CUI at rest.	Require encryption on Data Lake Store accounts	1.0.0 ↗

FedRAMP High

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - FedRAMP High](#). For more information about this compliance standard, see [FedRAMP High](#) [↗](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit And Accountability	AU-6 (4)	Central Review And Analysis	Resource logs in Azure Data Lake Store should be	5.0.0 ↗

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
				enabled ↗
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit And Accountability	AU-12	Audit Generation	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗

FedRAMP Moderate

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - FedRAMP Moderate](#). For more information about this compliance standard, see [FedRAMP Moderate \[↗\]\(#\)](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit And Accountability	AU-12	Audit Generation	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗

HIPAA HITRUST 9.2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - HIPAA HITRUST 9.2](#). For more information about this compliance standard, see [HIPAA HITRUST 9.2 \[↗\]\(#\)](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
03 Portable Media Security	0304.09o3Organizational.1-09.o	0304.09o3Organizational.1-09.o 09.07 Media Handling	Require encryption on Data Lake Store accounts	1.0.0 ↗
12 Audit Logging & Monitoring	1202.09aa1System.1-09.aa	1202.09aa1System.1-09.aa 09.10 Monitoring	Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗

Microsoft cloud security benchmark

The [Microsoft cloud security benchmark](#) provides recommendations on how you can secure your cloud solutions on Azure. To see how this service completely maps to the Microsoft cloud security benchmark, see the [Azure Security Benchmark mapping files](#) ↗.

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - Microsoft cloud security benchmark](#).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Logging and Threat Detection	LT-3	Enable logging for security investigation	Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗

New Zealand ISM Restricted

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - New Zealand ISM Restricted](#). For more information about this compliance standard, see [New Zealand ISM Restricted](#) ↗.

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control and Passwords	AC-17	16.6.9 Events to be logged	Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗

NIST SP 800-171 R2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-171 R2](#). For more information about this compliance standard, see [NIST SP 800-171 R2 \[↗\]\(#\)](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗

NIST SP 800-53 Rev. 4

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 4](#). For more information about this compliance standard, see [NIST SP 800-53 Rev. 4 \[↗\]\(#\)](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit And Accountability	AU-6 (4)	Central Review And Analysis	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit And Accountability	AU-12	Audit Generation	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗

NIST SP 800-53 Rev. 5

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 5](#). For more information about this compliance standard, see [NIST SP 800-53 Rev. 5](#) ↗.

↔ [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit and Accountability	AU-6 (4)	Central Review and Analysis	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit and Accountability	AU-6 (5)	Integrated Analysis of Audit Records	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit and Accountability	AU-12	Audit Record Generation	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗
Audit and Accountability	AU-12 (1)	System-wide and Time-correlated Audit Trail	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗

NL BIO Cloud Theme

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for NL BIO Cloud Theme](#). For more information about this compliance standard, see [Baseline Information Security Government Cybersecurity - Digital Government \(digitaleoverheid.nl\)](#) ↗.

[] [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗

NZ ISM Restricted v3.5

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - NZ ISM Restricted v3.5](#). For more information about this compliance standard, see [NZ ISM Restricted v3.5](#) ↗.

[] [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control and Passwords	AC-18	16.6.9 Events to be logged	Resource logs in Azure Data Lake Store should be enabled ↗	5.0.0 ↗

Reserve Bank of India IT Framework for Banks v2016

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - RBI ITF Banks v2016](#). For more information about this compliance standard, see [RBI ITF Banks v2016 \(PDF\)](#) ↗.

[] [Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit Log Settings	Audit Log Settings-17.1		Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗

RMIT Malaysia

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance - RMIT Malaysia](#). For more information about this compliance standard, see [RMIT Malaysia](#) [↗](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Security of Digital Services	10.66	Security of Digital Services - 10.66	Deploy Diagnostic Settings for Data Lake Storage Gen1 to Event Hub	2.0.0 ↗
Security of Digital Services	10.66	Security of Digital Services - 10.66	Deploy Diagnostic Settings for Data Lake Storage Gen1 to Log Analytics workspace	1.0.0 ↗

SWIFT CSP-CSCF v2021

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see [Azure Policy Regulatory Compliance details for SWIFT CSP-CSCF v2021](#). For more information about this compliance standard, see [SWIFT CSP CSCF v2021](#) [↗](#).

[\[+\] Expand table](#)

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Resource logs in Azure Data Lake Store should be enabled	5.0.0 ↗

Next steps

- Learn more about [Azure Policy Regulatory Compliance](#).
- See the built-ins on the [Azure Policy GitHub repo](#) ↗.

Planning for an Azure Files deployment

Article • 10/05/2023

You can deploy [Azure Files](#) in two main ways: by directly mounting the serverless Azure file shares or by caching Azure file shares on-premises using Azure File Sync.

Deployment considerations will differ based on which option you choose.

- **Direct mount of an Azure file share:** Because Azure Files provides either Server Message Block (SMB) or Network File System (NFS) access, you can mount Azure file shares on-premises or in the cloud using the standard SMB or NFS clients available in your OS. Because Azure file shares are serverless, deploying for production scenarios doesn't require managing a file server or NAS device. This means you don't have to apply software patches or swap out physical disks.
- **Cache Azure file share on-premises with Azure File Sync:** [Azure File Sync](#) enables you to centralize your organization's file shares in Azure Files, while keeping the flexibility, performance, and compatibility of an on-premises file server. Azure File Sync transforms an on-premises (or cloud) Windows Server into a quick cache of your SMB Azure file share.

This article primarily addresses deployment considerations for deploying an Azure file share to be directly mounted by an on-premises or cloud client. To plan for an Azure File Sync deployment, see [Planning for an Azure File Sync deployment](#).

Available protocols

Azure Files offers two industry-standard file system protocols for mounting Azure file shares: the [Server Message Block \(SMB\)](#) protocol and the [Network File System \(NFS\)](#) protocol, allowing you to choose the protocol that is the best fit for your workload. Azure file shares don't support both the SMB and NFS protocols on the same file share, although you can create SMB and NFS Azure file shares within the same storage account. NFS 4.1 is currently only supported within new [FileStorage](#) storage account type (premium file shares only).

With both SMB and NFS file shares, Azure Files offers enterprise-grade file shares that can scale up to meet your storage needs and can be accessed concurrently by thousands of clients.

Feature	SMB	NFS
Supported protocol	SMB 3.1.1, SMB 3.0, SMB 2.1	NFS 4.1

Feature	SMB	NFS
versions		
Recommended OS	<ul style="list-style-type: none"> Windows 11, version 21H2+ Windows 10, version 21H1+ Windows Server 2019+ Linux kernel version 5.3+ 	Linux kernel version 4.3+
Available tiers	Premium, transaction optimized, hot, and cool	Premium
Billing model	<ul style="list-style-type: none"> Provisioned capacity for premium file shares Pay-as-you-go for standard file shares 	Provisioned capacity
Azure DNS Zone endpoints (preview)	Supported	Supported
Redundancy	LRS, ZRS, GRS, GZRS	LRS, ZRS
File system semantics	Win32	POSIX
Authentication	Identity-based authentication (Kerberos), shared key authentication (NTLMv2)	Host-based authentication
Authorization	Win32-style access control lists (ACLs)	UNIX-style permissions
Case sensitivity	Case insensitive, case preserving	Case sensitive
Deleting or modifying open files	With lock only	Yes
File sharing	Windows sharing mode	Byte-range advisory network lock manager
Hard link support	Not supported	Supported
Symbolic link support	Not supported	Supported
Optionally internet accessible	Yes (SMB 3.0+ only)	No
Supports FileREST	Yes	<p>Subset:</p> <ul style="list-style-type: none"> Operations on the FileService Operations on FileShares

Feature	SMB	NFS
		<ul style="list-style-type: none"> Operations on Directories Operations on Files
Mandatory byte range locks	Supported	Not supported
Advisory byte range locks	Not supported	Supported
Extended/named attributes	Not supported	Not supported
Alternate data streams	Not supported	N/A
Object identifiers	Not supported	N/A
Reparse points	Not supported	N/A
Sparse files	Not supported	N/A
Compression	Not supported	N/A
Named pipes	Not supported	N/A
SMB Direct	Not supported	N/A
SMB Directory Leasing	Not supported	N/A
Volume Shadow Copy	Not supported	N/A
Short file names (8.3 alias)	Not supported	N/A
Server service	Not supported	N/A
File system transactions (TxF)	Not supported	N/A

Management concepts

Azure file shares are deployed into *storage accounts*, which are top-level objects that represent a shared pool of storage. This pool of storage can be used to deploy multiple file shares, as well as other storage resources such as blob containers, queues, or tables. All storage resources that are deployed into a storage account share the limits that apply to that storage account. For current storage account limits, see [Azure Files scalability and performance targets](#).

There are two main types of storage accounts you will use for Azure Files deployments:

- **General purpose version 2 (GPv2) storage accounts:** GPv2 storage accounts allow you to deploy Azure file shares on standard/hard disk-based (HDD-based) hardware. In addition to storing Azure file shares, GPv2 storage accounts can store other storage resources such as blob containers, queues, or tables.
- **FileStorage storage accounts:** FileStorage storage accounts allow you to deploy Azure file shares on premium/solid-state disk-based (SSD-based) hardware. FileStorage accounts can only be used to store Azure file shares; no other storage resources (blob containers, queues, tables, etc.) can be deployed in a FileStorage account. Only FileStorage accounts can deploy both SMB and NFS file shares.

There are several other storage account types you may come across in the Azure portal, PowerShell, or CLI. Two storage account types, BlockBlobStorage and BlobStorage storage accounts, cannot contain Azure file shares. The other two storage account types you may see are general purpose version 1 (GPv1) and classic storage accounts, both of which can contain Azure file shares. Although GPv1 and classic storage accounts may contain Azure file shares, most new features of Azure Files are available only in GPv2 and FileStorage storage accounts. We therefore recommend to only use GPv2 and FileStorage storage accounts for new deployments, and to upgrade GPv1 and classic storage accounts if they already exist in your environment.

When deploying Azure file shares into storage accounts, we recommend:

- Only deploying Azure file shares into storage accounts with other Azure file shares. Although GPv2 storage accounts allow you to have mixed purpose storage accounts, because storage resources such as Azure file shares and blob containers share the storage account's limits, mixing resources together may make it more difficult to troubleshoot performance issues later on.
- Paying attention to a storage account's IOPS limitations when deploying Azure file shares. Ideally, you would map file shares 1:1 with storage accounts. However, this may not always be possible due to various limits and restrictions, both from your organization and from Azure. When it is not possible to have only one file share deployed in one storage account, consider which shares will be highly active and which shares will be less active to ensure that the hottest file shares don't get put in the same storage account together.
- Only deploying GPv2 and FileStorage accounts, and upgrading GPv1 and classic storage accounts when you find them in your environment.

Identity

To access an Azure file share, the user of the file share must be authenticated and authorized to access the share. This is done based on the identity of the user accessing the file share. Azure Files supports the following methods of authentication:

- **On-premises Active Directory Domain Services (AD DS, or on-premises AD DS):** Azure storage accounts can be domain joined to a customer-owned Active Directory Domain Services, just like a Windows Server file server or NAS device. You can deploy a domain controller on-premises, in an Azure VM, or even as a VM in another cloud provider; Azure Files is agnostic to where your domain controller is hosted. Once a storage account is domain-joined, the end user can mount a file share with the user account they signed into their PC with. AD-based authentication uses the Kerberos authentication protocol.
- **Azure Active Directory Domain Services (Azure AD DS):** Azure AD DS provides a Microsoft-managed domain controller that can be used for Azure resources. Domain joining your storage account to Azure AD DS provides similar benefits to domain joining it to a customer-owned AD DS. This deployment option is most useful for application lift-and-shift scenarios that require AD-based permissions. Since Azure AD DS provides AD-based authentication, this option also uses the Kerberos authentication protocol.
- **Azure Active Directory (Azure AD) Kerberos for hybrid identities:** Azure AD Kerberos allows you to use Azure AD to authenticate [hybrid user identities](#), which are on-premises AD identities that are synced to the cloud. This configuration uses Azure AD to issue Kerberos tickets to access the file share with the SMB protocol. This means your end users can access Azure file shares over the internet without requiring a line-of-sight to domain controllers from hybrid Azure AD-joined and Azure AD-joined VMs.
- **Active Directory authentication over SMB for Linux clients:** Azure Files supports identity-based authentication over SMB for Linux clients using the Kerberos authentication protocol through either AD DS or Azure AD DS.
- **Azure storage account key:** Azure file shares may also be mounted with an Azure storage account key. To mount a file share this way, the storage account name is used as the username and the storage account key is used as a password. Using the storage account key to mount the Azure file share is effectively an administrator operation, because the mounted file share will have full permissions to all of the files and folders on the share, even if they have ACLs. When using the storage account key to mount over SMB, the NTLMv2 authentication protocol is used. If you intend to use the storage account key to access your Azure file shares, we recommend using private endpoints or service endpoints as described in the [Networking](#) section.

For customers migrating from on-premises file servers, or creating new file shares in Azure Files intended to behave like Windows file servers or NAS appliances, domain joining your storage account to **Customer-owned AD DS** is the recommended option. To learn more about domain joining your storage account to a customer-owned AD DS, see [Overview - on-premises Active Directory Domain Services authentication over SMB for Azure file shares](#).

Networking

Directly mounting your Azure file share often requires some thought about networking configuration because:

- The port that SMB file shares use for communication, port 445, is frequently blocked by many organizations and internet service providers (ISPs) for outbound (internet) traffic.
- NFS file shares rely on network-level authentication and are therefore only accessible via restricted networks. Using an NFS file share always requires some level of networking configuration.

To configure networking, Azure Files provides an internet accessible public endpoint and integration with Azure networking features like *service endpoints*, which help restrict the public endpoint to specified virtual networks, and *private endpoints*, which give your storage account a private IP address from within a virtual network IP address space. While there's no extra charge for using public endpoints or service endpoints, standard data processing rates apply for private endpoints.

From a practical perspective, this means you'll need to consider the following network configurations:

- If the required protocol is SMB and all access over SMB is from clients in Azure, no special networking configuration is required.
- If the required protocol is SMB and the access is from clients on-premises, then a VPN or ExpressRoute connection from on-premises to your Azure network is required, with Azure Files exposed on your internal network using private endpoints.
- If the required protocol is NFS, you can use either service endpoints or private endpoints to restrict the network to specified virtual networks. If you need a static IP address and/or your workload requires high availability, use a private endpoint.

To learn more about how to configure networking for Azure Files, see [Azure Files networking considerations](#).

In addition to directly connecting to the file share using the public endpoint or using a VPN/ExpressRoute connection with a private endpoint, SMB provides an additional client access strategy: SMB over QUIC. SMB over QUIC offers zero-config "SMB VPN" for SMB access over the QUIC transport protocol. Although Azure Files does not directly support SMB over QUIC, you can create a lightweight cache of your Azure file shares on a Windows Server 2022 Azure Edition VM using Azure File Sync. To learn more about this option, see [SMB over QUIC with Azure File Sync](#).

Encryption

Azure Files supports two different types of encryption:

- **Encryption in transit**, which relates to the encryption used when mounting/accessing the Azure file share
- **Encryption at rest**, which relates to how the data is encrypted when it's stored on disk

Encryption in transit

Important

This section covers encryption in transit details for SMB shares. For details regarding encryption in transit with NFS shares, see [Security and networking](#).

By default, all Azure storage accounts have encryption in transit enabled. This means that when you mount a file share over SMB or access it via the FileREST protocol (such as through the Azure portal, PowerShell/CLI, or Azure SDKs), Azure Files will only allow the connection if it is made with SMB 3.x with encryption or HTTPS. Clients that don't support SMB 3.x or clients that support SMB 3.x but not SMB encryption won't be able to mount the Azure file share if encryption in transit is enabled. For more information about which operating systems support SMB 3.x with encryption, see our documentation for [Windows](#), [macOS](#), and [Linux](#). All current versions of the PowerShell, CLI, and SDKs support HTTPS.

You can disable encryption in transit for an Azure storage account. When encryption is disabled, Azure Files will also allow SMB 2.1 and SMB 3.x without encryption, and unencrypted FileREST API calls over HTTP. The primary reason to disable encryption in transit is to support a legacy application that must be run on an older operating system, such as Windows Server 2008 R2 or an older Linux distribution. Azure Files only allows SMB 2.1 connections within the same Azure region as the Azure file share; an SMB 2.1

client outside of the Azure region of the Azure file share, such as on-premises or in a different Azure region, won't be able to access the file share.

We strongly recommend ensuring encryption of data in-transit is enabled.

For more information about encryption in transit, see [requiring secure transfer in Azure storage](#).

Encryption at rest

All data stored in Azure Files is encrypted at rest using Azure storage service encryption (SSE). Storage service encryption works similarly to BitLocker on Windows: data is encrypted beneath the file system level. Because data is encrypted beneath the Azure file share's file system, as it's encoded to disk, you don't have to have access to the underlying key on the client to read or write to the Azure file share. Encryption at rest applies to both the SMB and NFS protocols.

By default, data stored in Azure Files is encrypted with Microsoft-managed keys. With Microsoft-managed keys, Microsoft holds the keys to encrypt/decrypt the data, and is responsible for rotating them on a regular basis. You can also choose to manage your own keys, which gives you control over the rotation process. If you choose to encrypt your file shares with customer-managed keys, Azure Files is authorized to access your keys to fulfill read and write requests from your clients. With customer-managed keys, you can revoke this authorization at any time, but this means that your Azure file share will no longer be accessible via SMB or the FileREST API.

Azure Files uses the same encryption scheme as the other Azure storage services such as Azure Blob storage. To learn more about Azure storage service encryption (SSE), see [Azure storage encryption for data at rest](#).

Data protection

Azure Files has a multi-layered approach to ensuring your data is backed up, recoverable, and protected from security threats. See [Azure Files data protection overview](#).

Soft delete

Soft delete is a storage-account level setting for SMB file shares that allows you to recover your file share when it's accidentally deleted. When a file share is deleted, it transitions to a soft deleted state instead of being permanently erased. You can

configure the amount of time soft deleted shares are recoverable before they're permanently deleted, and undelete the share anytime during this retention period.

Soft delete is enabled by default for new storage accounts from January 2021 onward, and we recommend leaving it on for most SMB file shares. If you have a workflow where share deletion is common and expected, you might decide to have a short retention period or not have soft delete enabled at all. Soft delete doesn't work for NFS shares, even if it's enabled for the storage account.

For more information about soft delete, see [Prevent accidental data deletion](#).

Backup

You can back up your Azure file share via [share snapshots](#), which are read-only, point-in-time copies of your share. Snapshots are incremental, meaning they only contain as much data as has changed since the previous snapshot. You can have up to 200 snapshots per file share and retain them for up to 10 years. You can either manually take snapshots in the Azure portal, via PowerShell, or command-line interface (CLI), or you can use [Azure Backup](#).

[Azure Backup for Azure file shares](#) handles the scheduling and retention of snapshots. Its grandfather-father-son (GFS) capabilities mean that you can take daily, weekly, monthly, and yearly snapshots, each with their own distinct retention period. Azure Backup also orchestrates the enablement of soft delete and takes a delete lock on a storage account as soon as any file share within it is configured for backup. Lastly, Azure Backup provides certain key monitoring and alerting capabilities that allow customers to have a consolidated view of their backup estate.

You can perform both item-level and share-level restores in the Azure portal using Azure Backup. All you need to do is choose the restore point (a particular snapshot), the particular file or directory if relevant, and then the location (original or alternate) you wish you restore to. The backup service handles copying the snapshot data over and shows your restore progress in the portal.

Protect Azure Files with Microsoft Defender for Storage

Microsoft Defender for Storage is an Azure-native layer of security intelligence that detects potential threats to your storage accounts. It provides comprehensive security by analyzing the data plane and control plane telemetry generated by Azure Files. It uses advanced threat detection capabilities powered by [Microsoft Threat Intelligence](#) to provide contextual security alerts, including steps to mitigate the detected threats and prevent future attacks.

Defender for Storage continuously analyzes the telemetry stream generated by Azure Files. When potentially malicious activities are detected, security alerts are generated. These alerts are displayed in Microsoft Defender for Cloud, along with the details of the suspicious activity, investigation steps, remediation actions, and security recommendations.

Defender for Storage detects known malware, such as ransomware, viruses, spyware, and other malware uploaded to a storage account based on full file hash (only supported for REST API). This helps prevent malware from entering the organization and spreading to more users and resources. See [Understanding the differences between Malware Scanning and hash reputation analysis](#).

Defender for Storage doesn't access the storage account data and doesn't impact its performance. You can [enable Microsoft Defender for Storage](#) at the subscription level (recommended) or the resource level.

Storage tiers

Azure Files offers four different tiers of storage, premium, transaction optimized, hot, and cool to allow you to tailor your shares to the performance and price requirements of your scenario:

- **Premium:** Premium file shares are backed by solid-state drives (SSDs) and provide consistent high performance and low latency, within single-digit milliseconds for most IO operations, for IO-intensive workloads. Premium file shares are suitable for a wide variety of workloads like databases, web site hosting, and development environments. Premium file shares can be used with both Server Message Block (SMB) and Network File System (NFS) protocols.
- **Transaction optimized:** Transaction optimized file shares enable transaction heavy workloads that don't need the latency offered by premium file shares. Transaction optimized file shares are offered on the standard storage hardware backed by hard disk drives (HDDs). Transaction optimized has historically been called "standard", however this refers to the storage media type rather than the tier itself (the hot and cool are also "standard" tiers, because they are on standard storage hardware).
- **Hot:** Hot file shares offer storage optimized for general purpose file sharing scenarios such as team shares. Hot file shares are offered on the standard storage hardware backed by HDDs.
- **Cool:** Cool file shares offer cost-efficient storage optimized for online archive storage scenarios. Cool file shares are offered on the standard storage hardware backed by HDDs.

Premium file shares are deployed in the **FileStorage storage account** kind and are only available in a provisioned billing model. For more information on the provisioned billing model for premium file shares, see [Understanding provisioning for premium file shares](#). Standard file shares, including transaction optimized, hot, and cool file shares, are deployed in the **general purpose version 2 (GPv2) storage account** kind, and are available through pay as you go billing.

When selecting a storage tier for your workload, consider your performance and usage requirements. If your workload requires single-digit latency, or you are using SSD storage media on-premises, the premium tier is probably the best fit. If low latency isn't as much of a concern, for example with team shares mounted on-premises from Azure or cached on-premises using Azure File Sync, standard storage may be a better fit from a cost perspective.

Once you've created a file share in a storage account, you cannot move it to tiers exclusive to different storage account kinds. For example, to move a transaction optimized file share to the premium tier, you must create a new file share in a FileStorage storage account and copy the data from your original share to a new file share in the FileStorage account. We recommend using AzCopy to copy data between Azure file shares, but you may also use tools like `robocopy` on Windows or `rsync` for macOS and Linux.

See [Understanding Azure Files billing](#) for more information.

Limitations

Currently, standard file shares with **large file shares** enabled (up to 100 TiB capacity) have certain limitations.

- Only locally redundant storage (LRS) and zone redundant storage (ZRS) accounts are supported.
- Once you enable **large file shares** on a storage account, you can't convert the storage account to use geo-redundant storage (GRS) or geo-zone-redundant storage (GZRS).
- Once you enable **large file shares**, you can't disable it.

If you want to use GRS or GZRS with standard SMB Azure file shares, see [Azure Files geo-redundancy for large file shares preview](#).

Redundancy

To protect the data in your Azure file shares against data loss or corruption, Azure Files stores multiple copies of each file as they are written. Depending on your requirements, you can select different degrees of redundancy. Azure Files currently supports the following data redundancy options:

- **Locally-redundant storage (LRS):** With LRS, every file is stored three times within an Azure storage cluster. This protects against data loss due to hardware faults, such as a bad disk drive. However, if a disaster such as fire or flooding occurs within the data center, all replicas of a storage account using LRS might be lost or unrecoverable.
- **Zone-redundant storage (ZRS):** With ZRS, three copies of each file are stored. However, these copies are physically isolated in three distinct storage clusters in different Azure *availability zones*. Availability zones are unique physical locations within an Azure region. Each zone is made up of one or more data centers equipped with independent power, cooling, and networking. A write to storage isn't accepted until it's written to the storage clusters in all three availability zones.
- **Geo-redundant storage (GRS):** With GRS, you have two regions, a primary region and a secondary region. Files are stored three times within an Azure storage cluster in the primary region. Writes are asynchronously replicated to a Microsoft-defined secondary region. GRS provides six copies of your data spread between two Azure regions. In the event of a major disaster such as the permanent loss of an Azure region due to a natural disaster or other similar event, Microsoft will perform a failover. In this case, the secondary becomes the primary, serving all operations. Because the replication between the primary and secondary regions is asynchronous, in the event of a major disaster, data not yet replicated to the secondary region will be lost. You can also perform a manual failover of a geo-redundant storage account.
- **Geo-zone-redundant storage (GZRS):** You can think of GZRS as ZRS, but with geo-redundancy. With GZRS, files are stored three times across three distinct storage clusters in the primary region. All writes are then asynchronously replicated to a Microsoft-defined secondary region. The failover process for GZRS works the same as GRS.

Standard Azure file shares up to 5 TiB support all four redundancy types. Standard file shares larger than 5 TiB only support LRS and ZRS. Premium Azure file shares only support LRS and ZRS.

General purpose version 2 (GPv2) storage accounts provide two other redundancy options that Azure Files doesn't support: read accessible geo-redundant storage (RA-GRS) and read accessible geo-zone-redundant storage (RA-GZRS). You can provision Azure file shares in storage accounts with these options set, however Azure Files doesn't

support reading from the secondary region. Azure file shares deployed into RA-GRS or RA-GZRS storage accounts are billed as GRS or GZRS, respectively.

For more information about redundancy, see [Azure Files data redundancy](#).

Standard ZRS availability

ZRS for standard general-purpose v2 storage accounts is available for a [subset of Azure regions](#).

Premium ZRS availability

ZRS for premium file shares is available for a [subset of Azure regions](#).

Standard GZRS availability

GZRS is available for a [subset of Azure regions](#).

Disaster recovery and failover

In the case of an unplanned regional service outage, you should have a disaster recovery (DR) plan in place for your Azure file shares. To understand the concepts and processes involved with DR and storage account failover, see [Disaster recovery and failover for Azure Files](#).

Migration

In many cases, you won't be establishing a net new file share for your organization, but instead migrating an existing file share from an on-premises file server or NAS device to Azure Files. Picking the right migration strategy and tool for your scenario is important for the success of your migration.

The [migration overview article](#) briefly covers the basics and contains a table that leads you to migration guides that likely cover your scenario.

Next steps

- [Planning for an Azure File Sync Deployment](#)
- [Deploying Azure Files](#)
- [Deploying Azure File Sync](#)

- Check out the migration overview article to find the migration guide for your scenario

Overview of Azure Files identity-based authentication options for SMB access

Article • 11/22/2023

This article explains how Azure file shares can use domain services, either on-premises or in Azure, to support identity-based access to Azure file shares over SMB. Enabling identity-based access for your Azure file shares allows you to replace existing file servers with Azure file shares without replacing your existing directory service, maintaining seamless user access to shares.

Applies to

File share type	SMB	NFS
Standard file shares (GPv2), LRS/ZRS	✓	✗
Standard file shares (GPv2), GRS/GZRS	✓	✗
Premium file shares (FileStorage), LRS/ZRS	✓	✗

Glossary

It's helpful to understand some key terms relating to identity-based authentication for Azure file shares:

- **Kerberos authentication**

Kerberos is an authentication protocol that's used to verify the identity of a user or host. For more information on Kerberos, see [Kerberos Authentication Overview](#).

- **Server Message Block (SMB) protocol**

SMB is an industry-standard network file-sharing protocol. For more information on SMB, see [Microsoft SMB Protocol and CIFS Protocol Overview](#).

- **Microsoft Entra ID**

Microsoft Entra ID (formerly Azure AD) is Microsoft's multi-tenant cloud-based directory and identity management service. Microsoft Entra ID combines core directory services, application access management, and identity protection into a single solution.

- **Microsoft Entra Domain Services**

Microsoft Entra Domain Services provides managed domain services such as domain join, group policies, LDAP, and Kerberos/NTLM authentication. These services are fully compatible with Active Directory Domain Services. For more information, see [Microsoft Entra Domain Services](#).

- **On-premises Active Directory Domain Services (AD DS)**

On-premises Active Directory Domain Services (AD DS) integration with Azure Files provides the methods for storing directory data while making it available to network users and administrators. Security is integrated with AD DS through logon authentication and access control to objects in the directory. With a single network logon, administrators can manage directory data and organization throughout their network, and authorized network users can access resources anywhere on the network. AD DS is commonly adopted by enterprises in on-premises environments or on cloud-hosted VMs, and AD DS credentials are used for access control. For more information, see [Active Directory Domain Services Overview](#).

- **Azure role-based access control (Azure RBAC)**

Azure RBAC enables fine-grained access management for Azure. Using Azure RBAC, you can manage access to resources by granting users the fewest permissions needed to perform their jobs. For more information, see [What is Azure role-based access control?](#)

- **Hybrid identities**

[Hybrid user identities](#) are identities in AD DS that are synced to Microsoft Entra ID using either the on-premises [Microsoft Entra Connect Sync](#) application or [Microsoft Entra Connect cloud sync](#), a lightweight agent that can be installed from the Microsoft Entra Admin Center.

Supported authentication scenarios

Azure Files supports identity-based authentication over SMB through the following methods. You can only use one method per storage account.

- **On-premises AD DS authentication:** On-premises AD DS-joined or Microsoft Entra Domain Services-joined Windows machines can access Azure file shares with on-premises Active Directory credentials that are synched to Microsoft Entra ID over SMB. Your client must have unimpeded network connectivity to your AD DS. If you already have AD DS set up on-premises or on a VM in Azure where your devices

are domain-joined to your AD, you should use AD DS for Azure file shares authentication.

- **Microsoft Entra Domain Services authentication:** Cloud-based, Microsoft Entra Domain Services-joined Windows VMs can access Azure file shares with Microsoft Entra credentials. In this solution, Microsoft Entra ID runs a traditional Windows Server AD domain on behalf of the customer, which is a child of the customer's Microsoft Entra tenant.
- **Microsoft Entra Kerberos for hybrid identities:** Using Microsoft Entra ID for authenticating [hybrid user identities](#) allows Microsoft Entra users to access Azure file shares using Kerberos authentication. This means your end users can access Azure file shares over the internet without requiring network connectivity to domain controllers from Microsoft Entra hybrid joined and Microsoft Entra joined VMs. Cloud-only identities aren't currently supported.
- **AD Kerberos authentication for Linux clients:** Linux clients can use Kerberos authentication over SMB for Azure Files using on-premises AD DS or Microsoft Entra Domain Services.

Restrictions

- None of the authentication methods support assigning share-level permissions to computer accounts (machine accounts) using Azure RBAC, because computer accounts can't be synced to an identity in Microsoft Entra ID. If you want to allow a computer account to access Azure file shares using identity-based authentication, [use a default share-level permission](#) or consider using a service logon account instead.
- Identity-based authentication isn't supported with Network File System (NFS) shares.

Common use cases

Identity-based authentication with Azure Files can be useful in a variety of scenarios:

Replace on-premises file servers

Deprecating and replacing scattered on-premises file servers is a common problem that every enterprise encounters in their IT modernization journey. Azure file shares with on-premises AD DS authentication is the best fit here, when you can migrate the data to Azure Files. A complete migration will allow you to take advantage of the high availability and scalability benefits while also minimizing the client-side changes. It

provides a seamless migration experience to end users, so they can continue to access their data with the same credentials using their existing domain-joined machines.

Lift and shift applications to Azure

When you lift and shift applications to the cloud, you want to keep the same authentication model for your data. As we extend the identity-based access control experience to Azure file shares, it eliminates the need to change your application to modern auth methods and expedite cloud adoption. Azure file shares provide the option to integrate with either Microsoft Entra Domain Services or on-premises AD DS for authentication. If your plan is to be 100% cloud native and minimize the efforts managing cloud infrastructures, Microsoft Entra Domain Services might be a better fit as a fully managed domain service. If you need full compatibility with AD DS capabilities, you might want to consider extending your AD DS environment to cloud by self-hosting domain controllers on VMs. Either way, we provide the flexibility to choose the domain service that best suits your business needs.

Backup and disaster recovery (DR)

If you're keeping your primary file storage on-premises, Azure file shares can serve as an ideal storage for backup or DR, to improve business continuity. You can use Azure file shares to back up your data from existing file servers while preserving Windows discretionary access control lists (DACLs). For DR scenarios, you can configure an authentication option to support proper access control enforcement at failover.

Advantages of identity-based authentication

Identity-based authentication for Azure Files offers several benefits over using Shared Key authentication:

- **Extend the traditional identity-based file share access experience to the cloud**
If you plan to lift and shift your application to the cloud, replacing traditional file servers with Azure file shares, then you might want your application to authenticate with either on-premises AD DS or Microsoft Entra Domain Services credentials to access file data. Azure Files supports using either on-premises AD DS or Microsoft Entra Domain Services credentials to access Azure file shares over SMB from either on-premises AD DS or Microsoft Entra Domain Services domain-joined VMs.
- **Enforce granular access control on Azure file shares**
You can grant permissions to a specific identity at the share, directory, or file level.

For example, suppose that you have several teams using a single Azure file share for project collaboration. You can grant all teams access to non-sensitive directories, while limiting access to directories containing sensitive financial data to your finance team only.

- **Back up Windows ACLs (also known as NTFS permissions) along with your data**
You can use Azure file shares to back up your existing on-premises file shares. Azure Files preserves your ACLs along with your data when you back up a file share to Azure file shares over SMB.

How it works

Azure file shares use the Kerberos protocol to authenticate with an AD source. When an identity associated with a user or application running on a client attempts to access data in Azure file shares, the request is sent to the AD source to authenticate the identity. If authentication is successful, it returns a Kerberos token. The client sends a request that includes the Kerberos token, and Azure file shares use that token to authorize the request. Azure file shares only receive the Kerberos token, not the user's access credentials.

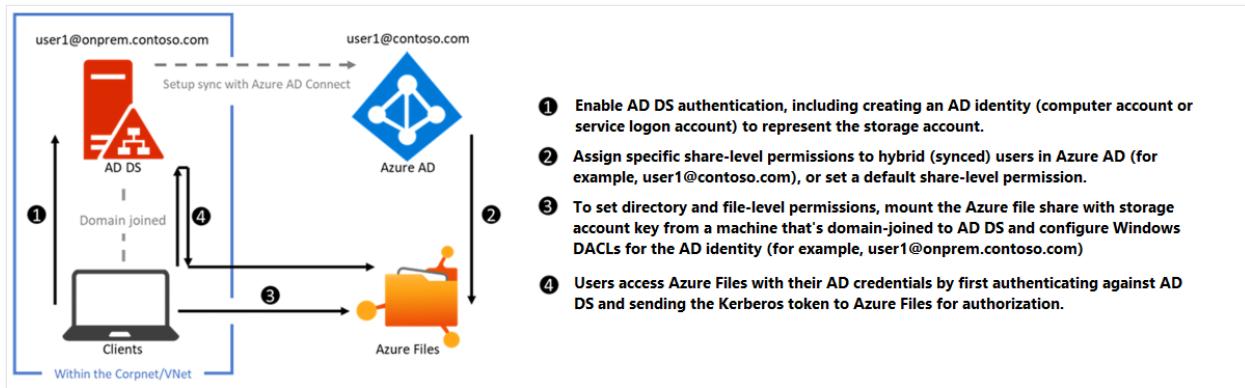
You can enable identity-based authentication on your new and existing storage accounts using one of three AD sources: AD DS, Microsoft Entra Domain Services, or Microsoft Entra Kerberos (hybrid identities only). Only one AD source can be used for file access authentication on the storage account, which applies to all file shares in the account. Before you can enable identity-based authentication on your storage account, you must first set up your domain environment.

AD DS

For on-premises AD DS authentication, you must set up your AD domain controllers and domain-join your machines or VMs. You can host your domain controllers on Azure VMs or on-premises. Either way, your domain-joined clients must have unimpeded network connectivity to the domain controller, so they must be within the corporate network or virtual network (VNET) of your domain service.

The following diagram depicts on-premises AD DS authentication to Azure file shares over SMB. The on-premises AD DS must be synced to Microsoft Entra ID using Microsoft Entra Connect Sync or Microsoft Entra Connect cloud sync. Only [hybrid user identities](#) that exist in both on-premises AD DS and Microsoft Entra ID can be authenticated and authorized for Azure file share access. This is because the share-level permission is configured against the identity represented in Microsoft Entra ID, whereas the

directory/file-level permission is enforced with that in AD DS. Make sure that you configure the permissions correctly against the same hybrid user.



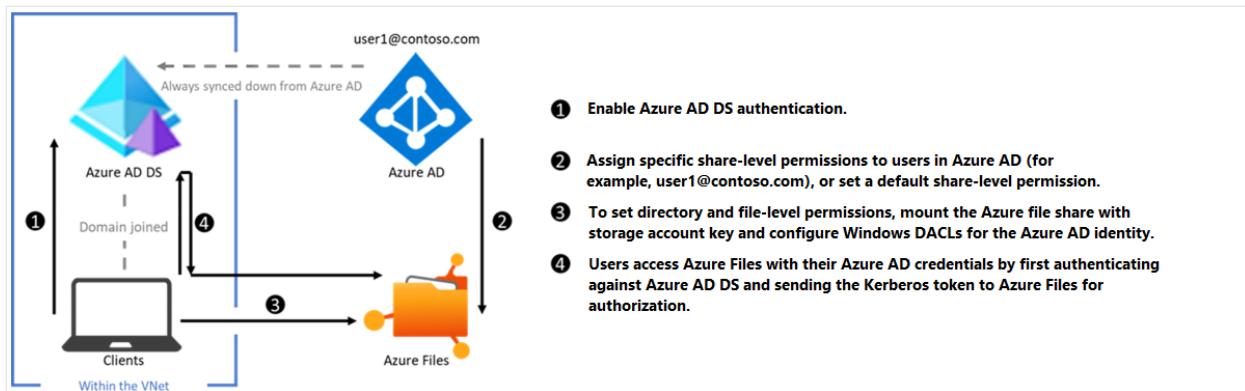
To learn how to enable AD DS authentication, first read [Overview - on-premises Active Directory Domain Services authentication over SMB for Azure file shares](#) and then see [Enable AD DS authentication for Azure file shares](#).

Microsoft Entra Domain Services

For Microsoft Entra Domain Services authentication, you should enable Microsoft Entra Domain Services and domain-join the VMs you plan to access file data from. Your domain-joined VM must reside in the same virtual network (VNET) as your Microsoft Entra Domain Services.

The following diagram represents the workflow for Microsoft Entra Domain Services authentication to Azure file shares over SMB. It follows a similar pattern to on-premises AD DS authentication, but there are two major differences:

1. You don't need to create the identity in Microsoft Entra Domain Services to represent the storage account. This is performed by the enablement process in the background.
2. All users that exist in Microsoft Entra ID can be authenticated and authorized. The user can be cloud-only or hybrid. The sync from Microsoft Entra ID to Microsoft Entra Domain Services is managed by the platform without requiring any user configuration. However, the client must be joined to the Microsoft Entra Domain Services hosted domain. It can't be Microsoft Entra joined or registered. Microsoft Entra Domain Services doesn't support non-Azure clients (i.e. user laptops, workstations, VMs in other clouds, etc.) being domain-joined to the Microsoft Entra Domain Services hosted domain. However, it's possible to mount a file share from a non-domain-joined client by providing explicit credentials such as `DOMAINNAME\username` or using the fully qualified domain name (`username@FQDN`).



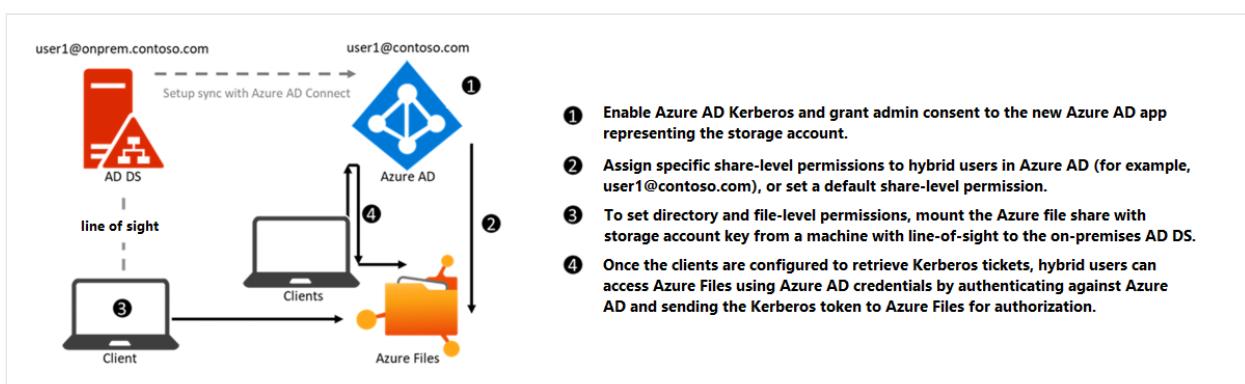
To learn how to enable Microsoft Entra Domain Services authentication, see [Enable Microsoft Entra Domain Services authentication on Azure Files](#).

Microsoft Entra Kerberos for hybrid identities

Enabling and configuring Microsoft Entra ID for authenticating [hybrid user identities](#) allows Microsoft Entra users to access Azure file shares using Kerberos authentication. This configuration uses Microsoft Entra ID to issue the necessary Kerberos tickets to access the file share with the industry-standard SMB protocol. This means your end users can access Azure file shares over the internet without requiring network connectivity to domain controllers from Microsoft Entra hybrid joined and Microsoft Entra joined VMs. However, configuring directory and file-level permissions for users and groups requires unimpeded network connectivity to the on-premises domain controller.

Important

Microsoft Entra Kerberos authentication only supports hybrid user identities; it doesn't support cloud-only identities. A traditional AD DS deployment is required, and it must be synced to Microsoft Entra ID using Microsoft Entra Connect Sync or Microsoft Entra Connect cloud sync. Clients must be Microsoft Entra joined or **Microsoft Entra hybrid joined**. Microsoft Entra Kerberos isn't supported on clients joined to Microsoft Entra Domain Services or joined to AD only.



To learn how to enable Microsoft Entra Kerberos authentication for hybrid identities, see [Enable Microsoft Entra Kerberos authentication for hybrid identities on Azure Files](#).

You can also use this feature to store FSLogix profiles on Azure file shares for Microsoft Entra joined VMs. For more information, see [Create a profile container with Azure Files and Microsoft Entra ID](#).

Access control

Azure Files enforces authorization on user access to both the share level and the directory/file levels. Share-level permission assignment can be performed on Microsoft Entra users or groups managed through Azure RBAC. With Azure RBAC, the credentials you use for file access should be available or synced to Microsoft Entra ID. You can assign Azure built-in roles like **Storage File Data SMB Share Reader** to users or groups in Microsoft Entra ID to grant access to an Azure file share.

At the directory/file level, Azure Files supports preserving, inheriting, and enforcing [Windows ACLs](#) just like any Windows file server. You can choose to keep Windows ACLs when copying data over SMB between your existing file share and your Azure file shares. Whether you plan to enforce authorization or not, you can use Azure file shares to back up ACLs along with your data.

Configure share-level permissions for Azure Files

Once you've enabled an AD source on your storage account, you must do one of the following to access the file share:

- Set a default share-level permission that applies to all authenticated users and groups
- Assign built-in Azure RBAC roles to users and groups, or
- Configure custom roles for Microsoft Entra identities and assign access rights to file shares in your storage account.

The assigned share-level permission allows the granted identity to get access to the share only, nothing else, not even the root directory. You still need to separately configure directory and file-level permissions.

Configure directory or file-level permissions for Azure Files

Azure file shares enforce standard Windows ACLs at both the directory and file level, including the root directory. Configuration of directory or file-level permissions is supported over both SMB and REST. Mount the target file share from your VM and configure permissions using Windows File Explorer, Windows [icacls](#), or the [Set-ACL](#) command.

Use the storage account key for superuser permissions

A user with the storage account key can access Azure file shares with superuser permissions. Superuser permissions bypass all access control restrictions.

 **Important**

Our recommended security best practice is to avoid sharing your storage account keys and leverage identity-based authentication whenever possible.

Preserve directory and file ACLs when importing data to Azure file shares

Azure Files supports preserving directory or file level ACLs when copying data to Azure file shares. You can copy ACLs on a directory or file to Azure file shares using either Azure File Sync or common file movement toolsets. For example, you can use [robocopy](#) with the `/copy:s` flag to copy data as well as ACLs to an Azure file share. ACLs are preserved by default, so you don't need to enable identity-based authentication on your storage account to preserve ACLs.

Pricing

There's no additional service charge to enable identity-based authentication over SMB on your storage account. For more information on pricing, see [Azure Files pricing](#) and [Microsoft Entra Domain Services pricing](#).

Next steps

For more information about Azure Files and identity-based authentication over SMB, see these resources:

- [Planning for an Azure Files deployment](#)

- Overview - on-premises Active Directory Domain Services authentication over SMB for Azure file shares
- Enable Microsoft Entra Domain Services authentication on Azure Files
- Enable Microsoft Entra Kerberos authentication for hybrid identities on Azure Files
- Enable AD Kerberos authentication for Linux clients
- FAQ

Azure Files networking considerations

Article • 11/21/2022

You can access your Azure file shares over the public internet accessible endpoint, over one or more private endpoints on your network(s), or by caching your Azure file share on-premises with Azure File Sync (SMB file shares only). This article focuses on how to configure Azure Files for direct access over public and/or private endpoints. To learn how to cache your Azure file share on-premises with Azure File Sync, see [Introduction to Azure File Sync](#).

We recommend reading [Planning for an Azure Files deployment](#) prior to reading this conceptual guide.

Directly accessing the Azure file share often requires additional thought with respect to networking:

- SMB file shares communicate over port 445, which many organizations and internet service providers (ISPs) block for outbound (internet) traffic. This practice originates from legacy security guidance about deprecated and non-internet safe versions of the SMB protocol. Although SMB 3.x is an internet-safe protocol, organizational or ISP policies may not be possible to change. Therefore, mounting an SMB file share often requires additional networking configuration to use outside of Azure.
- NFS file shares rely on network-level authentication and are therefore only accessible via restricted networks. Using an NFS file share always requires some level of networking configuration.

Configuring public and private endpoints for Azure Files is done on the top-level management object for Azure Files, the Azure storage account. A storage account is a management construct that represents a shared pool of storage in which you can deploy multiple Azure file shares, as well as the storage resources for other Azure storage services, such as blob containers or queues.



This video is a guide and demo for how to securely expose Azure file shares directly to information workers and apps in five simple steps. The sections below provide links and additional context to the documentation referenced in the video.

Applies to

File share type	SMB	NFS
Standard file shares (GPv2), LRS/ZRS	✓	✗
Standard file shares (GPv2), GRS/GZRS	✓	✗
Premium file shares (FileStorage), LRS/ZRS	✓	✓

Secure transfer

By default, Azure storage accounts require secure transfer, regardless of whether data is accessed over the public or private endpoint. For Azure Files, the **require secure transfer** setting is enforced for all protocol access to the data stored on Azure file shares, including SMB, NFS, and FileREST. You can disable the **require secure transfer** setting to allow unencrypted traffic. In the Azure portal, you may also see this setting labeled as **require secure transfer for REST API operations**.

The SMB, NFS, and FileREST protocols have slightly different behavior with respect to the **require secure transfer** setting:

- When **require secure transfer** is enabled on a storage account, all SMB file shares in that storage account will require the SMB 3.x protocol with AES-128-CCM, AES-

128-GCM, or AES-256-GCM encryption algorithms, depending on the available/required encryption negotiation between the SMB client and Azure Files. You can toggle which SMB encryption algorithms are allowed via the [SMB security settings](#). Disabling the **require secure transfer** setting enables SMB 2.1 and SMB 3.x mounts without encryption.

- NFS file shares don't support an encryption mechanism, so in order to use the NFS protocol to access an Azure file share, you must disable **require secure transfer** for the storage account.
- When secure transfer is required, the FileREST protocol may only be used with HTTPS. FileREST is only supported on SMB file shares today.

Public endpoint

The public endpoint for the Azure file shares within a storage account is an internet exposed endpoint. The public endpoint is the default endpoint for a storage account, however, it can be disabled if desired.

The SMB, NFS, and FileREST protocols can all use the public endpoint. However, each has slightly different rules for access:

- SMB file shares are accessible from anywhere in the world via the storage account's public endpoint with SMB 3.x with encryption. This means that authenticated requests, such as requests authorized by a user's logon identity, can originate securely from inside or outside of the Azure region. If SMB 2.1 or SMB 3.x without encryption is desired, two conditions must be met:
 1. The storage account's **require secure transfer** setting must be disabled.
 2. The request must originate from inside of the Azure region. As previously mentioned, encrypted SMB requests are allowed from anywhere, inside or outside of the Azure region.
- NFS file shares are accessible from the storage account's public endpoint if and only if the storage account's public endpoint is restricted to specific virtual networks using *service endpoints*. See [public endpoint firewall settings](#) for additional information on *service endpoints*.
- FileREST is accessible via the public endpoint. If secure transfer is required, only HTTPS requests are accepted. If secure transfer is disabled, HTTP requests are accepted by the public endpoint regardless of origin.

Public endpoint firewall settings

The storage account firewall restricts access to the public endpoint for a storage account. Using the storage account firewall, you can restrict access to certain IP addresses/IP address ranges, to specific virtual networks, or disable the public endpoint entirely.

When you restrict the traffic of the public endpoint to one or more virtual networks, you are using a capability of the virtual network called *service endpoints*. Requests directed to the service endpoint of Azure Files are still going to the storage account public IP address; however, the networking layer is doing additional verification of the request to validate that it is coming from an authorized virtual network. The SMB, NFS, and FileREST protocols all support service endpoints. Unlike SMB and FileREST, however, NFS file shares can only be accessed with the public endpoint through use of a *service endpoint*.

To learn more about how to configure the storage account firewall, see [configure Azure storage firewalls and virtual networks](#).

Public endpoint network routing

Azure Files supports multiple network routing options. The default option, Microsoft routing, works with all Azure Files configurations. The internet routing option does not support AD domain join scenarios or Azure File Sync.

Private endpoints

In addition to the default public endpoint for a storage account, Azure Files provides the option to have one or more private endpoints. A private endpoint is an endpoint that is only accessible within an Azure virtual network. When you create a private endpoint for your storage account, your storage account gets a private IP address from within the address space of your virtual network, much like how an on-premises file server or NAS device receives an IP address within the dedicated address space of your on-premises network.

An individual private endpoint is associated with a specific Azure virtual network subnet. A storage account may have private endpoints in more than one virtual network.

Using private endpoints with Azure Files enables you to:

- Securely connect to your Azure file shares from on-premises networks using a VPN or ExpressRoute connection with private-peering.

- Secure your Azure file shares by configuring the storage account firewall to block all connections on the public endpoint. By default, creating a private endpoint does not block connections to the public endpoint.
- Increase security for the virtual network by enabling you to block exfiltration of data from the virtual network (and peering boundaries).

To create a private endpoint, see [Configuring private endpoints for Azure Files](#).

Tunneling traffic over a virtual private network or ExpressRoute

To use private endpoints to access SMB or NFS file shares from on-premises, you must establish a network tunnel between your on-premises network and Azure. A [virtual network](#), or VNet, is similar to a traditional on-premises network. Like an Azure storage account or an Azure VM, a VNet is an Azure resource that is deployed in a resource group.

Azure Files supports the following mechanisms to tunnel traffic between your on-premises workstations and servers and Azure SMB/NFS file shares:

- [Azure VPN Gateway](#): A VPN gateway is a specific type of virtual network gateway that is used to send encrypted traffic between an Azure virtual network and an alternate location (such as on-premises) over the internet. An Azure VPN Gateway is an Azure resource that can be deployed in a resource group alongside of a storage account or other Azure resources. VPN gateways expose two different types of connections:
 - [Point-to-Site \(P2S\) VPN](#) gateway connections, which are VPN connections between Azure and an individual client. This solution is primarily useful for devices that are not part of your organization's on-premises network. A common use case is for telecommuters who want to be able to mount their Azure file share from home, a coffee shop, or hotel while on the road. To use a P2S VPN connection with Azure Files, you'll need to configure a P2S VPN connection for each client that wants to connect. To simplify the deployment of a P2S VPN connection, see [Configure a Point-to-Site \(P2S\) VPN on Windows for use with Azure Files](#) and [Configure a Point-to-Site \(P2S\) VPN on Linux for use with Azure Files](#).
 - [Site-to-Site \(S2S\) VPN](#), which are VPN connections between Azure and your organization's network. A S2S VPN connection enables you to configure a VPN connection once for a VPN server or device hosted on your organization's network, rather than configuring a connection for every client device that needs

to access your Azure file share. To simplify the deployment of a S2S VPN connection, see [Configure a Site-to-Site \(S2S\) VPN for use with Azure Files](#).

- [ExpressRoute](#), which enables you to create a defined route between Azure and your on-premises network that doesn't traverse the internet. Because ExpressRoute provides a dedicated path between your on-premises datacenter and Azure, ExpressRoute may be useful when network performance is a consideration. ExpressRoute is also a good option when your organization's policy or regulatory requirements require a deterministic path to your resources in the cloud.

 **Note**

Although we recommend using private endpoints to assist in extending your on-premises network into Azure, it is technically possible to route to the public endpoint over the VPN connection. However, this requires hard-coding the IP address for the public endpoint for the Azure storage cluster that serves your storage account. Because storage accounts may be moved between storage clusters at any time and new clusters are frequently added and removed, this requires regularly hard-coding all the possible Azure storage IP addresses into your routing rules.

DNS configuration

When you create a private endpoint, by default we also create a (or update an existing) private DNS zone corresponding to the `privatelink` subdomain. Strictly speaking, creating a private DNS zone is not required to use a private endpoint for your storage account. However, it is highly recommended in general and explicitly required when mounting your Azure file share with an Active Directory user principal or accessing it from the FileREST API.

 **Note**

This article uses the storage account DNS suffix for the Azure Public regions, `core.windows.net`. This commentary also applies to Azure Sovereign clouds such as the Azure US Government cloud and the Azure China cloud - just substitute the appropriate suffixes for your environment.

In your private DNS zone, we create an A record for `storageaccount.privatelink.file.core.windows.net` and a CNAME record for the regular name of the storage account, which follows the pattern

`storageaccount.file.core.windows.net`. Because your Azure private DNS zone is connected to the virtual network containing the private endpoint, you can observe the DNS configuration by calling the `Resolve-DnsName` cmdlet from PowerShell in an Azure VM (alternately `nslookup` in Windows and Linux):

PowerShell

```
Resolve-DnsName -Name "storageaccount.file.core.windows.net"
```

For this example, the storage account `storageaccount.file.core.windows.net` resolves to the private IP address of the private endpoint, which happens to be `192.168.0.4`.

Output

Name	Type	TTL	Section	NameHost
---	---	---	-----	-----
storageaccount.file.core.windows.net	CNAME	29	Answer	
csostoracct.privatelink.file.core.windows.net				
Name : storageaccount.privatelink.file.core.windows.net				
QueryType : A				
TTL : 1769				
Section : Answer				
IP4Address : 192.168.0.4				
Name : privatelink.file.core.windows.net				
QueryType : SOA				
TTL : 269				
Section : Authority				
NameAdministrator : azureprivatedns-host.microsoft.com				
SerialNumber : 1				
TimeToZoneRefresh : 3600				
TimeToZoneFailureRetry : 300				
TimeToExpiration : 2419200				
DefaultTTL : 300				

If you run the same command from on-premises, you'll see that the same storage account name resolves to the public IP address of the storage account instead; `storageaccount.file.core.windows.net` is a CNAME record for `storageaccount.privatelink.file.core.windows.net`, which in turn is a CNAME record for the Azure storage cluster hosting the storage account:

Output

Name	Type	TTL	Section	NameHost
storageaccount.file.core.windows.net	CNAME	60	Answer	
storageaccount.privatelink.file.core.windows.net	CNAME	60	Answer	
file.par20prdstr01a.store.core.windows.net				ore.windows.net
Name : file.par20prdstr01a.store.core.windows.net				
QueryType : A				
TTL : 60				
Section : Answer				
IP4Address : 52.239.194.40				

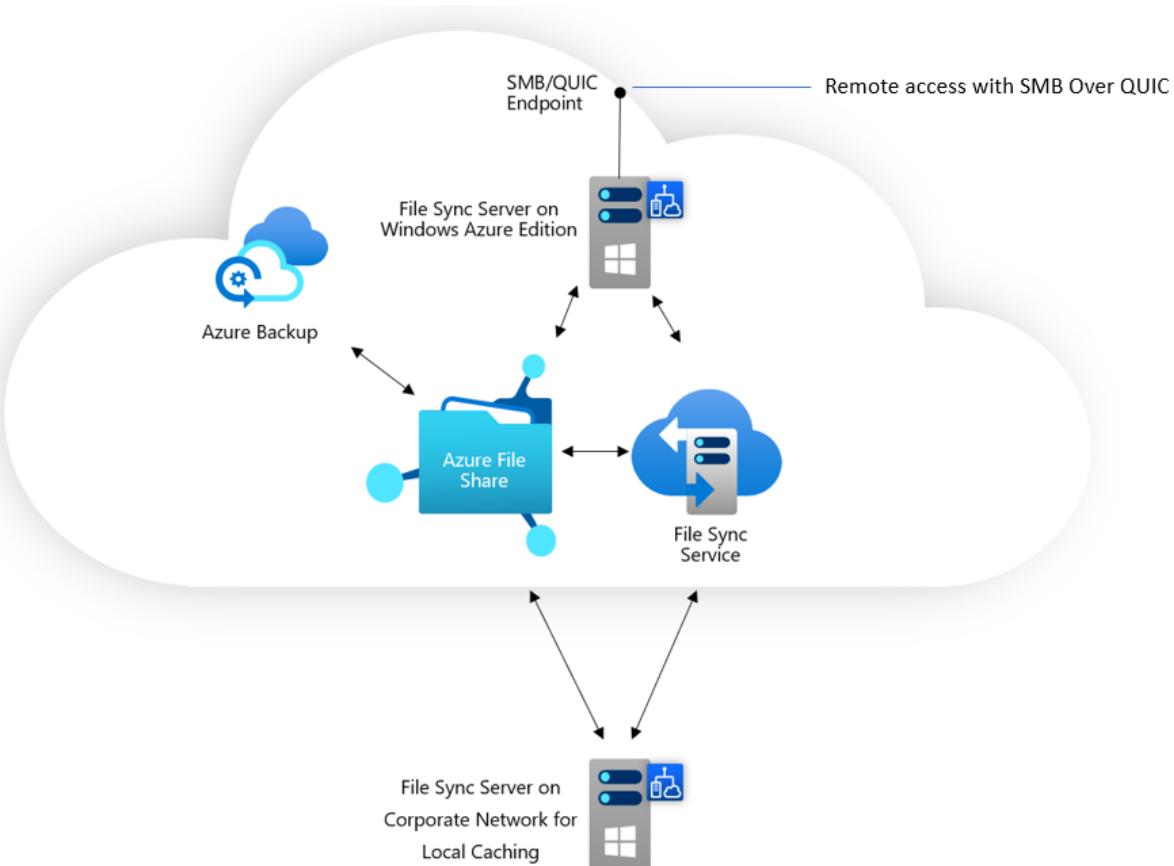
This reflects the fact that the storage account can expose both the public endpoint and one or more private endpoints. To ensure that the storage account name resolves to the private endpoint's private IP address, you must change the configuration on your on-premises DNS servers. This can be accomplished in several ways:

- Modifying the *hosts* file on your clients to make `storageaccount.file.core.windows.net` resolve to the desired private endpoint's private IP address. This is strongly discouraged for production environments, because you will need to make these changes to every client that wants to mount your Azure file shares, and changes to the storage account or private endpoint will not be automatically handled.
- Creating an A record for `storageaccount.file.core.windows.net` in your on-premises DNS servers. This has the advantage that clients in your on-premises environment will be able to automatically resolve the storage account without needing to configure each client. However, this solution is similarly brittle to modifying the *hosts* file because changes are not reflected. Although this solution is brittle, it may be the best choice for some environments.
- Forward the `core.windows.net` zone from your on-premises DNS servers to your Azure private DNS zone. The Azure private DNS host can be reached through a special IP address (`168.63.129.16`) that is only accessible inside virtual networks that are linked to the Azure private DNS zone. To work around this limitation, you can run additional DNS servers within your virtual network that will forward `core.windows.net` on to the Azure private DNS zone. To simplify this set up, we have provided PowerShell cmdlets that will auto-deploy DNS servers in your Azure virtual network and configure them as desired. To learn how to set up DNS forwarding, see [Configuring DNS with Azure Files](#).

SMB over QUIC

Windows Server 2022 Azure Edition supports a new transport protocol called QUIC for the SMB server provided by the File Server role. QUIC is a replacement for TCP that is built on top of UDP, providing numerous advantages over TCP while still providing a reliable transport mechanism. One key advantage for the SMB protocol is that instead of using port 445, all transport is done over port 443, which is widely open outbound to support HTTPS. This effectively means that SMB over QUIC offers an "SMB VPN" for file sharing over the public internet. Windows 11 ships with an SMB over QUIC capable client.

At this time, Azure Files doesn't directly support SMB over QUIC. However, you can get access to Azure file shares via Azure File Sync running on Windows Server as in the diagram below. This also gives you the option to have Azure File Sync caches both on-premises or in different Azure datacenters to provide local caches for a distributed workforce. To learn more about this option, see [Deploy Azure File Sync](#) and [SMB over QUIC](#).



See also

- [Azure Files overview](#)
- [Planning for an Azure Files deployment](#)

Azure storage disaster recovery planning and failover

Article • 01/11/2024

Microsoft strives to ensure that Azure services are always available. However, unplanned service outages may occur. Key components of a good disaster recovery plan include strategies for:

- [Data protection](#)
- [Backup and restore](#)
- [Data redundancy](#)
- [Failover](#)
- [Designing applications for high availability](#)

This article focuses on failover for globally redundant storage accounts (GRS, GZRS, and RA-GZRS), and how to design your applications to be highly available if there's an outage and subsequent failover.

Choose the right redundancy option

Azure Storage maintains multiple copies of your storage account to ensure durability and high availability. Which redundancy option you choose for your account depends on the degree of resiliency you need for your applications.

With locally redundant storage (LRS), three copies of your storage account are automatically stored and replicated within a single datacenter. With zone-redundant storage (ZRS), a copy is stored and replicated in each of three separate availability zones within the same region. For more information about availability zones, see [Azure availability zones](#).

Recovery of a single copy of a storage account occurs automatically with LRS and ZRS.

Globally redundant storage and failover

With globally redundant storage (GRS, GZRS, and RA-GZRS), Azure copies your data asynchronously to a secondary geographic region at least hundreds of miles away. This allows you to recover your data if there's an outage in the primary region. A feature that distinguishes globally redundant storage from LRS and ZRS is the ability to fail over to the secondary region if there's an outage in the primary region. The process of failing over updates the DNS entries for your storage account service endpoints such that the

endpoints for the secondary region become the new primary endpoints for your storage account. Once the failover is complete, clients can begin writing to the new primary endpoints.

RA-GRS and RA-GZRS redundancy configurations provide geo-redundant storage with the added benefit of read access to the secondary endpoint if there is an outage in the primary region. If an outage occurs in the primary endpoint, applications configured for read access to the secondary region and designed for high availability can continue to read from the secondary endpoint. Microsoft recommends RA-GZRS for maximum availability and durability of your storage accounts.

For more information about redundancy in Azure Storage, see [Azure Storage redundancy](#).

Plan for storage account failover

Azure Storage accounts support two types of failover:

- **Customer-managed failover** - Customers can manage storage account failover if there's an unexpected service outage.
- **Microsoft-managed failover** - Potentially initiated by Microsoft only in the case of a severe disaster in the primary region.^{1,2}

¹Microsoft-managed failover can't be initiated for individual storage accounts, subscriptions, or tenants. For more details see [Microsoft-managed failover](#).

² Your disaster recovery plan should be based on customer-managed failover. **Do not** rely on Microsoft-managed failover, which would only be used in extreme circumstances.

Each type of failover has a unique set of use cases, corresponding expectations for data loss, and support for accounts with a hierarchical namespace enabled (Azure Data Lake Storage Gen2). This table summarizes those aspects of each type of failover :

Expand table

Type	Failover Scope	Use case	Expected data loss	HNS supported
Customer-managed	Storage account	<p>The storage service endpoints for the primary region become unavailable, but the secondary region is available.</p> <p>You received an Azure Advisory in which Microsoft advises you to perform</p>	Yes	Yes (<i>In preview</i>)

Type	Failover Scope	Use case	Expected data loss	HNS supported
		a failover operation of storage accounts potentially affected by an outage.		
Microsoft-managed	Entire region or scale unit	The primary region becomes completely unavailable due to a significant disaster, but the secondary region is available.	Yes	Yes

Customer-managed failover

If the data endpoints for the storage services in your storage account become unavailable in the primary region, you can fail over to the secondary region. After the failover is complete, the secondary region becomes the new primary and users can proceed to access data in the new primary region.

To fully understand the impact that customer-managed account failover would have on your users and applications, it is helpful to know what happens during every step of the failover and fallback process. For details about how the process works, see [How customer-managed storage account failover works](#).

Microsoft-managed failover

In extreme circumstances where the original primary region is deemed unrecoverable within a reasonable amount of time due to a major disaster, Microsoft **may** initiate a regional failover. In this case, no action on your part is required. Until the Microsoft-managed failover has completed, you won't have write access to your storage account. Your applications can read from the secondary region if your storage account is configured for RA-GRS or RA-GZRS.

ⓘ Important

Your disaster recovery plan should be based on customer-managed failover. **Do not** rely on Microsoft-managed failover, which might only be used in extreme circumstances. A Microsoft-managed failover would be initiated for an entire physical unit, such as a region or scale unit. It can't be initiated for individual storage accounts, subscriptions, or tenants. For the ability to selectively failover your individual storage accounts, use [customer-managed account failover](#).

Anticipate data loss and inconsistencies

✖ Caution

Storage account failover usually involves some data loss, and potentially file and data inconsistencies. In your disaster recovery plan, it's important to consider the impact that an account failover would have on your data before initiating one.

Because data is written asynchronously from the primary region to the secondary region, there's always a delay before a write to the primary region is copied to the secondary. If the primary region becomes unavailable, the most recent writes may not yet have been copied to the secondary.

When a failover occurs, all data in the primary region is lost as the secondary region becomes the new primary. All data already copied to the secondary is maintained when the failover happens. However, any data written to the primary that hasn't also been copied to the secondary region is lost permanently.

The new primary region is configured to be locally redundant (LRS) after the failover.

You also might experience file or data inconsistencies if your storage accounts have one or more of the following enabled:

- [Hierarchical namespace \(Azure Data Lake Storage Gen2\)](#)
- [Change feed](#)
- [Point-in-time restore for block blobs](#)

Last sync time

The **Last Sync Time** property indicates the most recent time that data from the primary region is guaranteed to have been written to the secondary region. For accounts that have a hierarchical namespace, the same **Last Sync Time** property also applies to the metadata managed by the hierarchical namespace, including ACLs. All data and metadata written prior to the last sync time is available on the secondary, while data and metadata written after the last sync time may not have been written to the secondary, and may be lost. Use this property if there's an outage to estimate the amount of data loss you may incur by initiating an account failover.

As a best practice, design your application so that you can use the last sync time to evaluate expected data loss. For example, if you're logging all write operations, then you can compare the time of your last write operations to the last sync time to determine which writes haven't been synced to the secondary.

For more information about checking the **Last Sync Time** property, see [Check the Last Sync Time property for a storage account](#).

File consistency for Azure Data Lake Storage Gen2

Replication for storage accounts with a [hierarchical namespace enabled \(Azure Data Lake Storage Gen2\)](#) occurs at the file level. This means if an outage in the primary region occurs, it is possible that only some of the files in a container or directory might have successfully replicated to the secondary region. Consistency for all files in a container or directory after a storage account failover is not guaranteed.

Change feed and blob data inconsistencies

Storage account failover of geo-redundant storage accounts with [change feed](#) enabled may result in inconsistencies between the change feed logs and the blob data and/or metadata. Such inconsistencies can result from the asynchronous nature of both updates to the change logs and the replication of blob data from the primary to the secondary region. The only situation in which inconsistencies would not be expected is when all of the current log records have been successfully flushed to the log files, and all of the storage data has been successfully replicated from the primary to the secondary region.

For information about how change feed works see [How the change feed works](#).

Keep in mind that other storage account features require the change feed to be enabled such as [operational backup of Azure Blob Storage](#), [Object replication](#) and [Point-in-time restore for block blobs](#).

Point-in-time restore inconsistencies

Customer-managed failover is supported for general-purpose v2 standard tier storage accounts that include block blobs. However, performing a customer-managed failover on a storage account resets the earliest possible restore point for the account. Data for [Point-in-time restore for block blobs](#) is only consistent up to the failover completion time. As a result, you can only restore block blobs to a point in time no earlier than the failover completion time. You can check the failover completion time in the redundancy tab of your storage account in the Azure Portal.

For example, suppose you have set the retention period to 30 days. If more than 30 days have elapsed since the failover, then you can restore to any point within that 30 days. However, if fewer than 30 days have elapsed since the failover, then you can't restore to

a point prior to the failover, regardless of the retention period. For example, if it's been 10 days since the failover, then the earliest possible restore point is 10 days in the past, not 30 days in the past.

The time and cost of failing over

The time it takes for failover to complete after being initiated can vary, although it typically takes less than one hour.

A customer-managed failover loses its geo-redundancy after a failover (and failback). Your storage account is automatically converted to locally redundant storage (LRS) in the new primary region during a failover, and the storage account in the original primary region is deleted.

You can re-enable geo-redundant storage (GRS) or read-access geo-redundant storage (RA-GRS) for the account, but note that converting from LRS to GRS or RA-GRS incurs an additional cost. The cost is due to the network egress charges to re-replicate the data to the new secondary region. Also, all archived blobs need to be rehydrated to an online tier before the account can be configured for geo-redundancy, which will incur a cost.

For more information about pricing, see:

- [Bandwidth Pricing Details](#)
- [Azure Storage pricing](#)

After you re-enable GRS for your storage account, Microsoft begins replicating the data in your account to the new secondary region. Replication time depends on many factors, which include:

- The number and size of the objects in the storage account. Replicating many small objects can take longer than replicating fewer and larger objects.
- The available resources for background replication, such as CPU, memory, disk, and WAN capacity. Live traffic takes priority over geo replication.
- If your storage account contains blobs, the number of snapshots per blob.
- If your storage account contains tables, the [data partitioning strategy](#). The replication process can't scale beyond the number of partition keys that you use.

Supported storage account types

All geo-redundant offerings support Microsoft-managed failover. In addition, some account types support customer-managed account failover, as shown in the following table:

Type of failover	GRS/RA-GRS	GZRS/RA-GZRS
Customer-managed failover	General-purpose v2 accounts General-purpose v1 accounts Legacy Blob Storage accounts	General-purpose v2 accounts
Microsoft-managed failover	All account types	General-purpose v2 accounts

Classic storage accounts

ⓘ Important

Customer-managed account failover is only supported for storage accounts deployed using the Azure Resource Manager (ARM) deployment model. The Azure Service Manager (ASM) deployment model, also known as *classic*, isn't supported. To make classic storage accounts eligible for customer-managed account failover, they must first be [migrated to the ARM model](#). Your storage account must be accessible to perform the upgrade, so the primary region can't currently be in a failed state.

If there's a disaster that affects the primary region, Microsoft will manage the failover for classic storage accounts. For more information, see [Microsoft-managed failover](#).

Azure Data Lake Storage Gen2

ⓘ Important

Customer-managed account failover for accounts that have a hierarchical namespace (Azure Data Lake Storage Gen2) is currently in PREVIEW and only supported in the following regions:

- (Asia Pacific) Central India
- (Asia Pacific) South East Asia
- (Europe) North Europe
- (Europe) Switzerland North
- (Europe) Switzerland West
- (Europe) West Europe

- (North America) Canada Central
- (North America) East US 2
- (North America) South Central US

To opt in to the preview, see [Set up preview features in Azure subscription](#) and specify `AllowHNSAccountFailover` as the feature name.

See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

if there's a significant disaster that affects the primary region, Microsoft will manage the failover for accounts with a hierarchical namespace. For more information, see [Microsoft-managed failover](#).

Unsupported features and services

The following features and services aren't supported for account failover:

- Azure File Sync doesn't support storage account failover. Storage accounts containing Azure file shares being used as cloud endpoints in Azure File Sync shouldn't be failed over. Doing so will cause sync to stop working and may also cause unexpected data loss in the case of newly tiered files.
- A storage account containing premium block blobs can't be failed over. Storage accounts that support premium block blobs don't currently support geo-redundancy.
- Customer-managed failover isn't supported for either the source or the destination account in an [object replication policy](#).
- To failover an account with SSH File Transfer Protocol (SFTP) enabled, you must first [disable SFTP for the account](#). If you want to resume using SFTP after the failover is complete, simply [re-enable it](#).
- Network File System (NFS) 3.0 (NFSv3) isn't supported for storage account failover. You can't create a storage account configured for global-redundancy with NFSv3 enabled.

Failover is not for account migration

Storage account failover shouldn't be used as part of your data migration strategy. Failover is a temporary solution to a service outage. For information about how to migrate your storage accounts, see [Azure Storage migration overview](#).

Storage accounts containing archived blobs

Storage accounts containing archived blobs support account failover. However, after a [customer-managed failover](#) is complete, all archived blobs need to be rehydrated to an online tier before the account can be configured for geo-redundancy.

Storage resource provider

Microsoft provides two REST APIs for working with Azure Storage resources. These APIs form the basis of all actions you can perform against Azure Storage. The Azure Storage REST API enables you to work with data in your storage account, including blob, queue, file, and table data. The Azure Storage resource provider REST API enables you to manage the storage account and related resources.

After a failover is complete, clients can again read and write Azure Storage data in the new primary region. However, the Azure Storage resource provider does not fail over, so resource management operations must still take place in the primary region. If the primary region is unavailable, you will not be able to perform management operations on the storage account.

Because the Azure Storage resource provider does not fail over, the [Location](#) property will return the original primary location after the failover is complete.

Azure virtual machines

Azure virtual machines (VMs) don't fail over as part of an account failover. If the primary region becomes unavailable, and you fail over to the secondary region, then you will need to recreate any VMs after the failover. Also, there's a potential data loss associated with the account failover. Microsoft recommends following the [high availability](#) and [disaster recovery](#) guidance specific to virtual machines in Azure.

Keep in mind that any data stored in a temporary disk is lost when the VM is shut down.

Azure unmanaged disks

As a best practice, Microsoft recommends converting unmanaged disks to managed disks. However, if you need to fail over an account that contains unmanaged disks attached to Azure VMs, you will need to shut down the VM before initiating the failover.

Unmanaged disks are stored as page blobs in Azure Storage. When a VM is running in Azure, any unmanaged disks attached to the VM are leased. An account failover can't proceed when there's a lease on a blob. To perform the failover, follow these steps:

1. Before you begin, note the names of any unmanaged disks, their logical unit numbers (LUN), and the VM to which they are attached. Doing so will make it easier to reattach the disks after the failover.
2. Shut down the VM.
3. Delete the VM, but retain the VHD files for the unmanaged disks. Note the time at which you deleted the VM.
4. Wait until the **Last Sync Time** has updated, and is later than the time at which you deleted the VM. This step is important, because if the secondary endpoint hasn't been fully updated with the VHD files when the failover occurs, then the VM may not function properly in the new primary region.
5. Initiate the account failover.
6. Wait until the account failover is complete and the secondary region has become the new primary region.
7. Create a VM in the new primary region and reattach the VHDs.
8. Start the new VM.

Keep in mind that any data stored in a temporary disk is lost when the VM is shut down.

Copying data as an alternative to failover

If your storage account is configured for read access to the secondary region, then you can design your application to read from the secondary endpoint. If you prefer not to fail over if there's an outage in the primary region, you can use tools such as [AzCopy](#) or [Azure PowerShell](#) to copy data from your storage account in the secondary region to another storage account in an unaffected region. You can then point your applications to that storage account for both read and write availability.

Design for high availability

It's important to design your application for high availability from the start. Refer to these Azure resources for guidance in designing your application and planning for disaster recovery:

- [Designing resilient applications for Azure](#): An overview of the key concepts for architecting highly available applications in Azure.
- [Resiliency checklist](#): A checklist for verifying that your application implements the best design practices for high availability.
- [Use geo-redundancy to design highly available applications](#): Design guidance for building applications to take advantage of geo-redundant storage.
- [Tutorial: Build a highly available application with Blob storage](#): A tutorial that shows how to build a highly available application that automatically switches

between endpoints as failures and recoveries are simulated.

Keep in mind these best practices for maintaining high availability for your Azure Storage data:

- **Disks:** Use [Azure Backup](#) to back up the VM disks used by your Azure virtual machines. Also consider using [Azure Site Recovery](#) to protect your VMs if there's a regional disaster.
- **Block blobs:** Turn on [soft delete](#) to protect against object-level deletions and overwrites, or copy block blobs to another storage account in a different region using [AzCopy](#), [Azure PowerShell](#), or the [Azure Data Movement library](#).
- **Files:** Use [Azure Backup](#) to back up your file shares. Also enable [soft delete](#) to protect against accidental file share deletions. For geo-redundancy when GRS isn't available, use [AzCopy](#) or [Azure PowerShell](#) to copy your files to another storage account in a different region.
- **Tables:** use [AzCopy](#) to export table data to another storage account in a different region.

Track outages

Customers may subscribe to the [Azure Service Health Dashboard](#) to track the health and status of Azure Storage and other Azure services.

Microsoft also recommends that you design your application to prepare for the possibility of write failures. Your application should expose write failures in a way that alerts you to the possibility of an outage in the primary region.

See also

- [Use geo-redundancy to design highly available applications](#)
- [Tutorial: Build a highly available application with Blob storage](#)
- [Azure Storage redundancy](#)
- [How customer-managed storage account failover works](#)

About Azure file share backup

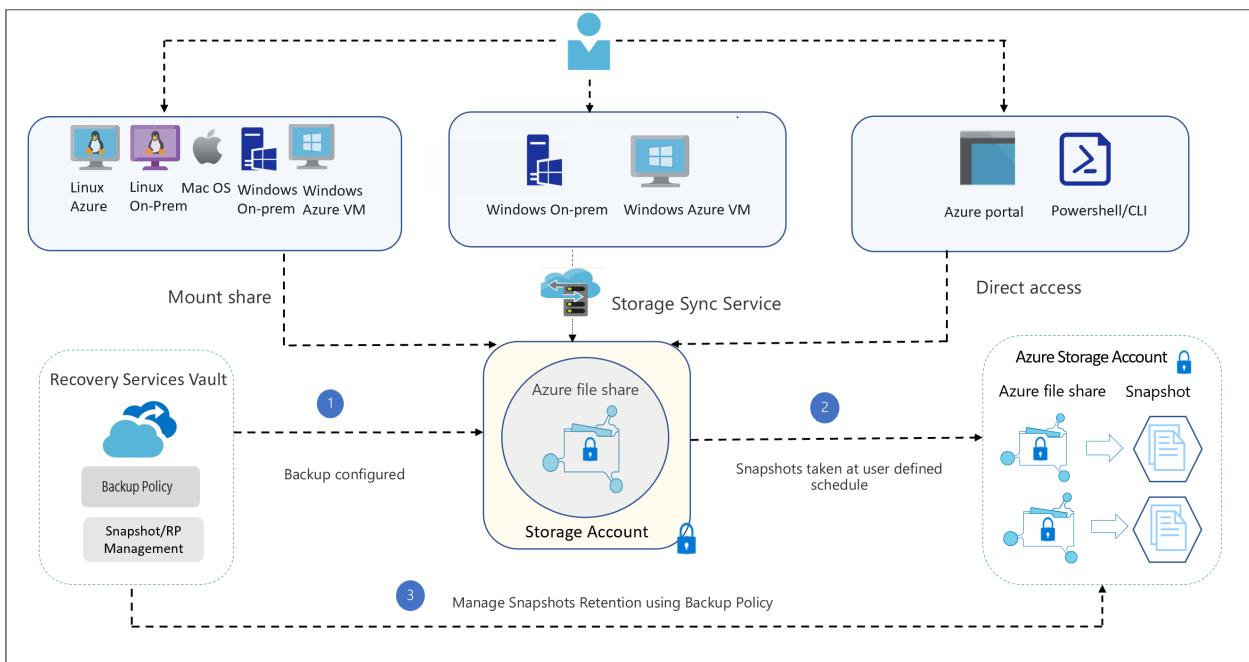
Article • 01/06/2023

Azure file share backup is a native, cloud based backup solution that protects your data in the cloud and eliminates additional maintenance overheads involved in on-premises backup solutions. The Azure Backup service smoothly integrates with Azure File Sync, and allows you to centralize your file share data as well as your backups. This simple, reliable, and secure solution enables you to configure protection for your enterprise file shares in a few simple steps with an assurance that you can recover your data if any accidental deletion.

Key benefits of Azure file share backup

- **Zero infrastructure:** No deployment is needed to configure protection for your file shares.
- **Customized retention:** You can configure backups with daily/weekly/monthly/yearly retention according to your requirements.
- **Built in management capabilities:** You can schedule backups and specify the desired retention period without the additional overhead of data pruning.
- **Instant restore:** Azure file share backup uses file share snapshots, so you can select just the files you want to restore instantly.
- **Alerting and reporting:** You can configure alerts for backup and restore failures and use the reporting solution provided by Azure Backup to get insights on backups across your files shares.
- **Protection against accidental deletion of file shares:** Azure Backup enables the [soft delete feature](#) on a storage account level with a retention period of 14 days. Even if a malicious actor deletes the file share, the file share's contents and recovery points (snapshots) are retained for a configurable retention period, allowing the successful and complete recovery of source contents and snapshots with no data loss.
- **Protection against accidental deletion of snapshots:** Azure Backup acquires a lease on the snapshots taken by scheduled/on-demand backup jobs. The lease acts as a lock that adds a layer of protection and secures the snapshots against accidental deletion.

Architecture



How the backup process works

1. The first step in configuring backup for Azure file shares is creating a Recovery Services vault. The vault gives you a consolidated view of the backups configured across different workloads.
2. Once you create a vault, the Azure Backup service discovers the storage accounts that can be registered with the vault. You can select the storage account hosting the file shares you want to protect.
3. After you select the storage account, the Azure Backup service lists the set of file shares present in the storage account and stores their names in the management layer catalog.
4. You then configure the backup policy (schedule and retention) according to your requirements, and select the file shares to back up. The Azure Backup service registers the schedules in the control plane to do scheduled backups.
5. Based on the policy specified, the Azure Backup scheduler triggers backups at the scheduled time. As part of that job, the file share snapshot is created using the File share API. Only the snapshot URL is stored in the metadata store.

⚠ Note

The file share data isn't transferred to the Backup service, since the Backup service creates and manages snapshots that are part of your storage account, and backups aren't transferred to the vault.

6. You can restore the Azure file share contents (individual files or the full share) from snapshots available on the source file share. Once the operation is triggered, the snapshot URL is retrieved from the metadata store and the data is listed and transferred from the source snapshot to the target file share of your choice.
7. If you're using Azure File Sync, the Backup service indicates to the Azure File Sync service the paths of the files being restored, which then triggers a background change detection process on these files. Any files that have changed are synced down to the server endpoint. This process happens in parallel with the original restore to the Azure file share.
8. The backup and restore job monitoring data is pushed to the Azure Backup Monitoring service. This allows you to monitor cloud backups for your file shares in a single dashboard. In addition, you can also configure alerts or email notifications when backup health is affected. Emails are sent via the Azure email service.

Backup costs

There are two costs associated with Azure file share backup solution:

1. **Snapshot storage cost:** Storage charges incurred for snapshots are billed along with Azure Files usage according to the pricing details mentioned [here](#)
2. **Protected Instance fee:** Starting from September 1, 2020, you're charged a protected instance fee as per the [pricing details](#). The protected instance fee depends on the total size of protected file shares in a storage account.

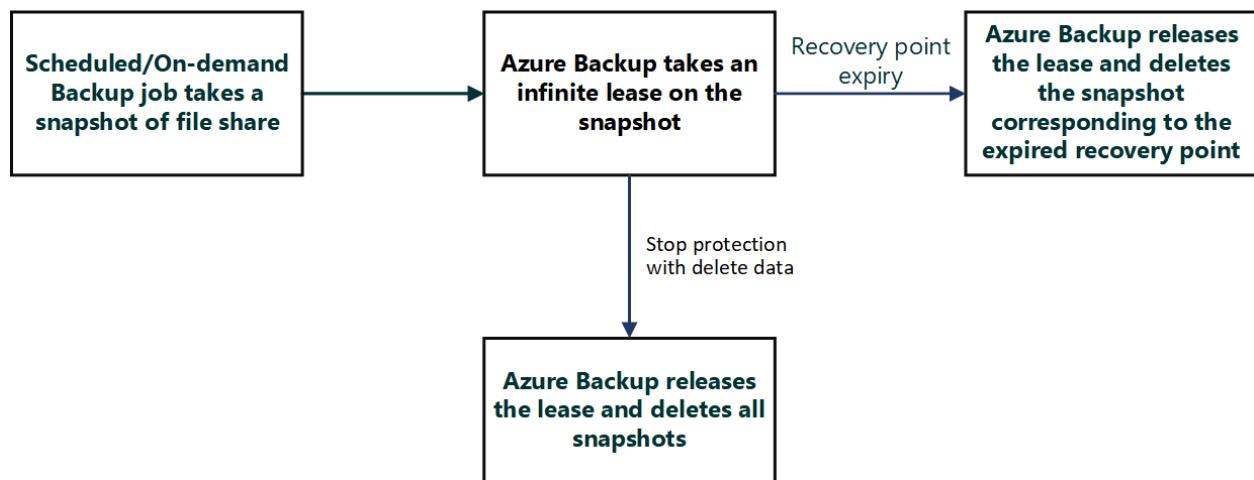
To get detailed estimates for backing up Azure file shares, you can download the detailed [Azure Backup pricing estimator](#).

How lease snapshot works?

When Azure Backup takes a snapshot, scheduled or on-demand, it adds a lock on the snapshot using the lease snapshot capability of the *Files* platform. The lock protects the snapshots from accidental deletion, and the lock's duration is infinite. If a file share has leased snapshots, the deletion is no more a one-click operation. Therefore, you also get protection against accidental deletion of the backed-up file share.

To protect a snapshot from deletion while restore operation is in progress, Azure Backup checks the lease status on the snapshot. If it's non-leased, it adds a lock by taking a lease on the snapshot.

The following diagram explains the lifecycle of the lease acquired by Azure Backup:



Next steps

- Learn how to [Back up Azure file shares](#)
- Find answers to [Questions about backing up Azure Files](#)

Authorize access to queues using Microsoft Entra ID

Article • 10/12/2023

Azure Storage supports using Microsoft Entra ID to authorize requests to queue data. With Microsoft Entra ID, you can use Azure role-based access control (Azure RBAC) to grant permissions to a security principal, which may be a user, group, or application service principal. The security principal is authenticated by Microsoft Entra ID to return an OAuth 2.0 token. The token can then be used to authorize a request against the Queue service.

Authorization with Microsoft Entra ID provides superior security and ease of use over Shared Key authorization. Microsoft recommends using Microsoft Entra authorization with your queue applications when possible to assure access with minimum required privileges.

Authorization with Microsoft Entra ID is available for all general-purpose storage accounts in all public regions and national clouds. Only storage accounts created with the Azure Resource Manager deployment model support Microsoft Entra authorization.

Overview of Microsoft Entra ID for queues

When a security principal (a user, group, or application) attempts to access a queue resource, the request must be authorized, unless it's a queue available for anonymous access. With Microsoft Entra ID, access to a resource is a two-step process:

1. First, the security principal's identity is authenticated and an OAuth 2.0 token is returned.

The authentication step requires that an application request an OAuth 2.0 access token at runtime. If an application is running from within an Azure entity such as an Azure VM, a Virtual Machine Scale Set, or an Azure Functions app, it can use a [managed identity](#) to access queue data.

2. Next, the token is passed as part of a request to the Queue service and used by the service to authorize access to the specified resource.

The authorization step requires that one or more Azure RBAC roles be assigned to the security principal making the request. For more information, see [Assign Azure roles for access rights](#).

Use a Microsoft Entra account with portal, PowerShell, or Azure CLI

To learn about how to access data in the Azure portal with a Microsoft Entra account, see [Data access from the Azure portal](#). To learn how to call Azure PowerShell or Azure CLI commands with a Microsoft Entra account, see [Data access from PowerShell or Azure CLI](#).

Use Microsoft Entra ID to authorize access in application code

To authorize access to Azure Storage with Microsoft Entra ID, you can use one of the following client libraries to acquire an OAuth 2.0 token:

- The Azure Identity client library is recommended for most development scenarios.
- The [Microsoft Authentication Library \(MSAL\)](#) may be suitable for certain advanced scenarios.

Azure Identity client library

The Azure Identity client library simplifies the process of getting an OAuth 2.0 access token for authorization with Microsoft Entra ID via the [Azure SDK](#). The latest versions of the Azure Storage client libraries for .NET, Java, Python, JavaScript, and Go integrate with the Azure Identity libraries for each of those languages to provide a simple and secure means to acquire an access token for authorization of Azure Storage requests.

An advantage of the Azure Identity client library is that it enables you to use the same code to acquire the access token whether your application is running in the development environment or in Azure. The Azure Identity client library returns an access token for a security principal. When your code is running in Azure, the security principal may be a managed identity for Azure resources, a service principal, or a user or group. In the development environment, the client library provides an access token for either a user or a service principal for testing purposes.

The access token returned by the Azure Identity client library is encapsulated in a token credential. You can then use the token credential to get a service client object to use in performing authorized operations against Azure Storage. A simple way to get the access token and token credential is to use the **DefaultAzureCredential** class that is provided by the Azure Identity client library. **DefaultAzureCredential** attempts to get the token credential by sequentially trying several different credential types.

DefaultAzureCredential works in both the development environment and in Azure.

The following table points to additional information for authorizing access to data in various scenarios:

Language	.NET	Java	JavaScript	Python	Go
Overview of auth with Microsoft Entra ID	How to authenticate .NET applications with Azure services	Azure authentication with Java and Azure Identity	Authenticate JavaScript apps to Azure using the Azure SDK	Authenticate Python apps to Azure using the Azure SDK	
Auth using developer service principals	Authenticate .NET apps to Azure services during local development using service principals	Azure authentication with service principal	Auth JS apps to Azure services with service principal	Authenticate Python apps to Azure services during local development using service principals	Azure SDK for Go authentication with a service principal
Auth using developer or user accounts	Authenticate .NET apps to Azure services during local development using developer accounts	Azure authentication with user credentials	Auth JS apps to Azure services with dev accounts	Authenticate Python apps to Azure services during local development using developer accounts	Azure authentication with the Azure SDK for Go
Auth from Azure-hosted apps	Authenticating Azure-hosted apps to Azure resources with the Azure SDK for .NET	Authenticate Azure-hosted Java applications	Authenticating Azure-hosted JavaScript apps to Azure resources with the Azure SDK for JavaScript	Authenticating Azure-hosted apps to Azure resources with the Azure SDK for Python	Authentication with the Azure SDK for Go using a managed identity
Auth from on-premises apps	Authenticate to Azure resources from .NET apps hosted on-premises		Authenticate on-premises JavaScript apps to Azure resources	Authenticate to Azure resources from Python apps hosted on-premises	
Identity client library overview	Azure Identity client library for .NET	Azure Identity client library for Java	Azure Identity client library for JavaScript	Azure Identity client library for Python	Azure Identity client library for Go ↗

Microsoft Authentication Library (MSAL)

While Microsoft recommends using the Azure Identity client library when possible, the MSAL library may be appropriate to use in certain advanced scenarios. For more information, see [Learn about MSAL](#).

When you use MSAL to acquire an OAuth token for access to Azure Storage, you need to provide a Microsoft Entra resource ID. The Microsoft Entra resource ID indicates the audience for which a token that is issued can be used to provide access to an Azure resource. In the case of Azure Storage, the resource ID may be specific to a single storage account, or it may apply to any storage account.

When you provide a resource ID that is specific to a single storage account and service, the resource ID is used to acquire a token for authorizing requests to the specified account and service only. The following table lists the value to use for the resource ID, based on the cloud you're working with. Replace `<account-name>` with the name of your storage account.

Cloud	Resource ID
Azure Global	<code>https://<account-name>.queue.core.windows.net</code>
Azure Government	<code>https://<account-name>.queue.core.usgovcloudapi.net</code>
Azure China 21Vianet	<code>https://<account-name>.queue.core.chinacloudapi.cn</code>

You can also provide a resource ID that applies to any storage account, as shown in the following table. This resource ID is the same for all public and sovereign clouds, and is used to acquire a token for authorizing requests to any storage account.

Cloud	Resource ID
Azure Global	<code>https://storage.azure.com/</code>
Azure Government	<code>https://storage.azure.com/</code>
Azure China 21Vianet	<code>https://storage.azure.com/</code>

Assign Azure roles for access rights

Microsoft Entra authorizes access rights to secured resources through Azure RBAC. Azure Storage defines a set of built-in RBAC roles that encompass common sets of permissions used to access queue data. You can also define custom roles for access to queue data. To learn more about assigning Azure roles for queue access, see [Assign an Azure role for access to queue data](#).

A Microsoft Entra security principal may be a user, a group, an application service principal, or a [managed identity for Azure resources](#). The RBAC roles that are assigned to a security principal determine the permissions that the principal will have. To learn more about assigning Azure roles for queue access, see [Assign an Azure role for access to queue data](#)

In some cases you may need to enable fine-grained access to queue resources or to simplify permissions when you have a large number of role assignments for a storage resource. You can use Azure attribute-based access control (Azure ABAC) to configure conditions on role assignments. You can use conditions with a [custom role](#) or select built-in roles. For more information about configuring conditions for Azure storage resources with ABAC, see [Authorize access to queues using Azure role assignment conditions](#). For details about supported conditions for queue data operations, see [Actions and attributes for Azure role assignment conditions for Azure queues](#).

 **Note**

When you create an Azure Storage account, you are not automatically assigned permissions to access data via Microsoft Entra ID. You must explicitly assign yourself an Azure role for access to Queue Storage. You can assign it at the level of your subscription, resource group, storage account, or queue.

Resource scope

Before you assign an Azure RBAC role to a security principal, determine the scope of access that the security principal should have. Best practices dictate that it's always best to grant only the narrowest possible scope. Azure RBAC roles defined at a broader scope are inherited by the resources beneath them.

You can scope access to Azure queue resources at the following levels, beginning with the narrowest scope:

- **An individual queue.** At this scope, a role assignment applies to messages in the queue, and to queue properties and metadata.
- **The storage account.** At this scope, a role assignment applies to all queues and their messages.
- **The resource group.** At this scope, a role assignment applies to all of the queues in all of the storage accounts in the resource group.
- **The subscription.** At this scope, a role assignment applies to all of the queues in all of the storage accounts in all of the resource groups in the subscription.

- **A management group.** At this scope, a role assignment applies to all of the queues in all of the storage accounts in all of the resource groups in all of the subscriptions in the management group.

For more information about scope for Azure RBAC role assignments, see [Understand scope for Azure RBAC](#).

Azure built-in roles for queues

Azure RBAC provides several built-in roles for authorizing access to queue data using Microsoft Entra ID and OAuth. Some examples of roles that provide permissions to data resources in Azure Storage include:

- [Storage Queue Data Contributor](#): Use to grant read/write/delete permissions to Azure queues.
- [Storage Queue Data Reader](#): Use to grant read-only permissions to Azure queues.
- [Storage Queue Data Message Processor](#): Use to grant peek, retrieve, and delete permissions to messages in Azure Storage queues.
- [Storage Queue Data Message Sender](#): Use to grant add permissions to messages in Azure Storage queues.

To learn how to assign an Azure built-in role to a security principal, see [Assign an Azure role for access to queue data](#). To learn how to list Azure RBAC roles and their permissions, see [List Azure role definitions](#).

For more information about how built-in roles are defined for Azure Storage, see [Understand role definitions](#). For information about creating Azure custom roles, see [Azure custom roles](#).

Only roles explicitly defined for data access permit a security principal to access queue data. Built-in roles such as **Owner**, **Contributor**, and **Storage Account Contributor** permit a security principal to manage a storage account, but don't provide access to the queue data within that account via Microsoft Entra ID. However, if a role includes **Microsoft.Storage/storageAccounts/listKeys/action**, then a user to whom that role is assigned can access data in the storage account via Shared Key authorization with the account access keys. For more information, see [Choose how to authorize access to queue data in the Azure portal](#).

For detailed information about Azure built-in roles for Azure Storage for both the data services and the management service, see the **Storage** section in [Azure built-in roles for Azure RBAC](#). Additionally, for information about the different types of roles that provide permissions in Azure, see [Azure roles, Microsoft Entra roles, and classic subscription administrator roles](#).

ⓘ Important

Azure role assignments may take up to 30 minutes to propagate.

Access permissions for data operations

For details on the permissions required to call specific Queue service operations, see [Permissions for calling data operations](#).

Access data with a Microsoft Entra account

Access to queue data via the Azure portal, PowerShell, or Azure CLI can be authorized either by using the user's Microsoft Entra account or by using the account access keys (Shared Key authorization).

ⓘ Caution

Authorization with Shared Key is not recommended as it may be less secure. For optimal security, disable authorization via Shared Key for your storage account, as described in [Prevent Shared Key authorization for an Azure Storage account](#).

Use of access keys and connection strings should be limited to initial proof of concept apps or development prototypes that don't access production or sensitive data. Otherwise, the token-based authentication classes available in the Azure SDK should always be preferred when authenticating to Azure resources.

Microsoft recommends that clients use either Microsoft Entra ID or a shared access signature (SAS) to authorize access to data in Azure Storage. For more information, see [Authorize operations for data access](#).

Data access from the Azure portal

The Azure portal can use either your Microsoft Entra account or the account access keys to access queue data in an Azure storage account. Which authorization scheme the Azure portal uses depends on the Azure roles that are assigned to you.

When you attempt to access queue data, the Azure portal first checks whether you've been assigned an Azure role with `Microsoft.Storage/storageAccounts/listkeys/action`. If you've been assigned a role with this action, then the Azure portal uses the account key for accessing queue data via Shared Key authorization. If you haven't been assigned a

role with this action, then the Azure portal attempts to access data using your Microsoft Entra account.

To access queue data from the Azure portal using your Microsoft Entra account, you need permissions to access queue data, and you also need permissions to navigate through the storage account resources in the Azure portal. The built-in roles provided by Azure Storage grant access to queue resources, but they don't grant permissions to storage account resources. For this reason, access to the portal also requires the assignment of an Azure Resource Manager role such as the [Reader](#) role, scoped to the level of the storage account or higher. The **Reader** role grants the most restricted permissions, but another Azure Resource Manager role that grants access to storage account management resources is also acceptable. To learn more about how to assign permissions to users for data access in the Azure portal with a Microsoft Entra account, see [Assign an Azure role for access to queue data](#).

The Azure portal indicates which authorization scheme is in use when you navigate to a queue. For more information about data access in the portal, see [Choose how to authorize access to queue data in the Azure portal](#).

Data access from PowerShell or Azure CLI

Azure CLI and PowerShell support signing in with Microsoft Entra credentials. After you sign in, your session runs under those credentials. To learn more, see one of the following articles:

- [Choose how to authorize access to queue data with Azure CLI](#)
- [Run PowerShell commands with Microsoft Entra credentials to access queue data](#)

Next steps

- [Authorize access to data in Azure Storage](#)
- [Assign an Azure role for access to queue data](#)

Performance and scalability checklist for Queue Storage

Article • 10/26/2023

Microsoft has developed many proven practices for developing high-performance applications with Queue Storage. This checklist identifies key practices that developers can follow to optimize performance. Keep these practices in mind while you're designing your application and throughout the process.

Azure Storage has scalability and performance targets for capacity, transaction rate, and bandwidth. For more information about Azure Storage scalability targets, see [Scalability and performance targets for standard storage accounts](#) and [Scalability and performance targets for Queue Storage](#).

Checklist

This article organizes proven practices for performance into a checklist you can follow while developing your Queue Storage application.

Done	Category	Design consideration
	Scalability targets	Can you design your application to use no more than the maximum number of storage accounts?
	Scalability targets	Are you avoiding approaching capacity and transaction limits?
	Networking	Do client-side devices have sufficiently high bandwidth and low latency to achieve the performance needed?
	Networking	Do client-side devices have a high quality network link?
	Networking	Is the client application in the same region as the storage account?
	Direct client access	Are you using shared access signatures (SAS) and cross-origin resource sharing (CORS) to enable direct access to Azure Storage?
	.NET configuration	For .NET Framework applications, have you configured your client to use a sufficient number of concurrent connections?
	.NET configuration	For .NET Framework applications, have you configured .NET to use a sufficient number of threads?
	Parallelism	Have you ensured that parallelism is bounded appropriately so that you don't overload your client's capabilities or approach the scalability

Done	Category	Design consideration
		targets?
	Tools	Are you using the latest versions of Microsoft-provided client libraries and tools?
	Retries	Are you using a retry policy with an exponential backoff for throttling errors and timeouts?
	Retries	Is your application avoiding retries for non-retryable errors?
	Configuration	Have you turned off Nagle's algorithm to improve the performance of small requests?
	Message size	Are your messages compact to improve the performance of the queue?
	Bulk retrieval	Are you retrieving multiple messages in a single get operation?
	Polling frequency	Are you polling frequently enough to reduce the perceived latency of your application?
	Update message	Are you performing an update message operation to store progress in processing messages, so that you can avoid having to reprocess the entire message if an error occurs?
	Architecture	Are you using queues to make your entire application more scalable by keeping long-running workloads out of the critical path and scale them independently?

Scalability targets

If your application approaches or exceeds any of the scalability targets, it might encounter increased transaction latencies or throttling. When Azure Storage throttles your application, the service begins to return 503 (Server Busy) or 500 (Operation Timeout) error codes. Avoiding these errors by staying within the limits of the scalability targets is an important part of enhancing your application's performance.

For more information about scalability targets for Queue Storage, see [Azure Storage scalability and performance targets](#).

Maximum number of storage accounts

If you're approaching the maximum number of storage accounts permitted for a particular subscription/region combination, are you using multiple storage accounts to shard to increase ingress, egress, I/O operations per second (IOPS), or capacity? In this scenario, Microsoft recommends that you take advantage of increased limits for storage

accounts to reduce the number of storage accounts required for your workload if possible. Contact [Azure Support](#) to request increased limits for your storage account.

Capacity and transaction targets

If your application is approaching the scalability targets for a single storage account, consider adopting one of the following approaches:

- If the scalability targets for queues are insufficient for your application, then use multiple queues and distribute messages across them.
- Reconsider the workload that causes your application to approach or exceed the scalability target. Can you design it differently to use less bandwidth or capacity, or fewer transactions?
- If your application must exceed one of the scalability targets, then create multiple storage accounts and partition your application data across those multiple storage accounts. If you use this pattern, then be sure to design your application so that you can add more storage accounts in the future for load balancing. Storage accounts themselves have no cost other than your usage in terms of data stored, transactions made, or data transferred.
- If your application is approaching the bandwidth targets, consider compressing data on the client side to reduce the bandwidth required to send the data to Azure Storage. While compressing data might save bandwidth and improve network performance, it can also have negative effects on performance. Evaluate the performance impact of the additional processing requirements for data compression and decompression on the client side. Keep in mind that storing compressed data can make troubleshooting more difficult because it might be more challenging to view the data using standard tools.
- If your application is approaching the scalability targets, then make sure that you're using an exponential backoff for retries. It's best to try to avoid reaching the scalability targets by implementing the recommendations described in this article. However, using an exponential backoff for retries prevents your application from retrying rapidly, which could make throttling worse. For more information, see the [Timeout and Server Busy errors](#) section.

Networking

The physical network constraints of the application might have a significant impact on performance. The following sections describe some of limitations users might encounter.

Client network capability

Bandwidth and the quality of the network link play important roles in application performance, as described in the following sections.

Throughput

For bandwidth, the problem is often the capabilities of the client. Larger Azure instances have NICs with greater capacity, so you should consider using a larger instance or more VMs if you need higher network limits from a single machine. If you're accessing Azure Storage from an on-premises application, then the same rule applies: understand the network capabilities of the client device and the network connectivity to the Azure Storage location and either improve them as needed or design your application to work within their capabilities.

Link quality

As with any network usage, keep in mind that network conditions resulting in errors and packet loss slows effective throughput. Using Wireshark or Network Monitor might help in diagnosing this issue.

Location

In any distributed environment, placing the client near to the server delivers in the best performance. For accessing Azure Storage with the lowest latency, the best location for your client is within the same Azure region. For example, if you have an Azure web app that uses Azure Storage, then locate them both within a single region, such as West US or Southeast Asia. Co-locating resources reduces the latency and the cost, as bandwidth usage within a single region is free.

If client applications access Azure Storage but aren't hosted within Azure, such as mobile device apps or on-premises enterprise services, then locating the storage account in a region near to those clients might reduce latency. If your clients are broadly distributed (for example, some in North America, and some in Europe), then consider using one storage account per region. This approach is easier to implement if the data the application stores is specific to individual users, and doesn't require replicating data between storage accounts.

SAS and CORS

Suppose that you need to authorize code such as JavaScript that is running in a user's web browser or in a mobile phone app to access data in Azure Storage. One approach is to build a service application that acts as a proxy. The user's device authenticates with the service, which in turn authorizes access to Azure Storage resources. In this way, you can avoid exposing your storage account keys on insecure devices. However, this approach places a significant overhead on the service application, because all of the data transferred between the user's device and Azure Storage must pass through the service application.

You can avoid using a service application as a proxy for Azure Storage by using shared access signatures (SAS). Using SAS, you can enable your user's device to make requests directly to Azure Storage by using a limited access token. For example, if a user wants to upload a photo to your application, then your service application can generate a SAS and send it to the user's device. The SAS token can grant permission to write to an Azure Storage resource for a specified interval of time, after which the SAS token expires. For more information about SAS, see [Grant limited access to Azure Storage resources using shared access signatures \(SAS\)](#).

Typically, a web browser won't allow JavaScript in a page that is hosted by a website on one domain to perform certain operations, such as write operations, to another domain. Known as the same-origin policy, this policy prevents a malicious script on one page from obtaining access to data on another web page. However, the same-origin policy can be a limitation when building a solution in the cloud. Cross-origin resource sharing (CORS) is a browser feature that enables the target domain to communicate to the browser that it trusts requests originating in the source domain.

For example, suppose a web application running in Azure makes a request for a resource to an Azure Storage account. The web application is the source domain, and the storage account is the target domain. You can configure CORS for any of the Azure Storage services to communicate to the web browser that requests from the source domain are trusted by Azure Storage. For more information about CORS, see [Cross-origin resource sharing \(CORS\) support for Azure Storage](#).

Both SAS and CORS can help you avoid unnecessary load on your web application.

.NET configuration

For projects using .NET Framework, this section lists some quick configuration settings that you can use to make significant performance improvements. If you're using a language other than .NET, check to see if similar concepts apply in your chosen language.

Increase default connection limit

ⓘ Note

This section applies to projects using .NET Framework, as connection pooling is controlled by the `ServicePointManager` class. .NET Core introduced a significant change around connection pool management, where connection pooling happens at the `HttpClient` level and the pool size is not limited by default. This means that HTTP connections are automatically scaled to satisfy your workload. Using the latest version of .NET is recommended, when possible, to take advantage of performance enhancements.

For projects using .NET Framework, you can use the following code to increase the default connection limit (which is usually 2 in a client environment or 10 in a server environment) to 100. Typically, you should set the value to approximately the number of threads used by your application. Set the connection limit before opening any connections.

C#

```
ServicePointManager.DefaultConnectionLimit = 100; // (Or More)
```

To learn more about connection pool limits in .NET Framework, see [.NET Framework Connection Pool Limits and the new Azure SDK for .NET](#).

For other programming languages, see the documentation to determine how to set the connection limit.

Increase minimum number of threads

If you're using synchronous calls together with asynchronous tasks, you might want to increase the number of threads in the thread pool:

C#

```
ThreadPool.SetMinThreads(100,100); // (Determine the right number for your application)
```

For more information, see the [ThreadPool.SetMinThreads](#) method.

Unbounded parallelism

While parallelism can be great for performance, be careful about using unbounded parallelism, meaning that there's no limit enforced on the number of threads or parallel requests. Be sure to limit parallel requests to upload or download data, to access multiple partitions in the same storage account, or to access multiple items in the same partition. If parallelism is unbounded, your application can exceed the client device's capabilities or the storage account's scalability targets, resulting in longer latencies and throttling.

Client libraries and tools

For best performance, always use the latest client libraries and tools provided by Microsoft. Azure Storage client libraries are available for various languages. Azure Storage also supports PowerShell and Azure CLI. Microsoft actively develops these client libraries and tools with performance in mind, keeps them up-to-date with the latest service versions, and ensures that they handle many of the proven performance practices internally. For more information, see the [Azure Storage reference documentation](#).

Handle service errors

Azure Storage returns an error when the service can't process a request. Understanding the errors that might be returned by Azure Storage in a given scenario is helpful for optimizing performance.

Timeout and Server Busy errors

Azure Storage might throttle your application if it approaches the scalability limits. In some cases, Azure Storage might be unable to handle a request due to some transient condition. In both cases, the service might return a 503 (`Server Busy`) or 500 (`Timeout`) error. These errors can also occur if the service is rebalancing data partitions to allow for higher throughput. The client application should typically retry the operation that causes one of these errors. However, if Azure Storage is throttling your application because it's exceeding scalability targets, or even if the service was unable to serve the request for some other reason, aggressive retries might make the problem worse. Using an exponential back off retry policy is recommended, and the client libraries default to this behavior. For example, your application might retry after 2 seconds, then 4 seconds, then 10 seconds, then 30 seconds, and then give up completely. In this way, your application significantly reduces its load on the service, rather than exacerbating behavior that could lead to throttling.

Connectivity errors can be retried immediately, because they aren't the result of throttling and are expected to be transient.

Non-retryable errors

The client libraries handle retries with an awareness of which errors can be retried and which can't. However, if you're calling the Azure Storage REST API directly, there are some errors that you shouldn't retry. For example, a 400 (`Bad Request`) error indicates that the client application sent a request that couldn't be processed because it wasn't in the expected form. Resending this request results the same response every time, so there's no point in retrying it. If you're calling the Azure Storage REST API directly, be aware of potential errors and whether they should be retried.

For more information on Azure Storage error codes, see [Status and error codes](#).

Disable Nagle's algorithm

Nagle's algorithm is widely implemented across TCP/IP networks as a means to improve network performance. However, it isn't optimal in all circumstances (such as highly interactive environments). Nagle's algorithm has a negative impact on the performance of requests to Azure Table Storage, and you should disable it if possible.

Message size

Queue performance and scalability decrease as message size increases. Put only the information the receiver needs in a message.

Batch retrieval

You can retrieve up to 32 messages from a queue in a single operation. Batch retrieval can reduce the number of round trips from the client application, which is especially useful for environments, such as mobile devices, with high latency.

Queue polling interval

Most applications poll for messages from a queue, which can be one of the largest sources of transactions for that application. Select your polling interval wisely: polling too frequently could cause your application to approach the scalability targets for the queue. However, at 200,000 transactions for \$0.01 (at the time of writing), a single

processor polling once every second for a month would cost less than 15 cents so cost isn't typically a factor that affects your choice of polling interval.

For up-to-date cost information, see [Azure Storage pricing](#).

Perform an update message operation

You can perform an update message operation to increase the invisibility timeout or to update the state information of a message. This approach can be a more efficient than having a workflow that passes a job from one queue to the next, as each step of the job is completed. Your application can save the job state to the message and then continue working, instead of requeueing the message for the next step of the job every time a step completes. Keep in mind that each update message operation counts towards the scalability target.

Application architecture

Use queues to make your application architecture scalable. The following lists some ways you can use queues to make your application more scalable:

- You can use queues to create backlogs of work for processing and smooth out workloads in your application. For example, you could queue up requests from users to perform processor intensive work such as resizing uploaded images.
- You can use queues to decouple parts of your application so that you can scale them independently. For example, a web front end could place survey results from users into a queue for later analysis and storage. You could add more worker role instances to process the queue data as required.

Next steps

- [Scalability and performance targets for Queue Storage](#)
- [Scalability and performance targets for standard storage accounts](#)
- [Status and error codes](#)

Authorize access to tables using Microsoft Entra ID

Article • 10/12/2023

Azure Storage supports using Microsoft Entra ID to authorize requests to table data. With Microsoft Entra ID, you can use Azure role-based access control (Azure RBAC) to grant permissions to a security principal, which may be a user, group, or application service principal. The security principal is authenticated by Microsoft Entra ID to return an OAuth 2.0 token. The token can then be used to authorize a request against the Table service.

Authorizing requests against Azure Storage with Microsoft Entra ID provides superior security and ease of use over Shared Key authorization. Microsoft recommends using Microsoft Entra authorization with your table applications when possible to assure access with minimum required privileges.

Authorization with Microsoft Entra ID is available for all general-purpose in all public regions and national clouds. Only storage accounts created with the Azure Resource Manager deployment model support Microsoft Entra authorization.

Overview of Microsoft Entra ID for tables

When a security principal (a user, group, or application) attempts to access a table resource, the request must be authorized. With Microsoft Entra ID, access to a resource is a two-step process. First, the security principal's identity is authenticated and an OAuth 2.0 token is returned. Next, the token is passed as part of a request to the Table service and used by the service to authorize access to the specified resource.

The authentication step requires that an application request an OAuth 2.0 access token at runtime. If an application is running from within an Azure entity such as an Azure VM, a virtual machine scale set, or an Azure Functions app, it can use a [managed identity](#) to access tables.

The authorization step requires that one or more Azure roles be assigned to the security principal. Azure Storage provides Azure roles that encompass common sets of permissions for table data. The roles that are assigned to a security principal determine the permissions that that principal will have. To learn more about assigning Azure roles for table access, see [Assign an Azure role for access to table data](#).

The following table points to additional information for authorizing access to data in various scenarios:

Language	.NET	Java	JavaScript	Python	Go
Overview of auth with Microsoft Entra ID	How to authenticate .NET applications with Azure services	Azure authentication with Java and Azure Identity	Authenticate JavaScript apps to Azure using the Azure SDK	Authenticate Python apps to Azure using the Azure SDK	
Auth using developer service principals	Authenticate .NET apps to Azure services during local development using service principals	Azure authentication with service principal	Auth JS apps to Azure services with service principal	Authenticate Python apps to Azure services during local development using service principals	Azure SDK for Go authentication with a service principal
Auth using developer or user accounts	Authenticate .NET apps to Azure services during local development using developer accounts	Azure authentication with user credentials	Auth JS apps to Azure services with dev accounts	Authenticate Python apps to Azure services during local development using developer accounts	Azure authentication with the Azure SDK for Go
Auth from Azure-hosted apps	Authenticating Azure-hosted apps to Azure resources with the Azure SDK for .NET	Authenticate Azure-hosted Java applications	Authenticating Azure-hosted JavaScript apps to Azure resources with the Azure SDK for JavaScript	Authenticating Azure-hosted apps to Azure resources with the Azure SDK for Python	Authentication with the Azure SDK for Go using a managed identity
Auth from on-premises apps	Authenticate to Azure resources from .NET apps hosted on-premises		Authenticate on-premises JavaScript apps to Azure resources	Authenticate to Azure resources from Python apps hosted on-premises	
Identity client library overview	Azure Identity client library for .NET	Azure Identity client library for Java	Azure Identity client library for JavaScript	Azure Identity client library for Python	Azure Identity client library for Go ↗

Assign Azure roles for access rights

Microsoft Entra authorizes access rights to secured resources through [Azure role-based access control \(Azure RBAC\)](#). Azure Storage defines a set of Azure built-in roles that encompass common sets of permissions used to access table data. You can also define custom roles for access to table data.

When an Azure role is assigned to a Microsoft Entra security principal, Azure grants access to those resources for that security principal. A Microsoft Entra security principal may be a user, a group, an application service principal, or a [managed identity for Azure resources](#).

Resource scope

Before you assign an Azure RBAC role to a security principal, determine the scope of access that the security principal should have. Best practices dictate that it's always best to grant only the narrowest possible scope. Azure RBAC roles defined at a broader scope are inherited by the resources beneath them.

You can scope access to Azure table resources at the following levels, beginning with the narrowest scope:

- **An individual table.** At this scope, a role assignment applies to the specified table.
- **The storage account.** At this scope, a role assignment applies to all tables in the account.
- **The resource group.** At this scope, a role assignment applies to all of the tables in all of the storage accounts in the resource group.
- **The subscription.** At this scope, a role assignment applies to all of the tables in all of the storage accounts in all of the resource groups in the subscription.
- **A management group.** At this scope, a role assignment applies to all of the tables in all of the storage accounts in all of the resource groups in all of the subscriptions in the management group.

For more information about scope for Azure RBAC role assignments, see [Understand scope for Azure RBAC](#).

Azure built-in roles for tables

Azure RBAC provides built-in roles for authorizing access to table data using Microsoft Entra ID and OAuth. Built-in roles that provide permissions to tables in Azure Storage include:

- **Storage Table Data Contributor:** Use to grant read/write/delete permissions to Table storage resources.
- **Storage Table Data Reader:** Use to grant read-only permissions to Table storage resources.

To learn how to assign an Azure built-in role to a security principal, see [Assign an Azure role for access to table data](#). To learn how to list Azure RBAC roles and their permissions, see [List Azure role definitions](#).

For more information about how built-in roles are defined for Azure Storage, see [Understand role definitions](#). For information about creating Azure custom roles, see [Azure custom roles](#).

Only roles explicitly defined for data access permit a security principal to access table data. Built-in roles such as **Owner**, **Contributor**, and **Storage Account Contributor** permit a security principal to manage a storage account, but do not provide access to the table data within that account via Microsoft Entra ID. However, if a role includes **Microsoft.Storage/storageAccounts/listKeys/action**, then a user to whom that role is assigned can access data in the storage account via Shared Key authorization with the account access keys.

For detailed information about Azure built-in roles for Azure Storage for both the data services and the management service, see the **Storage** section in [Azure built-in roles for Azure RBAC](#). Additionally, for information about the different types of roles that provide permissions in Azure, see [Azure roles, Microsoft Entra roles, and classic subscription administrator roles](#).

 **Important**

Azure role assignments may take up to 30 minutes to propagate.

Access permissions for data operations

For details on the permissions required to call specific Table service operations, see [Permissions for calling data operations](#).

Next steps

- [Authorize access to data in Azure Storage](#)
- [Assign an Azure role for access to table data](#)

Performance and scalability checklist for Table storage

Article • 11/29/2022

Microsoft has developed a number of proven practices for developing high-performance applications with Table storage. This checklist identifies key practices that developers can follow to optimize performance. Keep these practices in mind while you are designing your application and throughout the process.

Azure Storage has scalability and performance targets for capacity, transaction rate, and bandwidth. For more information about Azure Storage scalability targets, see [Scalability and performance targets for standard storage accounts](#) and [Scalability and performance targets for Table storage](#).

Checklist

This article organizes proven practices for performance into a checklist you can follow while developing your Table storage application.

Done	Category	Design consideration
	Scalability targets	Can you design your application to use no more than the maximum number of storage accounts?
	Scalability targets	Are you avoiding approaching capacity and transaction limits?
	Scalability targets	Are you approaching the scalability targets for entities per second?
	Networking	Do client-side devices have sufficiently high bandwidth and low latency to achieve the performance needed?
	Networking	Do client-side devices have a high quality network link?
	Networking	Is the client application in the same region as the storage account?
	Direct Client Access	Are you using shared access signatures (SAS) and cross-origin resource sharing (CORS) to enable direct access to Azure Storage?
	Batching	Is your application batching updates by using entity group transactions?
	.NET configuration	Are you using .NET Core 2.1 or later for optimum performance?

Done	Category	Design consideration
	.NET configuration	Have you configured your client to use a sufficient number of concurrent connections?
	.NET configuration	For .NET applications, have you configured .NET to use a sufficient number of threads?
	Parallelism	Have you ensured that parallelism is bounded appropriately so that you don't overload your client's capabilities or approach the scalability targets?
	Tools	Are you using the latest versions of Microsoft-provided client libraries and tools?
	Retries	Are you using a retry policy with an exponential backoff for throttling errors and timeouts?
	Retries	Is your application avoiding retries for non-retryable errors?
	Configuration	Are you using JSON for your table requests?
	Configuration	Have you turned off the Nagle algorithm to improve the performance of small requests?
	Tables and partitions	Have you properly partitioned your data?
	Hot partitions	Are you avoiding append-only and prepend-only patterns?
	Hot partitions	Are your inserts/updates spread across many partitions?
	Query scope	Have you designed your schema to allow for point queries to be used in most cases, and table queries to be used sparingly?
	Query density	Do your queries typically only scan and return rows that your application will use?
	Limiting returned data	Are you using filtering to avoid returning entities that are not needed?
	Limiting returned data	Are you using projection to avoid returning properties that are not needed?
	Denormalization	Have you denormalized your data such that you avoid inefficient queries or multiple read requests when trying to get data?
	Insert, update, and delete	Are you batching requests that need to be transactional or can be done at the same time to reduce round-trips?
	Insert, update, and delete	Are you avoiding retrieving an entity just to determine whether to call insert or update?

Done	Category	Design consideration
	Insert, update, and delete	Have you considered storing series of data that will frequently be retrieved together in a single entity as properties instead of multiple entities?
	Insert, update, and delete	For entities that will always be retrieved together and can be written in batches (for example, time series data), have you considered using blobs instead of tables?

Scalability targets

If your application approaches or exceeds any of the scalability targets, it may encounter increased transaction latencies or throttling. When Azure Storage throttles your application, the service begins to return 503 (Server busy) or 500 (Operation timeout) error codes. Avoiding these errors by staying within the limits of the scalability targets is an important part of enhancing your application's performance.

For more information about scalability targets for the Table service, see [Scalability and performance targets for Table storage](#).

Maximum number of storage accounts

If you're approaching the maximum number of storage accounts permitted for a particular subscription/region combination, are you using multiple storage accounts to shard to increase ingress, egress, I/O operations per second (IOPS), or capacity? In this scenario, Microsoft recommends that you take advantage of increased limits for storage accounts to reduce the number of storage accounts required for your workload if possible. Contact [Azure Support](#) to request increased limits for your storage account.

Capacity and transaction targets

If your application is approaching the scalability targets for a single storage account, consider adopting one of the following approaches:

- Reconsider the workload that causes your application to approach or exceed the scalability target. Can you design it differently to use less bandwidth or capacity, or fewer transactions?
- If your application must exceed one of the scalability targets, then create multiple storage accounts and partition your application data across those multiple storage accounts. If you use this pattern, then be sure to design your application so that you can add more storage accounts in the future for load balancing. Storage

accounts themselves have no cost other than your usage in terms of data stored, transactions made, or data transferred.

- If your application is approaching the bandwidth targets, consider compressing data on the client side to reduce the bandwidth required to send the data to Azure Storage. While compressing data may save bandwidth and improve network performance, it can also have negative effects on performance. Evaluate the performance impact of the additional processing requirements for data compression and decompression on the client side. Keep in mind that storing compressed data can make troubleshooting more difficult because it may be more challenging to view the data using standard tools.
- If your application is approaching the scalability targets, then make sure that you are using an exponential backoff for retries. It's best to try to avoid reaching the scalability targets by implementing the recommendations described in this article. However, using an exponential backoff for retries will prevent your application from retrying rapidly, which could make throttling worse. For more information, see the section titled [Timeout and Server Busy errors](#).

Targets for data operations

Azure Storage load balances as the traffic to your storage account increases, but if the traffic exhibits sudden bursts, you may not be able to get this volume of throughput immediately. Expect to see throttling and/or timeouts during the burst as Azure Storage automatically load balances your table. Ramping up slowly generally provides better results, as the system has time to load balance appropriately.

Entities per second (storage account)

The scalability limit for accessing tables is up to 20,000 entities (1 KB each) per second for an account. In general, each entity that is inserted, updated, deleted, or scanned counts toward this target. So a batch insert that contains 100 entities would count as 100 entities. A query that scans 1000 entities and returns 5 would count as 1000 entities.

Entities per second (partition)

Within a single partition, the scalability target for accessing tables is 2,000 entities (1 KB each) per second, using the same counting as described in the previous section.

Networking

The physical network constraints of the application may have a significant impact on performance. The following sections describe some of limitations users may encounter.

Client network capability

Bandwidth and the quality of the network link play important roles in application performance, as described in the following sections.

Throughput

For bandwidth, the problem is often the capabilities of the client. Larger Azure instances have NICs with greater capacity, so you should consider using a larger instance or more VMs if you need higher network limits from a single machine. If you are accessing Azure Storage from an on premises application, then the same rule applies: understand the network capabilities of the client device and the network connectivity to the Azure Storage location and either improve them as needed or design your application to work within their capabilities.

Link quality

As with any network usage, keep in mind that network conditions resulting in errors and packet loss will slow effective throughput. Using WireShark or NetMon may help in diagnosing this issue.

Location

In any distributed environment, placing the client near to the server delivers in the best performance. For accessing Azure Storage with the lowest latency, the best location for your client is within the same Azure region. For example, if you have an Azure web app that uses Azure Storage, then locate them both within a single region, such as US West or Asia Southeast. Co-locating resources reduces the latency and the cost, as bandwidth usage within a single region is free.

If client applications will access Azure Storage but are not hosted within Azure, such as mobile device apps or on premises enterprise services, then locating the storage account in a region near to those clients may reduce latency. If your clients are broadly distributed (for example, some in North America, and some in Europe), then consider using one storage account per region. This approach is easier to implement if the data the application stores is specific to individual users, and does not require replicating data between storage accounts.

SAS and CORS

Suppose that you need to authorize code such as JavaScript that is running in a user's web browser or in a mobile phone app to access data in Azure Storage. One approach is to build a service application that acts as a proxy. The user's device authenticates with the service, which in turn authorizes access to Azure Storage resources. In this way, you can avoid exposing your storage account keys on insecure devices. However, this approach places a significant overhead on the service application, because all of the data transferred between the user's device and Azure Storage must pass through the service application.

You can avoid using a service application as a proxy for Azure Storage by using shared access signatures (SAS). Using SAS, you can enable your user's device to make requests directly to Azure Storage by using a limited access token. For example, if a user wants to upload a photo to your application, then your service application can generate a SAS and send it to the user's device. The SAS token can grant permission to write to an Azure Storage resource for a specified interval of time, after which the SAS token expires. For more information about SAS, see [Grant limited access to Azure Storage resources using shared access signatures \(SAS\)](#).

Typically, a web browser will not allow JavaScript in a page that is hosted by a website on one domain to perform certain operations, such as write operations, to another domain. Known as the same-origin policy, this policy prevents a malicious script on one page from obtaining access to data on another web page. However, the same-origin policy can be a limitation when building a solution in the cloud. Cross-origin resource sharing (CORS) is a browser feature that enables the target domain to communicate to the browser that it trusts requests originating in the source domain.

For example, suppose a web application running in Azure makes a request for a resource to an Azure Storage account. The web application is the source domain, and the storage account is the target domain. You can configure CORS for any of the Azure Storage services to communicate to the web browser that requests from the source domain are trusted by Azure Storage. For more information about CORS, see [Cross-origin resource sharing \(CORS\) support for Azure Storage](#).

Both SAS and CORS can help you avoid unnecessary load on your web application.

Batch transactions

The Table service supports batch transactions on entities that are in the same table and belong to the same partition group. For more information, see [Performing entity group transactions](#).

.NET configuration

If using the .NET Framework, this section lists several quick configuration settings that you can use to make significant performance improvements. If using other languages, check to see if similar concepts apply in your chosen language.

Use .NET Core

Develop your Azure Storage applications with .NET Core 2.1 or later to take advantage of performance enhancements. Using .NET Core 3.x is recommended when possible.

For more information on performance improvements in .NET Core, see the following blog posts:

- [Performance Improvements in .NET Core 3.0](#)
- [Performance Improvements in .NET Core 2.1](#)

Increase default connection limit

In .NET, the following code increases the default connection limit (which is usually 2 in a client environment or 10 in a server environment) to 100. Typically, you should set the value to approximately the number of threads used by your application.

C#

```
ServicePointManager.DefaultConnectionLimit = 100; //Or More
```

Set the connection limit before opening any connections.

For other programming languages, see that language's documentation to determine how to set the connection limit.

For more information, see the blog post [Web Services: Concurrent Connections](#).

Increase minimum number of threads

If you are using synchronous calls together with asynchronous tasks, you may want to increase the number of threads in the thread pool:

C#

```
ThreadPool.SetMinThreads(100,100); //Determine the right number for your application
```

For more information, see the [ThreadPool.SetMinThreads](#) method.

Unbounded parallelism

While parallelism can be great for performance, be careful about using unbounded parallelism, meaning that there is no limit enforced on the number of threads or parallel requests. Be sure to limit parallel requests to upload or download data, to access multiple partitions in the same storage account, or to access multiple items in the same partition. If parallelism is unbounded, your application can exceed the client device's capabilities or the storage account's scalability targets, resulting in longer latencies and throttling.

Client libraries and tools

For best performance, always use the latest client libraries and tools provided by Microsoft. Azure Storage client libraries are available for a variety of languages. Azure Storage also supports PowerShell and Azure CLI. Microsoft actively develops these client libraries and tools with performance in mind, keeps them up-to-date with the latest service versions, and ensures that they handle many of the proven performance practices internally.

Handle service errors

Azure Storage returns an error when the service cannot process a request. Understanding the errors that may be returned by Azure Storage in a given scenario is helpful for optimizing performance.

Timeout and Server Busy errors

Azure Storage may throttle your application if it approaches the scalability limits. In some cases, Azure Storage may be unable to handle a request due to some transient condition. In both cases, the service may return a 503 (Server Busy) or 500 (Timeout) error. These errors can also occur if the service is rebalancing data partitions to allow for higher throughput. The client application should typically retry the operation that causes one of these errors. However, if Azure Storage is throttling your application because it is exceeding scalability targets, or even if the service was unable to serve the request for some other reason, aggressive retries may make the problem worse. Using an exponential back off retry policy is recommended, and the client libraries default to this behavior. For example, your application may retry after 2 seconds, then 4 seconds, then

10 seconds, then 30 seconds, and then give up completely. In this way, your application significantly reduces its load on the service, rather than exacerbating behavior that could lead to throttling.

Connectivity errors can be retried immediately, because they are not the result of throttling and are expected to be transient.

Non-retryable errors

The client libraries handle retries with an awareness of which errors can be retried and which cannot. However, if you are calling the Azure Storage REST API directly, there are some errors that you should not retry. For example, a 400 (Bad Request) error indicates that the client application sent a request that could not be processed because it was not in the expected form. Resending this request results the same response every time, so there is no point in retrying it. If you are calling the Azure Storage REST API directly, be aware of potential errors and whether they should be retried.

For more information on Azure Storage error codes, see [Status and error codes](#).

Configuration

This section lists several quick configuration settings that you can use to make significant performance improvements in the Table service:

Use JSON

Beginning with storage service version 2013-08-15, the Table service supports using JSON instead of the XML-based AtomPub format for transferring table data. Using JSON can reduce payload sizes by as much as 75% and can significantly improve the performance of your application.

For more information, see the post [Microsoft Azure Tables: Introducing JSON and Payload Format for Table Service Operations](#).

Disable Nagle

Nagle's algorithm is widely implemented across TCP/IP networks as a means to improve network performance. However, it is not optimal in all circumstances (such as highly interactive environments). Nagle's algorithm has a negative impact on the performance of requests to the Azure Table service, and you should disable it if possible.

Schema

How you represent and query your data is the biggest single factor that affects the performance of the Table service. While every application is different, this section outlines some general proven practices that relate to:

- Table design
- Efficient queries
- Efficient data updates

Tables and partitions

Tables are divided into partitions. Every entity stored in a partition shares the same partition key and has a unique row key to identify it within that partition. Partitions provide benefits but also introduce scalability limits.

- Benefits: You can update entities in the same partition in a single, atomic, batch transaction that contains up to 100 separate storage operations (limit of 4 MB total size). Assuming the same number of entities to be retrieved, you can also query data within a single partition more efficiently than data that spans partitions (though read on for further recommendations on querying table data).
- Scalability limit: Access to entities stored in a single partition cannot be load-balanced because partitions support atomic batch transactions. For this reason, the scalability target for an individual table partition is lower than for the Table service as a whole.

Because of these characteristics of tables and partitions, you should adopt the following design principles:

- Locate data that your client application frequently updates or queries in the same logical unit of work in the same partition. For example, locate data in the same partition if your application is aggregating writes or you are performing atomic batch operations. Also, data in a single partition can be more efficiently queried in a single query than data across partitions.
- Locate data that your client application does not insert, update, or query in the same logical unit of work (that is, in a single query or batch update) in separate partitions. Keep in mind that there is no limit to the number of partition keys in a single table, so having millions of partition keys is not a problem and will not impact performance. For example, if your application is a popular website with user login, using the User ID as the partition key could be a good choice.

Hot partitions

A hot partition is one that is receiving a disproportionate percentage of the traffic to an account, and cannot be load balanced because it is a single partition. In general, hot partitions are created one of two ways:

Append Only and Prepend Only patterns

The "Append Only" pattern is one where all (or nearly all) of the traffic to a given partition key increases and decreases according to the current time. For example, suppose that your application uses the current date as a partition key for log data. This design results in all of the inserts going to the last partition in your table, and the system cannot load balance properly. If the volume of traffic to that partition exceeds the partition-level scalability target, then it will result in throttling. It's better to ensure that traffic is sent to multiple partitions, to enable load balance the requests across your table.

High-traffic data

If your partitioning scheme results in a single partition that just has data that is far more used than other partitions, you may also see throttling as that partition approaches the scalability target for a single partition. It's better to make sure that your partition scheme results in no single partition approaching the scalability targets.

Querying

This section describes proven practices for querying the Table service.

Query scope

There are several ways to specify the range of entities to query. The following list describes each option for query scope.

- **Point queries**:- A point query retrieves exactly one entity by specifying both the partition key and row key of the entity to retrieve. These queries are efficient, and you should use them wherever possible.
- **Partition queries**: A partition query is a query that retrieves a set of data that shares a common partition key. Typically, the query specifies a range of row key values or a range of values for some entity property in addition to a partition key. These queries are less efficient than point queries, and should be used sparingly.

- **Table queries:** A table query is a query that retrieves a set of entities that does not share a common partition key. These queries are not efficient and you should avoid them if possible.

In general, avoid scans (queries larger than a single entity), but if you must scan, try to organize your data so that your scans retrieve the data you need without scanning or returning significant amounts of entities you don't need.

Query density

Another key factor in query efficiency is the number of entities returned as compared to the number of entities scanned to find the returned set. If your application performs a table query with a filter for a property value that only 1% of the data shares, the query will scan 100 entities for every one entity it returns. The table scalability targets discussed previously all relate to the number of entities scanned, and not the number of entities returned: a low query density can easily cause the Table service to throttle your application because it must scan so many entities to retrieve the entity you are looking for. For more information on how to avoid throttling, see the section titled [Denormalization](#).

Limiting the amount of data returned

When you know that a query will return entities that you don't need in the client application, consider using a filter to reduce the size of the returned set. While the entities not returned to the client still count toward the scalability limits, your application performance will improve because of the reduced network payload size and the reduced number of entities that your client application must process. Keep in mind that the scalability targets relate to the number of entities scanned, so a query that filters out many entities may still result in throttling, even if few entities are returned. For more information on making queries efficient, see the section titled [Query density](#).

If your client application needs only a limited set of properties from the entities in your table, you can use projection to limit the size of the returned data set. As with filtering, projection helps to reduce network load and client processing.

Denormalization

Unlike working with relational databases, the proven practices for efficiently querying table data lead to denormalizing your data. That is, duplicating the same data in multiple entities (one for each key you may use to find the data) to minimize the number of entities that a query must scan to find the data the client needs, rather than

having to scan large numbers of entities to find the data your application needs. For example, in an e-commerce website, you may want to find an order both by the customer ID (give me this customer's orders) and by the date (give me orders on a date). In Table Storage, it is best to store the entity (or a reference to it) twice – once with Table Name, PK, and RK to facilitate finding by customer ID, once to facilitate finding it by the date.

Insert, update, and delete

This section describes proven practices for modifying entities stored in the Table service.

Batching

Batch transactions are known as entity group transactions in Azure Storage. All operations within an entity group transaction must be on a single partition in a single table. Where possible, use entity group transactions to perform inserts, updates, and deletes in batches. Using entity group transactions reduces the number of round trips from your client application to the server, reduces the number of billable transactions (an entity group transaction counts as a single transaction for billing purposes and can contain up to 100 storage operations), and enables atomic updates (all operations succeed or all fail within an entity group transaction). Environments with high latencies such as mobile devices will benefit greatly from using entity group transactions.

Upsert

Use table **Upsert** operations wherever possible. There are two types of **Upsert**, both of which can be more efficient than a traditional **Insert** and **Update** operations:

- **InsertOrMerge**: Use this operation when you want to upload a subset of the entity's properties, but aren't sure whether the entity already exists. If the entity exists, this call updates the properties included in the **Upsert** operation, and leaves all existing properties as they are, if the entity does not exist, it inserts the new entity. This is similar to using projection in a query, in that you only need to upload the properties that are changing.
- **InsertOrReplace**: Use this operation when you want to upload an entirely new entity, but you aren't sure whether it already exists. Use this operation when you know that the newly uploaded entity is entirely correct because it completely overwrites the old entity. For example, you want to update the entity that stores a user's current location regardless of whether or not the application has previously stored location data for the user; the new location entity is complete, and you do not need any information from any previous entity.

Storing data series in a single entity

Sometimes, an application stores a series of data that it frequently needs to retrieve all at once: for example, an application might track CPU usage over time in order to plot a rolling chart of the data from the last 24 hours. One approach is to have one table entity per hour, with each entity representing a specific hour and storing the CPU usage for that hour. To plot this data, the application needs to retrieve the entities holding the data from the 24 most recent hours.

Alternatively, your application could store the CPU usage for each hour as a separate property of a single entity: to update each hour, your application can use a single **InsertOrMerge Upsert** call to update the value for the most recent hour. To plot the data, the application only needs to retrieve a single entity instead of 24, making for an efficient query. For more information on query efficiency, see the section titled [Query scope](#)).

Storing structured data in blobs

If you are performing batch inserts and then retrieving ranges of entities together, consider using blobs instead of tables. A good example is a log file. You can batch several minutes of logs and insert them, and then retrieve several minutes of logs at a time. In this case, performance is better if you use blobs instead of tables, since you can significantly reduce the number of objects written to or read, and also possibly the number of requests that need made.

Next steps

- [Scalability and performance targets for Table storage](#)
- [Scalability and performance targets for standard storage accounts](#)
- [Status and error codes](#)

Design scalable and performant tables

Article • 01/11/2023

Tip

The content in this article applies to the original Azure Table storage. However, the same concepts apply to the newer Azure Cosmos DB for Table, which offers higher performance and availability, global distribution, and automatic secondary indexes. It is also available in a consumption-based **serverless** mode. There are some **feature differences** between Table API in Azure Cosmos DB and Azure Table storage. For more information, see [Azure Cosmos DB for Table](#). For ease of development, we now provide a unified [Azure Tables SDK](#) that can be used to target both Azure Table storage and Azure Cosmos DB for Table.

To design scalable and performant tables, you must consider factors such as performance, scalability, and cost. If you have previously designed schemas for relational databases, these considerations are familiar, but while there are some similarities between the Azure Table service storage model and relational models, there are also important differences. These differences typically lead to different designs that may look counter-intuitive or wrong to someone familiar with relational databases, yet make sense if you are designing for a NoSQL key/value store such as the Azure Table service. Many of your design differences reflect the fact that the Table service is designed to support cloud-scale applications that can contain billions of entities (or rows in relational database terminology) of data or for datasets that must support high transaction volumes. Therefore, you must think differently about how you store your data and understand how the Table service works. A well-designed NoSQL data store can enable your solution to scale much further and at a lower cost than a solution that uses a relational database. This guide helps you with these topics.

About the Azure Table service

This section highlights some of the key features of the Table service that are especially relevant to designing for performance and scalability. If you're new to Azure Storage and the Table service, first read [Get started with Azure Table Storage using .NET](#) before reading the remainder of this article. Although the focus of this guide is on the Table service, it includes discussion of the Azure Queue and Blob services, and how you might use them with the Table service.

What is the Table service? As you might expect from the name, the Table service uses a tabular format to store data. In the standard terminology, each row of the table represents an entity, and the columns store the various properties of that entity. Every entity has a pair of keys to uniquely identify it, and a timestamp column that the Table service uses to track when the entity was last updated. The timestamp is applied automatically, and you cannot manually overwrite the timestamp with an arbitrary value. The Table service uses this last-modified timestamp (LMT) to manage optimistic concurrency.

 **Note**

The Table service REST API operations also return an **ETag** value that it derives from the LMT. This document uses the terms ETag and LMT interchangeably because they refer to the same underlying data.

The following example shows a simple table design to store employee and department entities. Many of the examples shown later in this guide are based on this simple design.

PartitionKey	RowKey	Timestamp									
Marketing	00001	2014-08-22T00:50:32Z	<table border="1"><thead><tr><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td>Don</td><td>Hall</td><td>34</td><td>dohn@contoso.com</td></tr></tbody></table>	FirstName	LastName	Age	Email	Don	Hall	34	dohn@contoso.com
FirstName	LastName	Age	Email								
Don	Hall	34	dohn@contoso.com								
Marketing	00002	2014-08-22T00:50:34Z	<table border="1"><thead><tr><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td>Jun</td><td>Cao</td><td>47</td><td>junc@contoso.com</td></tr></tbody></table>	FirstName	LastName	Age	Email	Jun	Cao	47	junc@contoso.com
FirstName	LastName	Age	Email								
Jun	Cao	47	junc@contoso.com								
Marketing	Department	2014-08-22T00:50:30Z	<table border="1"><thead><tr><th>DepartmentName</th><th>EmployeeCount</th></tr></thead><tbody><tr><td>Marketing</td><td>153</td></tr></tbody></table>	DepartmentName	EmployeeCount	Marketing	153				
DepartmentName	EmployeeCount										
Marketing	153										
Sales	00010	2014-08-22T00:50:44Z	<table border="1"><thead><tr><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td>Ken</td><td>Kwok</td><td>23</td><td>kenk@contoso.com</td></tr></tbody></table>	FirstName	LastName	Age	Email	Ken	Kwok	23	kenk@contoso.com
FirstName	LastName	Age	Email								
Ken	Kwok	23	kenk@contoso.com								

So far, this data appears similar to a table in a relational database with the key differences being the mandatory columns, and the ability to store multiple entity types in the same table. Also, each of the user-defined properties such as **FirstName** or **Age**

has a data type, such as integer or string, just like a column in a relational database. Although unlike in a relational database, the schema-less nature of the Table service means that a property need not have the same data type on each entity. To store complex data types in a single property, you must use a serialized format such as JSON or XML. For more information about the table service such as supported data types, supported date ranges, naming rules, and size constraints, see [Understanding the Table Service Data Model](#).

Your choice of **PartitionKey** and **RowKey** is fundamental to good table design. Every entity stored in a table must have a unique combination of **PartitionKey** and **RowKey**. As with keys in a relational database table, the **PartitionKey** and **RowKey** values are indexed to create a clustered index to enable fast look-ups. However, the Table service does not create any secondary indexes, so **PartitionKey** and **RowKey** are the only indexed properties. Some of the patterns described in [Table design patterns](#) illustrate how you can work around this apparent limitation.

A table comprises one or more partitions, and many of the design decisions you make will be around choosing a suitable **PartitionKey** and **RowKey** to optimize your solution. A solution may consist of a single table that contains all your entities organized into partitions, but typically a solution has multiple tables. Tables help you to logically organize your entities, help you manage access to the data using access control lists, and you can drop an entire table using a single storage operation.

Table partitions

The account name, table name, and **PartitionKey** together identify the partition within the storage service where the table service stores the entity. As well as being part of the addressing scheme for entities, partitions define a scope for transactions (see [Entity Group Transactions](#) below), and form the basis of how the table service scales. For more information on partitions, see [Performance and scalability checklist for Table storage](#).

In the Table service, an individual node services one or more complete partitions, and the service scales by dynamically load-balancing partitions across nodes. If a node is under load, the table service can *split* the range of partitions serviced by that node onto different nodes; when traffic subsides, the service can *merge* the partition ranges from quiet nodes back onto a single node.

For more information about the internal details of the Table service, and in particular how the service manages partitions, see the paper [Microsoft Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency](#).

Entity Group Transactions

In the Table service, Entity Group Transactions (EGTs) are the only built-in mechanism for performing atomic updates across multiple entities. EGTs are sometimes also referred to as *batch transactions*. EGTs can only operate on entities stored in the same partition (that is, share the same partition key in a given table). So anytime you require atomic transactional behavior across multiple entities, you must ensure that those entities are in the same partition. This is often a reason for keeping multiple entity types in the same table (and partition) and not using multiple tables for different entity types. A single EGT can operate on at most 100 entities. If you submit multiple concurrent EGTs for processing, it is important to ensure those EGTs do not operate on entities that are common across EGTs; otherwise, processing can be delayed.

EGTs also introduce a potential trade-off for you to evaluate in your design. That is, using more partitions increases the scalability of your application, because Azure has more opportunities for load balancing requests across nodes. But using more partitions might limit the ability of your application to perform atomic transactions and maintain strong consistency for your data. Furthermore, there are specific scalability targets at the level of a partition that might limit the throughput of transactions you can expect for a single node. For more information about scalability targets for Azure standard storage accounts, see [Scalability targets for standard storage accounts](#). For more information about scalability targets for the Table service, see [Scalability and performance targets for Table storage](#).

Capacity considerations

The following table describes capacity, scalability, and performance targets for Table storage.

Resource	Target
Number of tables in an Azure storage account	Limited only by the capacity of the storage account
Number of partitions in a table	Limited only by the capacity of the storage account
Number of entities in a partition	Limited only by the capacity of the storage account
Maximum size of a single table	500 TiB

Resource	Target
Maximum size of a single entity, including all property values	1 MiB
Maximum number of properties in a table entity	255 (including the three system properties, PartitionKey , RowKey , and Timestamp)
Maximum total size of an individual property in an entity	Varies by property type. For more information, see Property Types in Understanding the Table Service Data Model .
Size of the PartitionKey	A string up to 1 KiB in size
Size of the RowKey	A string up to 1 KiB in size
Size of an entity group transaction	A transaction can include at most 100 entities and the payload must be less than 4 MiB in size. An entity group transaction can include an update to an entity only once.
Maximum number of stored access policies per table	5
Maximum request rate per storage account	20,000 transactions per second, which assumes a 1-KiB entity size
Target throughput for a single table partition (1 KiB-entities)	Up to 2,000 entities per second

Cost considerations

Table storage is relatively inexpensive, but you should include cost estimates for both capacity usage and the quantity of transactions as part of your evaluation of any Table service solution. However, in many scenarios, storing denormalized or duplicate data in order to improve the performance or scalability of your solution is a valid approach. For more information about pricing, see [Azure Storage Pricing](#).

Next steps

- [Table Design Patterns](#)
- [Modeling relationships](#)
- [Design for querying](#)

- Encrypting Table Data
- Design for data modification

Design for querying

Article • 05/19/2023

Table service solutions may be read intensive, write intensive, or a mix of the two. This article focuses on the things to bear in mind when you are designing your Table service to support read operations efficiently. Typically, a design that supports read operations efficiently is also efficient for write operations. However, there are additional considerations to bear in mind when designing to support write operations, discussed in the article [Design for data modification](#).

A good starting point for designing your Table service solution to enable you to read data efficiently is to ask "What queries will my application need to execute to retrieve the data it needs from the Table service?"

ⓘ Note

With the Table service, it's important to get the design correct up front because it's difficult and expensive to change it later. For example, in a relational database it's often possible to address performance issues simply by adding indexes to an existing database: this is not an option with the Table service.

This section focuses on the key issues you must address when you design your tables for querying. The topics covered in this section include:

- [How your choice of PartitionKey and RowKey impacts query performance](#)
- [Choosing an appropriate PartitionKey](#)
- [Optimizing queries for the Table service](#)
- [Sorting data in the Table service](#)

How your choice of PartitionKey and RowKey impacts query performance

The following examples assume the table service is storing employee entities with the following structure (most of the examples omit the **Timestamp** property for clarity):

Column name	Data type
PartitionKey (Department Name)	String
RowKey (Employee ID)	String

Column name	Data type
FirstName	String
LastName	String
Age	Integer
EmailAddress	String

The article [Azure Table storage overview](#) describes some of the key features of the Azure Table service that have a direct influence on designing for query. These result in the following general guidelines for designing Table service queries. Note that the filter syntax used in the examples below is from the Table service REST API, for more information see [Query Entities](#).

- A **Point Query** is the most efficient lookup to use and is recommended to be used for high-volume lookups or lookups requiring lowest latency. Such a query can use the indexes to locate an individual entity very efficiently by specifying both the **PartitionKey** and **RowKey** values. For example: `$filter=(PartitionKey eq 'Sales') and (RowKey eq '2')`
- Second best is a **Range Query** that uses the **PartitionKey** and filters on a range of **RowKey** values to return more than one entity. The **PartitionKey** value identifies a specific partition, and the **RowKey** values identify a subset of the entities in that partition. For example: `$filter=PartitionKey eq 'Sales' and RowKey ge 'S' and RowKey lt 'T'`
- Third best is a **Partition Scan** that uses the **PartitionKey** and filters on another non-key property and that may return more than one entity. The **PartitionKey** value identifies a specific partition, and the property values select for a subset of the entities in that partition. For example: `$filter=PartitionKey eq 'Sales' and LastName eq 'Smith'`
- A **Table Scan** does not include the **PartitionKey** and is very inefficient because it searches all of the partitions that make up your table in turn for any matching entities. It will perform a table scan regardless of whether or not your filter uses the **RowKey**. For example: `$filter=LastName eq 'Jones'`
- Queries that return multiple entities return them sorted in **PartitionKey** and **RowKey** order. To avoid resorting the entities in the client, choose a **RowKey** that defines the most common sort order.

Note that using an "or" to specify a filter based on **RowKey** values results in a partition scan and is not treated as a range query. Therefore, you should avoid queries that use filters such as: `$filter=PartitionKey eq 'Sales' and (RowKey eq '121' or RowKey eq '322')`

For examples of client-side code that use the Storage Client Library to execute efficient queries, see:

- [Execute a point query using the Storage Client Library](#)
- [Retrieve multiple entities using LINQ](#)
- [Server-side projection](#)

For examples of client-side code that can handle multiple entity types stored in the same table, see:

- [Work with heterogeneous entity types](#)

Choosing an appropriate PartitionKey

Your choice of **PartitionKey** should balance the need to enable the use of entity group transactions (to ensure consistency) against the requirement to distribute your entities across multiple partitions (to ensure a scalable solution).

At one extreme, you could store all your entities in a single partition, but this may limit the scalability of your solution and would prevent the table service from being able to load-balance requests. At the other extreme, you could store one entity per partition, which would be highly scalable and which enables the table service to load-balance requests, but which would prevent you from using entity group transactions.

An ideal **PartitionKey** is one that enables you to use efficient queries and that has sufficient partitions to ensure your solution is scalable. Typically, you will find that your entities will have a suitable property that distributes your entities across sufficient partitions.

Note

For example, in a system that stores information about users or employees, UserID may be a good PartitionKey. You may have several entities that use a given UserID as the partition key. Each entity that stores data about a user is grouped into a single partition, and so these entities are accessible via entity group transactions, while still being highly scalable.

There are additional considerations in your choice of **PartitionKey** that relate to how you will insert, update, and delete entities. For more information, see [Designing tables for data modification](#).

Optimizing queries for the Table service

The Table service automatically indexes your entities using the **PartitionKey** and **RowKey** values in a single clustered index, hence the reason that point queries are the most efficient to use. However, there are no indexes other than that on the clustered index on the **PartitionKey** and **RowKey**.

Many designs must meet requirements to enable lookup of entities based on multiple criteria. For example, locating employee entities based on email, employee ID, or last name. The patterns described in [Table Design Patterns](#) address these types of requirement and describe ways of working around the fact that the Table service does not provide secondary indexes:

- [Intra-partition secondary index pattern](#) - Store multiple copies of each entity using different **RowKey** values (in the same partition) to enable fast and efficient lookups and alternate sort orders by using different **RowKey** values.
- [Inter-partition secondary index pattern](#) - Store multiple copies of each entity using different **RowKey** values in separate partitions or in separate tables to enable fast and efficient lookups and alternate sort orders by using different **RowKey** values.
- [Index Entities Pattern](#) - Maintain index entities to enable efficient searches that return lists of entities.

Sorting data in the Table service

The Table service returns entities sorted in ascending order based on **PartitionKey** and then by **RowKey**. These keys are string values and to ensure that numeric values sort correctly, you should convert them to a fixed length and pad them with zeroes. For example, if the employee ID value you use as the **RowKey** is an integer value, you should convert employee ID **123** to **00000123**.

Many applications have requirements to use data sorted in different orders: for example, sorting employees by name, or by joining date. The following patterns address how to alternate sort orders for your entities:

- [Intra-partition secondary index pattern](#) - Store multiple copies of each entity using different **RowKey** values (in the same partition) to enable fast and efficient lookups and alternate sort orders by using different **RowKey** values.
- [Inter-partition secondary index pattern](#) - Store multiple copies of each entity using different **RowKey** values in separate partitions in separate tables to enable fast and efficient lookups and alternate sort orders by using different **RowKey** values.
- [Log tail pattern](#) - Retrieve the n entities most recently added to a partition by using a **RowKey** value that sorts in reverse date and time order.

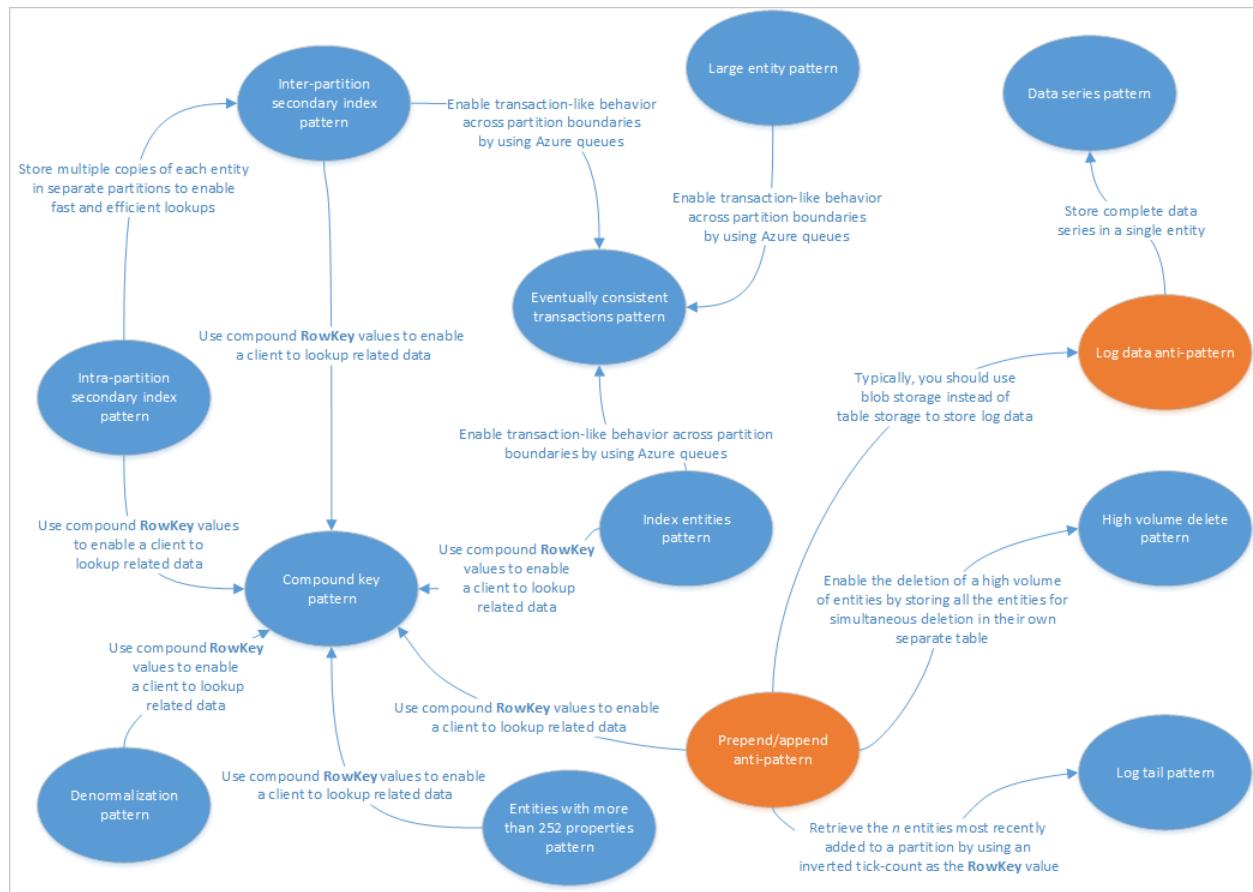
Next steps

- Table design patterns
- Modeling relationships
- Encrypt table data
- Design for data modification

Table design patterns

Article • 04/26/2023

This article describes some patterns appropriate for use with Table service solutions. Also, you will see how you can practically address some of the issues and trade-offs discussed in other Table storage design articles. The following diagram summarizes the relationships between the different patterns:



The pattern map above highlights some relationships between patterns (blue) and anti-patterns (orange) that are documented in this guide. There are many other patterns that are worth considering. For example, one of the key scenarios for Table Service is to use the [Materialized View Pattern](#) from the [Command Query Responsibility Segregation \(CQRS\)](#) pattern.

Intra-partition secondary index pattern

Store multiple copies of each entity using different **RowKey** values (in the same partition) to enable fast and efficient lookups and alternate sort orders by using different **RowKey** values. Updates between copies can be kept consistent using entity group transactions (EGTs).

Context and problem

The Table service automatically indexes entities using the **PartitionKey** and **RowKey** values. This enables a client application to retrieve an entity efficiently using these values. For example, using the table structure shown below, a client application can use a point query to retrieve an individual employee entity by using the department name and the employee ID (the **PartitionKey** and **RowKey** values). A client can also retrieve entities sorted by employee ID within each department.

Employee entity
PartitionKey (Department name)
RowKey (Employee Id)

FirstName (string)
LastName (string)
Age (integer)
EmailAddress (string)

If you also want to be able to find an employee entity based on the value of another property, such as email address, you must use a less efficient partition scan to find a match. This is because the table service does not provide secondary indexes. In addition, there is no option to request a list of employees sorted in a different order than **RowKey** order.

Solution

To work around the lack of secondary indexes, you can store multiple copies of each entity with each copy using a different **RowKey** value. If you store an entity with the structures shown below, you can efficiently retrieve employee entities based on email address or employee ID. The prefix values for the **RowKey**, "empid_" and "email_" enable you to query for a single employee or a range of employees by using a range of email addresses or employee IDs.

Employee entity
PartitionKey (Department name)
RowKey ("empid_" + Employee Id)

FirstName (string)
LastName (string)
Age (integer)
EmailAddress (string)

Employee entity
PartitionKey (Department name)
RowKey ("email_" + Email address)

FirstName (string)
LastName (string)
Age (integer)
EmployeeId (string)

The following two filter criteria (one looking up by employee ID and one looking up by email address) both specify point queries:

- \$filter=(PartitionKey eq 'Sales') and (RowKey eq 'empid_000223')

- `$filter=(PartitionKey eq 'Sales') and (RowKey eq 'email_jonesj@contoso.com')`

If you query for a range of employee entities, you can specify a range sorted in employee ID order, or a range sorted in email address order by querying for entities with the appropriate prefix in the **RowKey**.

- To find all the employees in the Sales department with an employee ID in the range 000100 to 000199 use: `$filter=(PartitionKey eq 'Sales') and (RowKey ge 'empid_000100') and (RowKey le 'empid_000199')`
- To find all the employees in the Sales department with an email address starting with the letter 'a' use: `$filter=(PartitionKey eq 'Sales') and (RowKey ge 'email_a') and (RowKey lt 'email_b')`

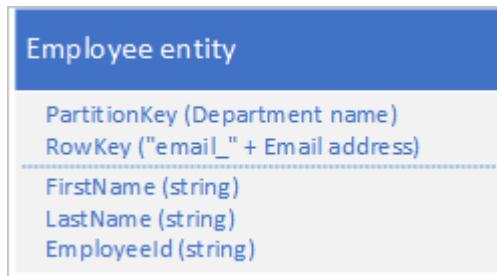
The filter syntax used in the examples above is from the Table service REST API, for more information, see [Query Entities](#).

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- Table storage is relatively cheap to use so the cost overhead of storing duplicate data should not be a major concern. However, you should always evaluate the cost of your design based on your anticipated storage requirements and only add duplicate entities to support the queries your client application will execute.
- Because the secondary index entities are stored in the same partition as the original entities, you should ensure that you do not exceed the scalability targets for an individual partition.
- You can keep your duplicate entities consistent with each other by using EGTs to update the two copies of the entity atomically. This implies that you should store all copies of an entity in the same partition. For more information, see the section [Using Entity Group Transactions](#).
- The value used for the **RowKey** must be unique for each entity. Consider using compound key values.
- Padding numeric values in the **RowKey** (for example, the employee ID 000223), enables correct sorting and filtering based on upper and lower bounds.
- You do not necessarily need to duplicate all the properties of your entity. For example, if the queries that look up the entities using the email address in the

RowKey never need the employee's age, these entities could have the following structure:



- It is typically better to store duplicate data and ensure that you can retrieve all the data you need with a single query, than to use one query to locate an entity and another to look up the required data.

When to use this pattern

Use this pattern when your client application needs to retrieve entities using a variety of different keys, when your client needs to retrieve entities in different sort orders, and where you can identify each entity using a variety of unique values. However, you should be sure that you do not exceed the partition scalability limits when you are performing entity lookups using the different RowKey values.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Inter-partition secondary index pattern](#)
- [Compound key pattern](#)
- Entity Group Transactions
- [Working with heterogeneous entity types](#)

Inter-partition secondary index pattern

Store multiple copies of each entity using different RowKey values in separate partitions or in separate tables to enable fast and efficient lookups and alternate sort orders by using different RowKey values.

Context and problem

The Table service automatically indexes entities using the **PartitionKey** and **RowKey** values. This enables a client application to retrieve an entity efficiently using these values. For example, using the table structure shown below, a client application can use a point query to retrieve an individual employee entity by using the department name and the employee ID (the **PartitionKey** and **RowKey** values). A client can also retrieve entities sorted by employee ID within each department.

Employee entity
PartitionKey (Department name)
RowKey (Employee Id)

FirstName (string)
LastName (string)
Age (integer)
EmailAddress (string)

If you also want to be able to find an employee entity based on the value of another property, such as email address, you must use a less efficient partition scan to find a match. This is because the table service does not provide secondary indexes. In addition, there is no option to request a list of employees sorted in a different order than **RowKey** order.

You are anticipating a high volume of transactions against these entities and want to minimize the risk of the Table service throttling your client.

Solution

To work around the lack of secondary indexes, you can store multiple copies of each entity with each copy using different **PartitionKey** and **RowKey** values. If you store an entity with the structures shown below, you can efficiently retrieve employee entities based on email address or employee ID. The prefix values for the **PartitionKey**, "empid_" and "email_" enable you to identify which index you want to use for a query.

Employee entity (primary index)	Employee entity (secondary index)
PartitionKey ("empid_" + Department name)	PartitionKey ("email_" + Department name)
RowKey (Employee Id)	RowKey (Email address)
-----	-----
FirstName (string)	FirstName (string)
LastName (string)	LastName (string)
Age (integer)	Age (integer)
EmailAddress (string)	EmployeeId (string)

The following two filter criteria (one looking up by employee ID and one looking up by email address) both specify point queries:

- `$filter=(PartitionKey eq 'empid_Sales') and (RowKey eq '000223')`
- `$filter=(PartitionKey eq 'email_Sales') and (RowKey eq 'jonesj@contoso.com')`

If you query for a range of employee entities, you can specify a range sorted in employee ID order, or a range sorted in email address order by querying for entities with the appropriate prefix in the **RowKey**.

- To find all the employees in the Sales department with an employee ID in the range **000100** to **000199** sorted in employee ID order use: `$filter=(PartitionKey eq 'empid_Sales') and (RowKey ge '000100') and (RowKey le '000199')`
- To find all the employees in the Sales department with an email address that starts with 'a' sorted in email address order use: `$filter=(PartitionKey eq 'email_Sales') and (RowKey ge 'a') and (RowKey lt 'b')`

The filter syntax used in the examples above is from the Table service REST API, for more information, see [Query Entities](#).

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- You can keep your duplicate entities eventually consistent with each other by using the [Eventually consistent transactions pattern](#) to maintain the primary and secondary index entities.
- Table storage is relatively cheap to use so the cost overhead of storing duplicate data should not be a major concern. However, you should always evaluate the cost of your design based on your anticipated storage requirements and only add duplicate entities to support the queries your client application will execute.
- The value used for the **RowKey** must be unique for each entity. Consider using compound key values.
- Padding numeric values in the **RowKey** (for example, the employee ID **000223**), enables correct sorting and filtering based on upper and lower bounds.
- You do not necessarily need to duplicate all the properties of your entity. For example, if the queries that look up the entities using the email address in the **RowKey** never need the employee's age, these entities could have the following structure:

Employee entity (secondary index)

`PartitionKey ("email_" + Department name)`

`RowKey (Email address)`

`FirstName (string)`

`LastName (string)`

`EmployeeId (string)`

- It is typically better to store duplicate data and ensure that you can retrieve all the data you need with a single query than to use one query to locate an entity using the secondary index and another to look up the required data in the primary index.

When to use this pattern

Use this pattern when your client application needs to retrieve entities using a variety of different keys, when your client needs to retrieve entities in different sort orders, and where you can identify each entity using a variety of unique values. Use this pattern when you want to avoid exceeding the partition scalability limits when you are performing entity lookups using the different **RowKey** values.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Eventually consistent transactions pattern](#)
- [Intra-partition secondary index pattern](#)
- [Compound key pattern](#)
- Entity Group Transactions
- [Working with heterogeneous entity types](#)

Eventually consistent transactions pattern

Enable eventually consistent behavior across partition boundaries or storage system boundaries by using Azure queues.

Context and problem

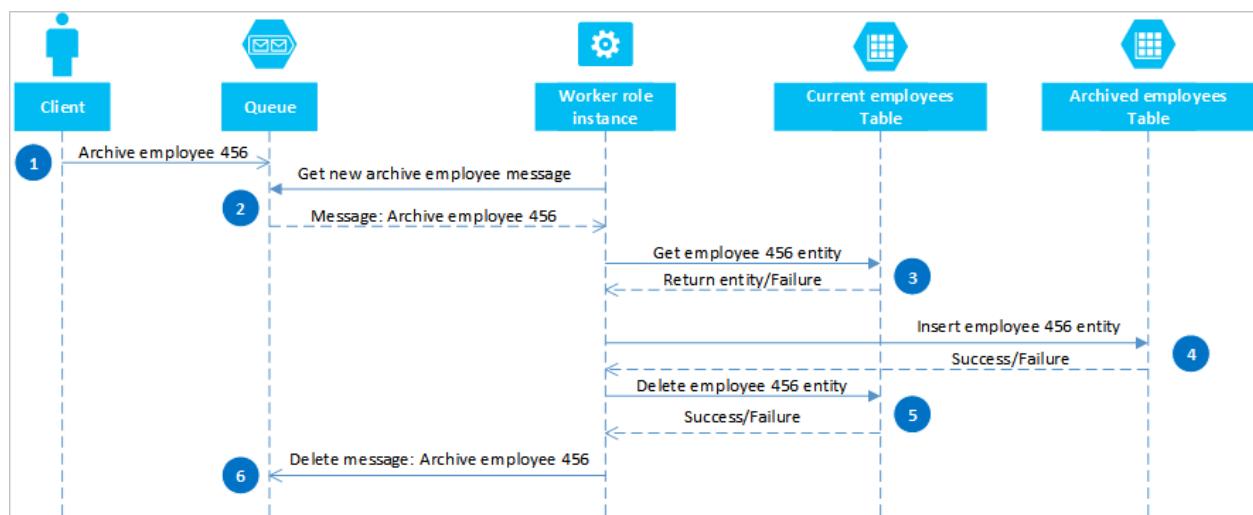
EGTs enable atomic transactions across multiple entities that share the same partition key. For performance and scalability reasons, you might decide to store entities that have consistency requirements in separate partitions or in a separate storage system: in such a scenario, you cannot use EGTs to maintain consistency. For example, you might have a requirement to maintain eventual consistency between:

- Entities stored in two different partitions in the same table, in different tables, or in different storage accounts.
- An entity stored in the Table service and a blob stored in the Blob service.
- An entity stored in the Table service and a file in a file system.

- An entity stored in the Table service yet indexed using the Azure Cognitive Search service.

Solution

By using Azure queues, you can implement a solution that delivers eventual consistency across two or more partitions or storage systems. To illustrate this approach, assume you have a requirement to be able to archive old employee entities. Old employee entities are rarely queried and should be excluded from any activities that deal with current employees. To implement this requirement, you store active employees in the **Current** table and old employees in the **Archive** table. Archiving an employee requires you to delete the entity from the **Current** table and add the entity to the **Archive** table, but you cannot use an EGT to perform these two operations. To avoid the risk that a failure causes an entity to appear in both or neither tables, the archive operation must be eventually consistent. The following sequence diagram outlines the steps in this operation. More detail is provided for exception paths in the text following.



A client initiates the archive operation by placing a message on an Azure queue, in this example to archive employee #456. A worker role polls the queue for new messages; when it finds one, it reads the message and leaves a hidden copy on the queue. The worker role next fetches a copy of the entity from the **Current** table, inserts a copy in the **Archive** table, and then deletes the original from the **Current** table. Finally, if there were no errors from the previous steps, the worker role deletes the hidden message from the queue.

In this example, step 4 inserts the employee into the **Archive** table. It could add the employee to a blob in the Blob service or a file in a file system.

Recovering from failures

It is important that the operations in steps 4 and 5 must be *idempotent* in case the worker role needs to restart the archive operation. If you are using the Table service, for step 4 you should use an "insert or replace" operation; for step 5 you should use a "delete if exists" operation in the client library you are using. If you are using another storage system, you must use an appropriate idempotent operation.

If the worker role never completes step 6, then after a timeout the message reappears on the queue ready for the worker role to try to reprocess it. The worker role can check how many times a message on the queue has been read and, if necessary, flag it as a "poison" message for investigation by sending it to a separate queue. For more information about reading queue messages and checking the dequeue count, see [Get Messages](#).

Some errors from the Table and Queue services are transient errors, and your client application should include suitable retry logic to handle them.

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- This solution does not provide for transaction isolation. For example, a client could read the **Current** and **Archive** tables when the worker role was between steps 4 and 5, and see an inconsistent view of the data. The data will be consistent eventually.
- You must be sure that steps 4 and 5 are idempotent in order to ensure eventual consistency.
- You can scale the solution by using multiple queues and worker role instances.

When to use this pattern

Use this pattern when you want to guarantee eventual consistency between entities that exist in different partitions or tables. You can extend this pattern to ensure eventual consistency for operations across the Table service and the Blob service and other non-Azure Storage data sources such as database or the file system.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- Entity Group Transactions
- [Merge or replace](#)

(!) Note

If transaction isolation is important to your solution, you should consider redesigning your tables to enable you to use EGTs.

Index entities pattern

Maintain index entities to enable efficient searches that return lists of entities.

Context and problem

The Table service automatically indexes entities using the **PartitionKey** and **RowKey** values. This enables a client application to retrieve an entity efficiently using a point query. For example, using the table structure shown below, a client application can efficiently retrieve an individual employee entity by using the department name and the employee ID (the **PartitionKey** and **RowKey**).

Employee entity
PartitionKey (Department name)
RowKey (Employee Id)
FirstName (string)
LastName (string)
Age (integer)
EmailAddress (string)

If you also want to be able to retrieve a list of employee entities based on the value of another non-unique property, such as their last name, you must use a less efficient partition scan to find matches rather than using an index to look them up directly. This is because the table service does not provide secondary indexes.

Solution

To enable lookup by last name with the entity structure shown above, you must maintain lists of employee IDs. If you want to retrieve the employee entities with a particular last name, such as Jones, you must first locate the list of employee IDs for employees with Jones as their last name, and then retrieve those employee entities. There are three main options for storing the lists of employee IDs:

- Use blob storage.
- Create index entities in the same partition as the employee entities.
- Create index entities in a separate partition or table.

Option #1: Use blob storage

For the first option, you create a blob for every unique last name, and in each blob store a list of the **PartitionKey** (department) and **RowKey** (employee ID) values for employees that have that last name. When you add or delete an employee, you should ensure that the content of the relevant blob is eventually consistent with the employee entities.

Option #2: Create index entities in the same partition

For the second option, use index entities that store the following data:

Employee index entity
PartitionKey (Department name)
RowKey (LastName)
EmployeeIDs (String containing a list of employee ids with same last name)

The **EmployeeIDs** property contains a list of employee IDs for employees with the last name stored in the **RowKey**.

The following steps outline the process you should follow when you are adding a new employee if you are using the second option. In this example, we are adding an employee with ID 000152 and a last name Jones in the Sales department:

1. Retrieve the index entity with a **PartitionKey** value "Sales" and the **RowKey** value "Jones." Save the ETag of this entity to use in step 2.
2. Create an entity group transaction (that is, a batch operation) that inserts the new employee entity (**PartitionKey** value "Sales" and **RowKey** value "000152"), and updates the index entity (**PartitionKey** value "Sales" and **RowKey** value "Jones") by adding the new employee ID to the list in the **EmployeeIDs** field. For more information about entity group transactions, see Entity Group Transactions.
3. If the entity group transaction fails because of an optimistic concurrency error (someone else has just modified the index entity), then you need to start over at step 1 again.

You can use a similar approach to deleting an employee if you are using the second option. Changing an employee's last name is slightly more complex because you will need to execute an entity group transaction that updates three entities: the employee

entity, the index entity for the old last name, and the index entity for the new last name. You must retrieve each entity before making any changes in order to retrieve the ETag values that you can then use to perform the updates using optimistic concurrency.

The following steps outline the process you should follow when you need to look up all the employees with a given last name in a department if you are using the second option. In this example, we are looking up all the employees with last name Jones in the Sales department:

1. Retrieve the index entity with a **PartitionKey** value "Sales" and the **RowKey** value "Jones."
2. Parse the list of employee IDs in the EmployeeIDs field.
3. If you need additional information about each of these employees (such as their email addresses), retrieve each of the employee entities using **PartitionKey** value "Sales" and **RowKey** values from the list of employees you obtained in step 2.

Option #3: Create index entities in a separate partition or table

For the third option, use index entities that store the following data:

Employee index entity
PartitionKey ("indexentities") RowKey (LastName) ----- EmployeeDetails (A list of (EmployeeIDs, DepartmentName) pairs)

The **EmployeeDetails** property contains a list of employee IDs and department name pairs for employees with the last name stored in the **RowKey**.

With the third option, you cannot use EGTs to maintain consistency because the index entities are in a separate partition from the employee entities. Ensure that the index entities are eventually consistent with the employee entities.

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- This solution requires at least two queries to retrieve matching entities: one to query the index entities to obtain the list of **RowKey** values, and then queries to retrieve each entity in the list.

- Given that an individual entity has a maximum size of 1 MB, option #2 and option #3 in the solution assume that the list of employee IDs for any given last name is never greater than 1 MB. If the list of employee IDs is likely to be greater than 1 MB in size, use option #1 and store the index data in blob storage.
- If you use option #2 (using EGTs to handle adding and deleting employees, and changing an employee's last name) you must evaluate if the volume of transactions will approach the scalability limits in a given partition. If this is the case, you should consider an eventually consistent solution (option #1 or option #3) that uses queues to handle the update requests and enables you to store your index entities in a separate partition from the employee entities.
- Option #2 in this solution assumes that you want to look up by last name within a department: for example, you want to retrieve a list of employees with a last name Jones in the Sales department. If you want to be able to look up all the employees with a last name Jones across the whole organization, use either option #1 or option #3.
- You can implement a queue-based solution that delivers eventual consistency (see the [Eventually consistent transactions pattern](#) for more details).

When to use this pattern

Use this pattern when you want to look up a set of entities that all share a common property value, such as all employees with the last name Jones.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Compound key pattern](#)
- [Eventually consistent transactions pattern](#)
- Entity Group Transactions
- [Working with heterogeneous entity types](#)

Denormalization pattern

Combine related data together in a single entity to enable you to retrieve all the data you need with a single point query.

Context and problem

In a relational database, you typically normalize data to remove duplication resulting in queries that retrieve data from multiple tables. If you normalize your data in Azure tables, you must make multiple round trips from the client to the server to retrieve your related data. For example, with the table structure shown below you need two round trips to retrieve the details for a department: one to fetch the department entity that includes the manager's ID, and then another request to fetch the manager's details in an employee entity.

Department entity	Employee entity
PartitionKey (Department name) RowKey ("Department") ----- DepartmentName (string) EmployeeCount (integer) ManagerId (string - employee id of manager)	PartitionKey (Department name) RowKey (Employee Id) ----- FirstName (string) LastName (string) Age (integer) EmailAddress (string)

Solution

Instead of storing the data in two separate entities, denormalize the data and keep a copy of the manager's details in the department entity. For example:

Department entity
PartitionKey (Department name) RowKey ("Department") ----- DepartmentName (string) EmployeeCount (integer) ManagerName (string) ManagerEmailAddress (string)

With department entities stored with these properties, you can now retrieve all the details you need about a department using a point query.

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- There is some cost overhead associated with storing some data twice. The performance benefit (resulting from fewer requests to the storage service) typically outweighs the marginal increase in storage costs (and this cost is partially offset by a reduction in the number of transactions you require to fetch the details of a department).
- You must maintain the consistency of the two entities that store information about managers. You can handle the consistency issue by using EGTs to update multiple

entities in a single atomic transaction: in this case, the department entity, and the employee entity for the department manager are stored in the same partition.

When to use this pattern

Use this pattern when you frequently need to look up related information. This pattern reduces the number of queries your client must make to retrieve the data it requires.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Compound key pattern](#)
- [Entity Group Transactions](#)
- [Working with heterogeneous entity types](#)

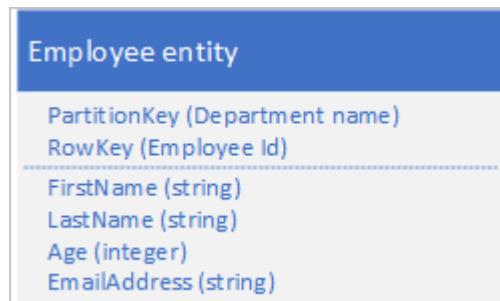
Compound key pattern

Use compound **RowKey** values to enable a client to look up related data with a single point query.

Context and problem

In a relational database, it is natural to use joins in queries to return related pieces of data to the client in a single query. For example, you might use the employee ID to look up a list of related entities that contain performance and review data for that employee.

Assume you are storing employee entities in the Table service using the following structure:



You also need to store historical data relating to reviews and performance for each year the employee has worked for your organization and you need to be able to access this

information by year. One option is to create another table that stores entities with the following structure:

Employee review entity	
PartitionKey (Employee Id)	
RowKey (Year)	
FirstName (string)	
LastName (string)	
ManagerRating (integer)	
PeerRating (integer)	
Comments (string)	

Notice that with this approach you may decide to duplicate some information (such as first name and last name) in the new entity to enable you to retrieve your data with a single request. However, you cannot maintain strong consistency because you cannot use an EGT to update the two entities atomically.

Solution

Store a new entity type in your original table using entities with the following structure:

Employee entity	
PartitionKey (Department name)	
RowKey (Employee Id + Year)	
FirstName (string)	
LastName (string)	
Age (integer)	
EmailAddress (string)	
ManagerRating (integer)	
PeerRating (integer)	
Comments (string)	

Notice how the **RowKey** is now a compound key made up of the employee ID and the year of the review data that enables you to retrieve the employee's performance and review data with a single request for a single entity.

The following example outlines how you can retrieve all the review data for a particular employee (such as employee 000123 in the Sales department):

```
$filter=(PartitionKey eq 'Sales') and (RowKey ge 'empid_000123') and (RowKey lt '000123_2012')&$select=RowKey,Manager Rating,Peer Rating,Comments
```

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- You should use a suitable separator character that makes it easy to parse the **RowKey** value: for example, **000123_2012**.
- You are also storing this entity in the same partition as other entities that contain related data for the same employee, which means you can use EGTs to maintain strong consistency.
- You should consider how frequently you will query the data to determine whether this pattern is appropriate. For example, if you will access the review data infrequently and the main employee data often you should keep them as separate entities.

When to use this pattern

Use this pattern when you need to store one or more related entities that you query frequently.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- Entity Group Transactions
- [Working with heterogeneous entity types](#)
- [Eventually consistent transactions pattern](#)

Log tail pattern

Retrieve the n entities most recently added to a partition by using a **RowKey** value that sorts in reverse date and time order.

Context and problem

A common requirement is been able to retrieve the most recently created entities, for example the 10 most recent expense claims submitted by an employee. Table queries support a **\$top** query operation to return the first n entities from a set: there is no equivalent query operation to return the last n entities in a set.

Solution

Store the entities using a **RowKey** that naturally sorts in reverse date/time order by using so the most recent entry is always the first one in the table.

For example, to be able to retrieve the 10 most recent expense claims submitted by an employee, you can use a reverse tick value derived from the current date/time. The following C# code sample shows one way to create a suitable "inverted ticks" value for a RowKey that sorts from the most recent to the oldest:

```
string invertedTicks = string.Format("{0:D19}", DateTime.MaxValue.Ticks -  
DateTime.UtcNow.Ticks);
```

You can get back to the date time value using the following code:

```
DateTime dt = new DateTime(DateTime.MaxValue.Ticks - Int64.Parse(invertedTicks));
```

The table query looks like this:

```
https://myaccount.table.core.windows.net/EmployeeExpense(PartitionKey='empid')?  
$top=10
```

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- You must pad the reverse tick value with leading zeroes to ensure the string value sorts as expected.
- You must be aware of the scalability targets at the level of a partition. Be careful not create hot spot partitions.

When to use this pattern

Use this pattern when you need to access entities in reverse date/time order or when you need to access the most recently added entities.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Prepend / append anti-pattern](#)
- [Retrieving entities](#)

High volume delete pattern

Enable the deletion of a high volume of entities by storing all the entities for simultaneous deletion in their own separate table; you delete the entities by deleting the table.

Context and problem

Many applications delete old data that no longer needs to be available to a client application, or that the application has archived to another storage medium. You typically identify such data by a date: for example, you have a requirement to delete records of all login requests that are more than 60 days old.

One possible design is to use the date and time of the login request in the **RowKey**:

Login attempt entity
PartitionKey (Username)
RowKey (Date and time of login attempt)
SourceIPAddress (string)
SuccessfulLogin (boolean)

This approach avoids partition hotspots because the application can insert and delete login entities for each user in a separate partition. However, this approach may be costly and time consuming if you have a large number of entities because first you need to perform a table scan in order to identify all the entities to delete, and then you must delete each old entity. You can reduce the number of round trips to the server required to delete the old entities by batching multiple delete requests into EGTs.

Solution

Use a separate table for each day of login attempts. You can use the entity design above to avoid hotspots when you are inserting entities, and deleting old entities is now simply a question of deleting one table every day (a single storage operation) instead of finding and deleting hundreds and thousands of individual login entities every day.

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- Does your design support other ways your application will use the data such as looking up specific entities, linking with other data, or generating aggregate information?
- Does your design avoid hot spots when you are inserting new entities?

- Expect a delay if you want to reuse the same table name after deleting it. It's better to always use unique table names.
- Expect some throttling when you first use a new table while the Table service learns the access patterns and distributes the partitions across nodes. You should consider how frequently you need to create new tables.

When to use this pattern

Use this pattern when you have a high volume of entities that you must delete at the same time.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

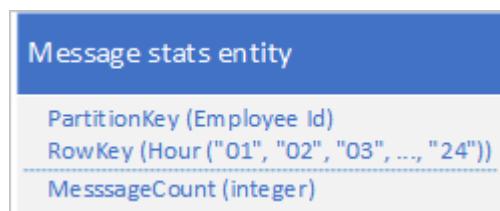
- Entity Group Transactions
- [Modifying entities](#)

Data series pattern

Store complete data series in a single entity to minimize the number of requests you make.

Context and problem

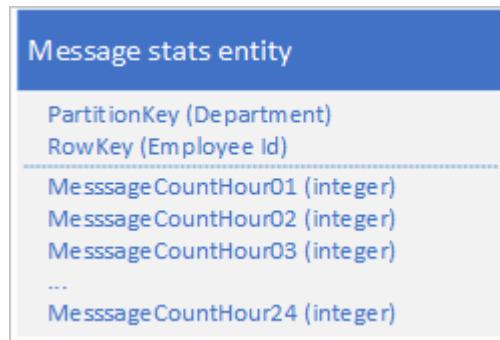
A common scenario is for an application to store a series of data that it typically needs to retrieve all at once. For example, your application might record how many IM messages each employee sends every hour, and then use this information to plot how many messages each user sent over the preceding 24 hours. One design might be to store 24 entities for each employee:



With this design, you can easily locate and update the entity to update for each employee whenever the application needs to update the message count value. However, to retrieve the information to plot a chart of the activity for the preceding 24 hours, you must retrieve 24 entities.

Solution

Use the following design with a separate property to store the message count for each hour:



With this design, you can use a merge operation to update the message count for an employee for a specific hour. Now, you can retrieve all the information you need to plot the chart using a request for a single entity.

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- If your complete data series does not fit into a single entity (an entity can have up to 252 properties), use an alternative data store such as a blob.
- If you have multiple clients updating an entity simultaneously, you will need to use the **ETag** to implement optimistic concurrency. If you have many clients, you may experience high contention.

When to use this pattern

Use this pattern when you need to update and retrieve a data series associated with an individual entity.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Large entities pattern](#)
- [Merge or replace](#)
- [Eventually consistent transactions pattern](#) (if you are storing the data series in a blob)

Wide entities pattern

Use multiple physical entities to store logical entities with more than 252 properties.

Context and problem

An individual entity can have no more than 252 properties (excluding the mandatory system properties) and cannot store more than 1 MB of data in total. In a relational database, you would typically get round any limits on the size of a row by adding a new table and enforcing a 1-to-1 relationship between them.

Solution

Using the Table service, you can store multiple entities to represent a single large business object with more than 252 properties. For example, if you want to store a count of the number of IM messages sent by each employee for the last 365 days, you could use the following design that uses two entities with different schemas:

Message stats entity	Message stats entity
<code>PartitionKey (Department)</code> <code>RowKey (Employee Id + "_01")</code> <code>MessageCountDay01 (integer)</code> <code>MessageCountDay02 (integer)</code> <code>MessageCountDay03 (integer)</code> ... <code>MessageCountDay252 (integer)</code>	<code>PartitionKey (Department)</code> <code>RowKey (Employee Id + "_02")</code> <code>MessageCountDay253 (integer)</code> <code>MessageCountDay254 (integer)</code> <code>MessageCountDay255 (integer)</code> ... <code>MessageCountDay365 (integer)</code>

If you need to make a change that requires updating both entities to keep them synchronized with each other, you can use an EGT. Otherwise, you can use a single merge operation to update the message count for a specific day. To retrieve all the data for an individual employee you must retrieve both entities, which you can do with two efficient requests that use both a **PartitionKey** and a **RowKey** value.

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- Retrieving a complete logical entity involves at least two storage transactions: one to retrieve each physical entity.

When to use this pattern

Use this pattern when need to store entities whose size or number of properties exceeds the limits for an individual entity in the Table service.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- Entity Group Transactions
- [Merge or replace](#)

Large entities pattern

Use blob storage to store large property values.

Context and problem

An individual entity cannot store more than 1 MB of data in total. If one or several of your properties store values that cause the total size of your entity to exceed this value, you cannot store the entire entity in the Table service.

Solution

If your entity exceeds 1 MB in size because one or more properties contain a large amount of data, you can store data in the Blob service and then store the address of the blob in a property in the entity. For example, you can store the photo of an employee in blob storage and store a link to the photo in the **Photo** property of your employee entity:

Employee entity
PartitionKey (Department name)
RowKey (Email address)

FirstName (string)
LastName (string)
Age (integer)
Photo (string)

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- To maintain eventual consistency between the entity in the Table service and the data in the Blob service, use the [Eventually consistent transactions pattern](#) to maintain your entities.
- Retrieving a complete entity involves at least two storage transactions: one to retrieve the entity and one to retrieve the blob data.

When to use this pattern

Use this pattern when you need to store entities whose size exceeds the limits for an individual entity in the Table service.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Eventually consistent transactions pattern](#)
- [Wide entities pattern](#)

Prepend/append anti-pattern

Increase scalability when you have a high volume of inserts by spreading the inserts across multiple partitions.

Context and problem

Prepending or appending entities to your stored entities typically results in the application adding new entities to the first or last partition of a sequence of partitions. In this case, all of the inserts at any given time are taking place in the same partition, creating a hotspot that prevents the table service from load-balancing inserts across multiple nodes, and possibly causing your application to hit the scalability targets for partition. For example, if you have an application that logs network and resource access by employees, then an entity structure as shown below could result in the current hour's partition becoming a hotspot if the volume of transactions reaches the scalability target for an individual partition:

Employee entity

PartitionKey (Year + Month + Day + Hour)

RowKey (Employee Id + Event Id)

FirstName (string)

LastName (string)

EventType (string)

EventTimestamp (datetime)

EventText (string)

Solution

The following alternative entity structure avoids a hotspot on any particular partition as the application logs events:

Employee entity

PartitionKey (Department name + Employee Id)

RowKey (Year + Month + Day + Hour + Event Id)

FirstName (string)

LastName (string)

EventType (string)

EventTimestamp (datetime)

EventText (string)

Notice with this example how both the **PartitionKey** and **RowKey** are compound keys. The **PartitionKey** uses both the department and employee ID to distribute the logging across multiple partitions.

Issues and considerations

Consider the following points when deciding how to implement this pattern:

- Does the alternative key structure that avoids creating hot partitions on inserts efficiently support the queries your client application makes?
- Does your anticipated volume of transactions mean that you are likely to reach the scalability targets for an individual partition and be throttled by the storage service?

When to use this pattern

Avoid the prepend/append anti-pattern when your volume of transactions is likely to result in throttling by the storage service when you access a hot partition.

Related patterns and guidance

The following patterns and guidance may also be relevant when implementing this pattern:

- [Compound key pattern](#)
- [Log tail pattern](#)
- [Modifying entities](#)

Log data anti-pattern

Typically, you should use the Blob service instead of the Table service to store log data.

Context and problem

A common use case for log data is to retrieve a selection of log entries for a specific date/time range: for example, you want to find all the error and critical messages that your application logged between 15:04 and 15:06 on a specific date. You do not want to use the date and time of the log message to determine the partition you save log entities to: that results in a hot partition because at any given time, all the log entities will share the same **PartitionKey** value (see the section [Prepend/append anti-pattern](#)). For example, the following entity schema for a log message results in a hot partition because the application writes all log messages to the partition for the current date and hour:

Log message entity
PartitionKey (Date and hour of log message)
RowKey (Time of log message and unique log message id (GUID))

Severity (integer)
Source (string)
Message (string)
Department (string)
ClientIP (string)

In this example, the **RowKey** includes the date and time of the log message to ensure that log messages are stored sorted in date/time order, and includes a message ID in case multiple log messages share the same date and time.

Another approach is to use a **PartitionKey** that ensures that the application writes messages across a range of partitions. For example, if the source of the log message provides a way to distribute messages across many partitions, you could use the following entity schema:

Log message entity
PartitionKey (Source)
RowKey (Time of log message and unique log message id (GUID))
Severity (integer)
Message (string)
Department (string)
ClientIP (string)

However, the problem with this schema is that to retrieve all the log messages for a specific time span you must search every partition in the table.

Solution

The previous section highlighted the problem of trying to use the Table service to store log entries and suggested two, unsatisfactory, designs. One solution led to a hot partition with the risk of poor performance writing log messages; the other solution resulted in poor query performance because of the requirement to scan every partition in the table to retrieve log messages for a specific time span. Blob storage offers a better solution for this type of scenario and this is how Azure Storage Analytics stores the log data it collects.

This section outlines how Storage Analytics stores log data in blob storage as an illustration of this approach to storing data that you typically query by range.

Storage Analytics stores log messages in a delimited format in multiple blobs. The delimited format makes it easy for a client application to parse the data in the log message.

Storage Analytics uses a naming convention for blobs that enables you to locate the blob (or blobs) that contain the log messages for which you are searching. For example, a blob named "queue/2014/07/31/1800/000001.log" contains log messages that relate to the queue service for the hour starting at 18:00 on 31 July 2014. The "000001" indicates that this is the first log file for this period. Storage Analytics also records the timestamps of the first and last log messages stored in the file as part of the blob's metadata. The API for blob storage enables you to locate blobs in a container based on a name prefix: to locate all the blobs that contain queue log data for the hour starting at 18:00, you can use the prefix "queue/2014/07/31/1800."

Storage Analytics buffers log messages internally and then periodically updates the appropriate blob or creates a new one with the latest batch of log entries. This reduces the number of writes it must perform to the blob service.

If you are implementing a similar solution in your own application, you must consider how to manage the trade-off between reliability (writing every log entry to blob storage as it happens) and cost and scalability (buffering updates in your application and writing them to blob storage in batches).

Issues and considerations

Consider the following points when deciding how to store log data:

- If you create a table design that avoids potential hot partitions, you may find that you cannot access your log data efficiently.
- To process log data, a client often needs to load many records.
- Although log data is often structured, blob storage may be a better solution.

Implementation considerations

This section discusses some of the considerations to bear in mind when you implement the patterns described in the previous sections. Most of this section uses examples written in C# that use the Storage client library (version 4.3.0 at the time of writing).

Retrieving entities

As discussed in the section Design for querying, the most efficient query is a point query. However, in some scenarios you may need to retrieve multiple entities. This section describes some common approaches to retrieving entities using the Storage client library.

Executing a point query using the Storage client library

The easiest way to execute a point query is to use the `GetEntityAsync` method as shown in the following C# code snippet that retrieves an entity with a `PartitionKey` of value "Sales" and a `RowKey` of value "212":

C#

```
EmployeeEntity employee = await employeeTable.GetEntityAsync<EmployeeEntity>("Sales", "212");
```

Notice how this example expects the entity it retrieves to be of type `EmployeeEntity`.

Retrieving multiple entities using LINQ

You can use LINQ to retrieve multiple entities from the Table service when working with Microsoft Azure Cosmos DB Table Standard Library.

Azure CLI

```
dotnet add package Azure.Data.Tables
```

To make the below examples work, you'll need to include namespaces:

C#

```
using System.Linq;
using Azure.Data.Tables
```

Retrieving multiple entities can be achieved by specifying a query with a `filter` clause. To avoid a table scan, you should always include the `PartitionKey` value in the filter clause, and if possible the `RowKey` value to avoid table and partition scans. The table service supports a limited set of comparison operators (greater than, greater than or equal, less than, less than or equal, equal, and not equal) to use in the filter clause.

In the following example, `employeeTable` is a `TableClient` object. This example finds all the employees whose last name starts with "B" (assuming that the `RowKey` stores the last name) in the sales department (assuming the `PartitionKey` stores the department name):

C#

```
var employees = employeeTable.Query<EmployeeEntity>(e => (e.PartitionKey == "Sales" && e.RowKey.CompareTo("B") >= 0 && e.RowKey.CompareTo("C") < 0));
```

Notice how the query specifies both a `RowKey` and a `PartitionKey` to ensure better performance.

The following code sample shows equivalent functionality without using LINQ syntax:

C#

```
var employees = employeeTable.Query<EmployeeEntity>(filter: $"PartitionKey eq 'Sales' and RowKey ge 'B' and RowKey lt 'C'");
```

(!) Note

The sample `Query` methods include the three filter conditions.

Retrieving large numbers of entities from a query

An optimal query returns an individual entity based on a `PartitionKey` value and a `RowKey` value. However, in some scenarios you may have a requirement to return many entities from the same partition or even from many partitions.

You should always fully test the performance of your application in such scenarios.

A query against the table service may return a maximum of 1,000 entities at one time and may execute for a maximum of five seconds. If the result set contains more than 1,000 entities, if the query did not complete within five seconds, or if the query crosses the partition boundary, the Table service returns a continuation token to enable the client application to request the next set of entities. For more information about how continuation tokens work, see [Query Timeout and Pagination](#).

If you are using the Azure Tables client library, it can automatically handle continuation tokens for you as it returns entities from the Table service. The following C# code sample using the client library automatically handles continuation tokens if the table service returns them in a response:

C#

```
var employees = employeeTable.Query<EmployeeEntity>("PartitionKey eq 'Sales'")  
  
foreach (var emp in employees)  
{  
    // ...  
}
```

You can also specify the maximum number of entities that are returned per page. The following example shows how to query entities with `maxPerPage`:

C#

```
var employees = employeeTable.Query<EmployeeEntity>(maxPerPage: 10);  
  
// Iterate the Pageable object by page  
foreach (var page in employees.AsPages())  
{
```

```
// Iterate the entities returned for this page
foreach (var emp in page.Values)
{
    // ...
}
}
```

In more advanced scenarios, you may want to store the continuation token returned from the service so that your code controls exactly when the next pages is fetched. The following example shows a basic scenario of how the token can be fetched and applied to paginated results:

C#

```
string continuationToken = null;
bool moreResultsAvailable = true;
while (moreResultsAvailable)
{
    var page = employeeTable
        .Query<EmployeeEntity>()
        .AsPages(continuationToken, pageSizeHint: 10)
        .FirstOrDefault(); // pageSizeHint limits the number of results in a
single page, so we only enumerate the first page

    if (page == null)
        break;

    // Get the continuation token from the page
    // Note: This value can be stored so that the next page query can be
executed later
    continuationToken = page.ContinuationToken;

    var pageResults = page.Values;
    moreResultsAvailable = pageResults.Any() && continuationToken != null;

    // Iterate the results for this page
    foreach (var result in pageResults)
    {
        // ...
    }
}
```

By using continuation tokens explicitly, you can control when your application retrieves the next segment of data. For example, if your client application enables users to page through the entities stored in a table, a user may decide not to page through all the entities retrieved by the query so your application would only use a continuation token to retrieve the next segment when the user had finished paging through all the entities in the current segment. This approach has several benefits:

- It enables you to limit the amount of data to retrieve from the Table service and that you move over the network.
- It enables you to perform asynchronous IO in .NET.
- It enables you to serialize the continuation token to persistent storage so you can continue in the event of an application crash.

ⓘ Note

A continuation token typically returns a segment containing 1,000 entities, although it may be fewer. This is also the case if you limit the number of entries a query returns by using **Take** to return the first n entities that match your lookup criteria: the table service may return a segment containing fewer than n entities along with a continuation token to enable you to retrieve the remaining entities.

Server-side projection

A single entity can have up to 255 properties and be up to 1 MB in size. When you query the table and retrieve entities, you may not need all the properties and can avoid transferring data unnecessarily (to help reduce latency and cost). You can use server-side projection to transfer just the properties you need. The following example retrieves just the **Email** property (along with **PartitionKey**, **RowKey**, **Timestamp**, and **ETag**) from the entities selected by the query.

C#

```
var subsetResults = query{
    for employee in employeeTable.Query<EmployeeEntity>("PartitionKey eq
'Sales'") do
    select employee.Email
}
foreach (var e in subsetResults)
{
    Console.WriteLine("RowKey: {0}, EmployeeEmail: {1}", e.RowKey, e.Email);
}
```

Notice how the **RowKey** value is available even though it was not included in the list of properties to retrieve.

Modifying entities

The Storage client library enables you to modify your entities stored in the table service by inserting, deleting, and updating entities. You can use EGTs to batch multiple inserts,

update, and delete operations together to reduce the number of round trips required and improve the performance of your solution.

Exceptions thrown when the Storage client library executes an EGT typically include the index of the entity that caused the batch to fail. This is helpful when you are debugging code that uses EGTs.

You should also consider how your design affects how your client application handles concurrency and update operations.

Managing concurrency

By default, the table service implements optimistic concurrency checks at the level of individual entities for **Insert**, **Merge**, and **Delete** operations, although it is possible for a client to force the table service to bypass these checks. For more information about how the table service manages concurrency, see [Managing Concurrency in Microsoft Azure Storage](#).

Merge or replace

The **Replace** method of the **TableOperation** class always replaces the complete entity in the Table service. If you do not include a property in the request when that property exists in the stored entity, the request removes that property from the stored entity. Unless you want to remove a property explicitly from a stored entity, you must include every property in the request.

You can use the **Merge** method of the **TableOperation** class to reduce the amount of data that you send to the Table service when you want to update an entity. The **Merge** method replaces any properties in the stored entity with property values from the entity included in the request, but leaves intact any properties in the stored entity that are not included in the request. This is useful if you have large entities and only need to update a small number of properties in a request.

Note

The **Replace** and **Merge** methods fail if the entity does not exist. As an alternative, you can use the **InsertOrReplace** and **InsertOrMerge** methods that create a new entity if it doesn't exist.

Working with heterogeneous entity types

The Table service is a *schema-less* table store that means that a single table can store entities of multiple types providing great flexibility in your design. The following example illustrates a table storing both employee and department entities:

PartitionKey	RowKey	Timestamp									
			<table border="1"><thead><tr><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr></tbody></table>	FirstName	LastName	Age	Email				
FirstName	LastName	Age	Email								
			<table border="1"><thead><tr><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr></tbody></table>	FirstName	LastName	Age	Email				
FirstName	LastName	Age	Email								
			<table border="1"><thead><tr><th>DepartmentName</th><th>EmployeeCount</th></tr></thead><tbody><tr><td></td><td></td></tr></tbody></table>	DepartmentName	EmployeeCount						
DepartmentName	EmployeeCount										
			<table border="1"><thead><tr><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr></tbody></table>	FirstName	LastName	Age	Email				
FirstName	LastName	Age	Email								

Each entity must still have **PartitionKey**, **RowKey**, and **Timestamp** values, but may have any set of properties. Furthermore, there is nothing to indicate the type of an entity unless you choose to store that information somewhere. There are two options for identifying the entity type:

- Prepend the entity type to the **RowKey** (or possibly the **PartitionKey**). For example, **EMPLOYEE_000123** or **DEPARTMENT_SALES** as **RowKey** values.
- Use a separate property to record the entity type as shown in the table below.

PartitionKey	RowKey	Timestamp											
			<table border="1"><thead><tr><th>EntityType</th><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td>Employee</td><td></td><td></td><td></td><td></td></tr></tbody></table>	EntityType	FirstName	LastName	Age	Email	Employee				
EntityType	FirstName	LastName	Age	Email									
Employee													
			<table border="1"><thead><tr><th>EntityType</th><th>FirstName</th><th>LastName</th><th>Age</th><th>Email</th></tr></thead><tbody><tr><td>Employee</td><td></td><td></td><td></td><td></td></tr></tbody></table>	EntityType	FirstName	LastName	Age	Email	Employee				
EntityType	FirstName	LastName	Age	Email									
Employee													
			<table border="1"><thead><tr><th>EntityType</th><th>DepartmentName</th><th>EmployeeCount</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr></tbody></table>	EntityType	DepartmentName	EmployeeCount							
EntityType	DepartmentName	EmployeeCount											

	Department
EntityType FirstName LastName Age Email	
	Employee

The first option, prepending the entity type to the **RowKey**, is useful if there is a possibility that two entities of different types might have the same key value. It also groups entities of the same type together in the partition.

The techniques discussed in this section are especially relevant to the discussion [Inheritance relationships](#) earlier in this guide in the article [Modeling relationships](#).

 **Note**

You should consider including a version number in the entity type value to enable client applications to evolve POCO objects and work with different versions.

The remainder of this section describes some of the features in the Storage client library that facilitate working with multiple entity types in the same table.

Retrieving heterogeneous entity types

If you are using the Table client library, you have three options for working with multiple entity types.

If you know the type of the entity stored with a specific **RowKey** and **PartitionKey** values, then you can specify the entity type when you retrieve the entity as shown in the previous two examples that retrieve entities of type **EmployeeEntity**: [Executing a point query using the Storage client library](#) and [Retrieving multiple entities using LINQ](#).

The second option is to use the **TableEntity** type (a property bag) instead of a concrete POCO entity type (this option may also improve performance because there is no need to serialize and deserialize the entity to .NET types). The following C# code potentially retrieves multiple entities of different types from the table, but returns all entities as **TableEntity** instances. It then uses the **EntityType** property to determine the type of each entity:

C#

```
Pageable<TableEntity> entities = employeeTable.Query<TableEntity>(x =>
    x.PartitionKey == "Sales" && x.RowKey.CompareTo("B") >= 0 &&
```

```

x.RowKey.CompareTo("F") <= 0)

foreach (var entity in entities)
{
    if (entity.GetString("EntityType") == "Employee")
    {
        // use entityTypeProperty, RowKey, PartitionKey, Etag, and Timestamp
    }
}

```

To retrieve other properties you must use the `GetString` method on the `entity` of the `TableEntity` class.

Modifying heterogeneous entity types

You do not need to know the type of an entity to delete it, and you always know the type of an entity when you insert it. However, you can use `TableEntity` type to update an entity without knowing its type and without using a POCO entity class. The following code sample retrieves a single entity, and checks the `EmployeeCount` property exists before updating it.

C#

```

var result = employeeTable.GetEntity<TableEntity>(partitionKey, rowKey);
TableEntity department = result.Value;
if (department.GetInt32("EmployeeCount") == null)
{
    throw new InvalidOperationException("Invalid entity, EmployeeCount
property not found.");
}
employeeTable.UpdateEntity(department, ETag.All, TableUpdateMode.Merge);

```

Controlling access with Shared Access Signatures

You can use Shared Access Signature (SAS) tokens to enable client applications to modify (and query) table entities without the need to include your storage account key in your code. Typically, there are three main benefits to using SAS in your application:

- You do not need to distribute your storage account key to an insecure platform (such as a mobile device) in order to allow that device to access and modify entities in the Table service.
- You can offload some of the work that web and worker roles perform in managing your entities to client devices such as end-user computers and mobile devices.

- You can assign a constrained and time limited set of permissions to a client (such as allowing read-only access to specific resources).

For more information about using SAS tokens with the Table service, see [Using Shared Access Signatures \(SAS\)](#).

However, you must still generate the SAS tokens that grant a client application to the entities in the table service: you should do this in an environment that has secure access to your storage account keys. Typically, you use a web or worker role to generate the SAS tokens and deliver them to the client applications that need access to your entities. Because there is still an overhead involved in generating and delivering SAS tokens to clients, you should consider how best to reduce this overhead, especially in high-volume scenarios.

It is possible to generate a SAS token that grants access to a subset of the entities in a table. By default, you create a SAS token for an entire table, but it is also possible to specify that the SAS token grant access to either a range of **PartitionKey** values, or a range of **PartitionKey** and **RowKey** values. You might choose to generate SAS tokens for individual users of your system such that each user's SAS token only allows them access to their own entities in the table service.

Asynchronous and parallel operations

Provided you are spreading your requests across multiple partitions, you can improve throughput and client responsiveness by using asynchronous or parallel queries. For example, you might have two or more worker role instances accessing your tables in parallel. You could have individual worker roles responsible for particular sets of partitions, or simply have multiple worker role instances, each able to access all the partitions in a table.

Within a client instance, you can improve throughput by executing storage operations asynchronously. The Storage client library makes it easy to write asynchronous queries and modifications. For example, you might start with the synchronous method that retrieves all the entities in a partition as shown in the following C# code:

C#

```
private static void ManyEntitiesQuery(TableClient employeeTable, string
department)
{
    TableContinuationToken continuationToken = null;
    do
    {
        var employees = employeeTable.Query<EmployeeEntity>($"PartitionKey
$eq {department} &amp; RowKey ge {continuationToken}");
```

```
eq {department}");  
    foreach (var emp in employees.AsPages())  
    {  
        // ...  
        continuationToken = emp.ContinuationToken;  
    }  
  
} while (continuationToken != null);  
}
```

You can easily modify this code so that the query runs asynchronously as follows:

C#

```
private static async Task ManyEntitiesQueryAsync(TableClient employeeTable,  
string department)  
{  
    TableContinuationToken continuationToken = null;  
    do  
    {  
        var employees = await employeeTable.QueryAsync<EmployeeEntity>(  
($"PartitionKey eq {department}");  
        foreach (var emp in employees.AsPages())  
        {  
            // ...  
            continuationToken = emp.ContinuationToken;  
        }  
  
    } while (continuationToken != null);  
}
```

In this asynchronous example, you can see the following changes from the synchronous version:

- The method signature now includes the **async** modifier and returns a **Task** instance.
- Instead of calling the **Query** method to retrieve results, the method now calls the **QueryAsync** method and uses the **await** modifier to retrieve results asynchronously.

The client application can call this method multiple times (with different values for the **department** parameter), and each query will run on a separate thread.

You can also insert, update, and delete entities asynchronously. The following C# example shows a simple, synchronous method to insert or replace an employee entity:

C#

```
private static void SimpleEmployeeUpsert(
    TableClient employeeTable,
    EmployeeEntity employee)
{
    var result = employeeTable.UpdateEntity(employee, Azure.ETag.All,
TableUpdateMode.Replace);
    Console.WriteLine("HTTP Status: {0}", result.Status);
}
```

You can easily modify this code so that the update runs asynchronously as follows:

C#

```
private static async Task SimpleEmployeeUpsertAsync(
    TableClient employeeTable,
    EmployeeEntity employee)
{
    var result = await employeeTable.UpdateEntityAsync(employee,
Azure.ETag.All, TableUpdateMode.Replace);
    Console.WriteLine("HTTP Status: {0}", result.Result.Status);
}
```

In this asynchronous example, you can see the following changes from the synchronous version:

- The method signature now includes the **async** modifier and returns a **Task** instance.
- Instead of calling the **Execute** method to update the entity, the method now calls the **ExecuteAsync** method and uses the **await** modifier to retrieve results asynchronously.

The client application can call multiple asynchronous methods like this one, and each method invocation will run on a separate thread.

Next steps

- [Modeling relationships](#)
- [Design for querying](#)
- [Encrypting table data](#)
- [Design for data modification](#)

Azure managed disk types

Article • 01/10/2024

Applies to: ✓ Linux VMs ✓ Windows VMs ✓ Flexible scale sets ✓ Uniform scale sets

Azure managed disks currently offers five disk types, each intended to address a specific customer scenario:

- Ultra disks
- Premium SSD v2
- Premium SSDs (solid-state drives)
- Standard SSDs
- Standard HDDs (hard disk drives)

Disk type comparison

The following table provides a comparison of the five disk types to help you decide which to use.

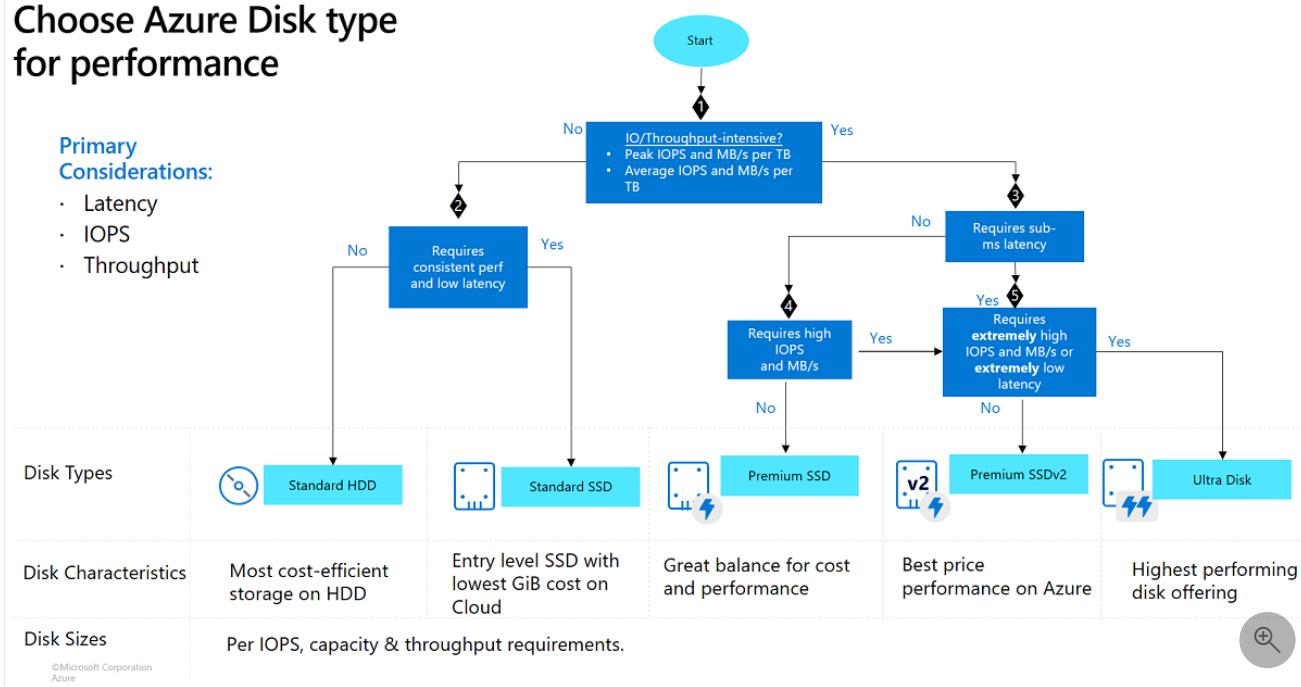
 Expand table

	Ultra disk	Premium SSD v2	Premium SSD	Standard SSD	Standard HDD
Disk type	SSD	SSD	SSD	SSD	HDD
Scenario	IO-intensive workloads such as SAP HANA , top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads.	Production and performance-sensitive workloads that consistently require low latency and high IOPS and throughput	Production and performance sensitive workloads	Web servers, lightly used enterprise applications and dev/test	Backup, non-critical, infrequent access
Max disk size	65,536 GiB	65,536 GiB	32,767 GiB	32,767 GiB	32,767 GiB
Max throughput	4,000 MB/s	1,200 MB/s	900 MB/s	750 MB/s	500 MB/s
Max IOPS	160,000	80,000	20,000	6,000	2,000, 3,000*
Usable as OS Disk?	No	No	Yes	Yes	Yes

* Only applies to disks with performance plus (preview) enabled.

For more help deciding which disk type suits your needs, this decision tree should help with typical scenarios:

Choose Azure Disk type for performance



For a video that covers some high level differences for the different disk types, as well as some ways for determining what impacts your workload requirements, see [Block storage options with Azure Disk Storage and Elastic SAN](#).

Ultra disks

Azure ultra disks are the highest-performing storage option for Azure virtual machines (VMs). You can change the performance parameters of an ultra disk without having to restart your VMs. Ultra disks are suited for data-intensive workloads such as SAP HANA, top-tier databases, and transaction-heavy workloads.

Ultra disks must be used as data disks and can only be created as empty disks. You should use Premium solid-state drives (SSDs) as operating system (OS) disks.

Ultra disk size

Azure ultra disks offer up to 32-TiB per region per subscription by default, but ultra disks support higher capacity by request. To request an increase in capacity, request a quota increase or contact Azure Support.

The following table provides a comparison of disk sizes and performance caps to help you decide which to use.

[Expand table](#)

Disk Size (GiB)	IOPS Cap	Throughput Cap (MB/s)
4	1,200	300
8	2,400	600
16	4,800	1,200
32	9,600	2,400
64	19,200	4,000
128	38,400	4,000
256	76,800	4,000
512	153,600	4,000
1,024-65,536 (sizes in this range increasing in increments of 1 TiB)	160,000	4,000

Ultra disk performance

Ultra disks are designed to provide low sub millisecond latencies and provisioned IOPS and throughput 99.99% of the time. Ultra disks also feature a flexible performance configuration model that allows you to independently configure IOPS and throughput, before and after you provision the disk. Ultra disks come in several fixed sizes, ranging from 4 GiB up to 64 TiB.

Ultra disk IOPS

Ultra disks support IOPS limits of 300 IOPS/GiB, up to a maximum of 160,000 IOPS per disk. To achieve the target IOPS for the disk, ensure that the selected disk IOPS are less than the VM IOPS limit.

The current maximum limit for IOPS for a single VM in generally available sizes is 80,000. Ultra disks with greater IOPS can be used as shared disks to support multiple VMs.

The minimum guaranteed IOPS per disk are 1 IOPS/GiB, with an overall baseline minimum of 100 IOPS. For example, if you provisioned a 4-GiB ultra disk, the minimum IOPS for that disk is 100, instead of four.

For more information about IOPS, see [Virtual machine and disk performance](#).

Ultra disk throughput

The throughput limit of a single ultra disk is 256-kB/s for each provisioned IOPS, up to a maximum of 4000 MB/s per disk (where MB/s = 10^6 Bytes per second). The minimum guaranteed throughput per disk is 4kB/s for each provisioned IOPS, with an overall baseline minimum of 1 MB/s.

You can adjust ultra disk IOPS and throughput performance at runtime without detaching the disk from the virtual machine. After a performance resize operation has been issued on a disk, it can take up to an hour for the change to take effect. Up to four performance resize operations are permitted during a 24-hour window.

It's possible for a performance resize operation to fail because of a lack of performance bandwidth capacity.

Ultra disk limitations

Ultra disks can't be used as OS disks, they can only be created as empty data disks. Ultra disks also can't be used with some features and functionality, including disk export, changing disk type, VM images, availability sets, or Azure disk encryption. The size of an Ultra Disk can't be expanded without either deallocating the VM or detaching the disk. Azure Site Recovery doesn't support ultra disks. In addition, only un-cached reads and un-cached writes are supported. Snapshots for ultra disks are available but have additional limitations. See [Incremental snapshots of Premium SSD v2 and Ultra Disks](#) for details. Azure Backup support for VMs with Ultra Disks is currently in [public preview](#).

Ultra disks support a 4k physical sector size by default. A 512E sector size is available as a generally available offering with no sign-up required. While most applications are compatible with 4k sector sizes, some require 512 byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks. For older versions of Oracle DB, 512 byte sector size is required.

The only infrastructure redundancy options currently available to ultra disks are availability zones. VMs using any other redundancy options cannot attach an ultra disk.

The following table outlines the regions ultra disks are available in, as well as their corresponding availability options.

Note

If a region in the following list lacks availability zones that support ultra disks, then a VM in that region must be deployed without infrastructure redundancy in order to attach an ultra disk.

 Expand table

Redundancy options	Regions
Single VMs	Australia Central Brazil South

Redundancy options	Regions
	Central India East Asia Germany West Central Korea Central Korea South UK West North Central US, South Central US, West US US Gov Arizona, US Gov Texas, US Gov Virginia
One availability zone	Brazil Southeast Poland Central UAE North
Two availability zones	South Africa North China North 3 France Central Qatar Central Switzerland North
Three availability zones	Australia East Canada Central North Europe, West Europe Japan East Southeast Asia Sweden Central UK South Central US, East US, East US 2, West US 2, West US 3

Not every VM size is available in every supported region with ultra disks. The following table lists VM series which are compatible with ultra disks.

[Expand table](#)

VM Type	Sizes	Description
General purpose	DSv3-series , Ddsv4-series , Dsv4-series , Dasv4-series , Dsv5-series , Ddsv5-series , Dasv5-series	Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.
Compute optimized	FSv2-series	High CPU-to-memory ratio. Good for medium traffic web servers, network appliances, batch processes, and application servers.
Memory optimized	ESv3-series , Easv4-series , Edsv4-series , Esv4-series , Esv5-series , Edsv5-series , Easv5-series , Ebsv5 series , Ebdsv5 series , M-series , Mv2-series , Msv2/Mdsv2-series	High memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage optimized	LSv2-series , Lsv3-series , Lasv3-series	High disk throughput and IO ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases.
GPU optimized	NCv2-series , NCv3-series , NCasT4_v3-series , ND-series , NDv2-series , NVv3-series , NVv4-series , NVadsA10 v5-series	Specialized virtual machines targeted for heavy graphic rendering and video editing, as well as model training and inferencing (ND) with deep learning. Available with single or multiple GPUs.
Performance optimized	HB-series , HC-series , HBv2-series	The fastest and most powerful CPU virtual machines with optional high-throughput network interfaces (RDMA).

If you would like to start using ultra disks, see the article on [using Azure Ultra Disks](#).

Premium SSD v2

Premium SSD v2 offers higher performance than Premium SSDs while also generally being less costly. You can individually tweak the performance (capacity, throughput, and IOPS) of Premium SSD v2 disks at any time, allowing workloads to be cost efficient while meeting shifting performance needs. For example, a transaction-intensive database may need a large amount of IOPS at a small size, or a gaming application may need a large amount of IOPS but only during peak hours. Because of this, for most general purpose workloads, Premium SSD v2 can provide the best price performance.

Premium SSD v2 is suited for a broad range of workloads such as SQL server, Oracle, MariaDB, SAP, Cassandra, Mongo DB, big data/analytics, and gaming, on virtual machines or stateful containers.

Premium SSD v2 support a 4k physical sector size by default, but can be configured to use a 512E sector size as well. While most applications are compatible with 4k sector sizes, some require 512 byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks.

Differences between Premium SSD and Premium SSD v2

Unlike Premium SSDs, Premium SSD v2 doesn't have dedicated sizes. You can set a Premium SSD v2 to any supported size you prefer, and make granular adjustments to the performance without downtime. Premium SSD v2 doesn't support host caching but, benefits significantly from lower latency, which addresses some of the same core problems host caching addresses. The ability to adjust IOPS, throughput, and size at any time also means you can avoid the maintenance overhead of having to stripe disks to meet your needs.

Premium SSD v2 limitations

- Premium SSD v2 disks can't be used as an OS disk.
- Currently, Premium SSD v2 disks can only be attached to zonal VMs.
- Encryption at host is supported on Premium SSD v2 disks with some limitations and in select regions. For more information, see [Encryption at host](#).
- Azure Disk Encryption (guest VM encryption via Bitlocker/DM-Crypt) isn't supported for VMs with Premium SSD v2 disks. We recommend you to use encryption at rest with platform-managed or customer-managed keys, which is supported for Premium SSD v2.
- Currently, Premium SSD v2 disks can't be attached to VMs in Availability Sets.
- Azure Site Recovery isn't supported for VMs with Premium SSD v2 disks.
- Azure Backup support for VMs with Premium SSD v2 disks is currently in [public preview](#).
- The size of a Premium SSD v2 can't be expanded without either deallocating the VM or detaching the disk.
- Premium SSDv2 does NOT support host caching.

Regional availability

Currently only available in the following regions:

- Australia East (Three availability zones)
- Brazil South (Two availability zones)
- Canada Central (Three availability zones)
- Central India (Three availability zones)
- Central US (One availability zone)
- China North 3 (Three availability zones)
- East Asia (Three availability zones)
- East US (Three availability zones)
- East US 2 (Three availability zones)
- France Central (Three availability zones)
- Germany West Central (Two availability zones)
- Israel Central (Two availability zones)
- Japan East (Three availability zones)
- Korea Central (Three availability zones)
- North Europe (Three availability zones)
- Norway East (Three availability zones)
- Poland Central (Three availability zones)
- South Africa North (Three availability zones)
- South Central US (Three availability zones)
- Southeast Asia (Two availability zones)
- Sweden Central (Three availability zones)
- Switzerland North (Three availability zones)
- UAE North (Three availability zones)
- UK South (Three availability zones)
- US Gov Virginia (Three availability zones)
- West Europe (Three availability zones)
- West US 2 (Three availability zones)

- West US 3 (Three availability zones)

To learn when support for particular regions was added, see either [Azure Updates](#) or [What's new for Azure Disk Storage](#).

Premium SSD v2 performance

Premium SSD v2 disks are designed to provide sub millisecond latencies and provisioned IOPS and throughput 99.9% of the time. With Premium SSD v2 disks, you can individually set the capacity, throughput, and IOPS of a disk based on your workload needs, providing you with more flexibility and reduced costs. Each of these values determines the cost of your disk.

Premium SSD v2 capacities

Premium SSD v2 capacities range from 1 GiB to 64 TiBs, in 1-GiB increments. You're billed on a per GiB ratio, see the [pricing page](#) for details.

Premium SSD v2 offers up to 32 TiBs per region per subscription by default, but supports higher capacity by request. To request an increase in capacity, request a quota increase or contact Azure Support.

Premium SSD v2 IOPS

All Premium SSD v2 disks have a baseline IOPS of 3000 that is free of charge. After 6 GiB, the maximum IOPS a disk can have increases at a rate of 500 per GiB, up to 80,000 IOPS. So an 8 GiB disk can have up to 4,000 IOPS, and a 10 GiB can have up to 5,000 IOPS. To be able to set 80,000 IOPS on a disk, that disk must have at least 160 GiBs. Increasing your IOPS beyond 3000 increases the price of your disk.

Premium SSD v2 throughput

All Premium SSD v2 disks have a baseline throughput of 125 MB/s that is free of charge. After 6 GiB, the maximum throughput that can be set increases by 0.25 MB/s per set IOPS. If a disk has 3,000 IOPS, the max throughput it can set is 750 MB/s. To raise the throughput for this disk beyond 750 MB/s, its IOPS must be increased. For example, if you increased the IOPS to 4,000, then the max throughput that can be set is 1,000. 1,200 MB/s is the maximum throughput supported for disks that have 5,000 IOPS or more. Increasing your throughput beyond 125 increases the price of your disk.

Premium SSD v2 Sector Sizes

Premium SSD v2 supports a 4k physical sector size by default. A 512E sector size is also supported. While most applications are compatible with 4k sector sizes, some require 512-byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks.

Summary

The following table provides a comparison of disk capacities and performance maximums to help you decide which to use.

[Expand table](#)

Disk Size	Maximum available IOPS	Maximum available throughput (MB/s)
1 GiB-64 TiBs	3,000-80,000 (Increases by 500 IOPS per GiB)	125-1,200 (increases by 0.25 MB/s per set IOPS)

To deploy a Premium SSD v2, see [Deploy a Premium SSD v2](#).

Premium SSDs

Azure Premium SSDs deliver high-performance and low-latency disk support for virtual machines (VMs) with input/output (IO)-intensive workloads. To take advantage of the speed and performance of Premium SSDs, you can migrate existing VM disks to Premium SSDs. Premium SSDs are suitable for mission-critical production applications, but you can use them only with compatible VM series. Premium SSDs support the [512E sector size](#).

To learn more about individual Azure VM types and sizes for Windows or Linux, including size compatibility for premium storage, see [Sizes for virtual machines in Azure](#). You'll need to check each individual VM size article to determine if it's premium storage-compatible.

Premium SSD size

[Expand table](#)

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base provisioned IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
**Expanded provisioned IOPS per disk	N/A	8,000	16,000	20,000	20,000	20,000	20,000							
Base provisioned throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	750 MB/s	900 MB/s
**Expanded provisioned throughput per disk	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	900 MB/s	900 MB/s							
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,000*	30,000*
Max burst throughput per disk	170 MB/s	1,000 MB/s*												
Max burst duration	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*							
Eligible for reservation	No	Yes, up to one year												

*Applies only to disks with on-demand bursting enabled.

** Only applies to disks with performance plus (preview) enabled.

Capacity, IOPS, and throughput are guaranteed when a premium storage disk is provisioned. For example, if you create a P50 disk, Azure provisions 4,095-GB storage capacity, 7,500 IOPS, and 250-MB/s throughput for that disk. Your application can use all or part of the capacity and performance. Premium SSDs are designed to provide the single-digit millisecond latencies, target IOPS, and throughput described in the preceding table 99.9% of the time.

Premium SSD bursting

Premium SSDs offer disk bursting, which provides better tolerance on unpredictable changes of IO patterns. Disk bursting is especially useful during OS disk boot and for applications with spiky traffic. To learn more about how bursting for Azure disks works, see [Disk-level bursting](#).

Premium SSD transactions

For Premium SSDs, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of throughput are considered multiple I/Os of size 256 kB.

Standard SSDs

Azure standard SSDs are optimized for workloads that need consistent performance at lower IOPS levels. They're an especially good choice for customers with varying workloads supported by on-premises hard disk drive (HDD) solutions. Compared to standard HDDs, standard SSDs deliver better availability, consistency, reliability, and latency. Standard SSDs are suitable for web servers, low IOPS application servers, lightly used enterprise applications, and non-production workloads. Like standard HDDs, standard SSDs are available on all Azure VMs. Standard SSDs support the [512E sector size](#).

Standard SSD size

[Expand table](#)

Standard SSD sizes	E1	E2	E3	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to 500	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000									
*Expanded IOPS per disk	N/A	Up to 1,500	Up to 3,000	Up to 6,000	Up to 6,000	Up to 6,000	Up to 6,000							
Base throughput per disk	Up to 60 MB/s	Up to 60 MB/s	Up to 400 MB/s	Up to 600 MB/s	Up to 750 MB/s									
*Expanded throughput per disk	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 600 MB/s	Up to 750 MB/s	Up to 750 MB/s	Up to 750 MB/s							
Max burst IOPS per disk	600	600	600	600	600	600	600	600	1000					
Max burst throughput per disk	150 MB/s	250 MB/s												
Max burst duration	30 min	30 min												

* Only applies to disks with performance plus (preview) enabled.

Standard SSDs are designed to provide single-digit millisecond latencies and the IOPS and throughput up to the limits described in the preceding table 99% of the time. Actual IOPS and throughput may vary sometimes depending on the traffic patterns. Standard SSDs provide more consistent performance than the HDD disks with the lower latency.

Standard SSD transactions

For standard SSDs, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of throughput are considered multiple I/Os of size 256 kB. These transactions incur a billable cost but, there's an hourly limit on the number of transactions that can incur a billable cost. If that hourly limit is reached, additional transactions during that hour no longer incur a cost. For details, see the [blog post](#).

Standard SSD Bursting

Standard SSDs offer disk bursting, which provides better tolerance for the unpredictable IO pattern changes. OS boot disks and applications prone to traffic spikes will both benefit from disk bursting. To learn more about how bursting for Azure disks works, see [Disk-level bursting](#).

Standard HDDs

Azure standard HDDs deliver reliable, low-cost disk support for VMs running latency-tolerant workloads. With standard storage, your data is stored on HDDs, and performance may vary more widely than that of SSD-based disks. Standard HDDs are designed to deliver write latencies of less than 10 ms and read latencies of less than 20 ms for most IO operations. Actual performance may vary depending on IO size and workload pattern, however. When working with VMs, you can use standard HDD disks for dev/test scenarios and less critical workloads. Standard HDDs are available in all Azure regions and can be used with all Azure VMs. Standard HDDs support the [512E sector size](#).

Standard HDD size

 [Expand table](#)

Standard Disk Type	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to 500	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000	Up to 2,000					
*Expanded IOPS per disk	N/A	N/A	N/A	N/A	N/A	Up to 1,500	Up to 3,000				
Base throughput per disk	Up to 60 MB/s	Up to 60 MB/s	Up to 300 MB/s	Up to 500 MB/s	Up to 500 MB/s	Up to 500 MB/s					
*Expanded throughput per disk	N/A	N/A	N/A	N/A	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 500 MB/s			

* Only applies to disks with performance plus (preview) enabled.

Standard HDD Transactions

For Standard HDDs, each I/O operation is considered as a single transaction, whatever the I/O size. These transactions have a billing impact.

Billing

When using managed disks, the following billing considerations apply:

- Disk type
- Managed disk Size
- Snapshots
- Outbound data transfers
- Number of transactions

Managed disk size: Managed disks are billed according to their provisioned size. Azure maps the provisioned size (rounded up) to the nearest offered disk size. For details of the disk sizes offered, see the previous tables. Each disk maps to a supported provisioned disk-size offering and is billed accordingly. For example, if you provisioned a 200-GiB standard SSD, it maps to the disk size offer of E15 (256 GiB). Billing for any provisioned disk is prorated hourly by using the monthly price for the storage offering. For example, you provision an E10 disk and delete it after 20 hours of use. In this case, you're billed for the E10 offering prorated to 20 hours, regardless of the amount of data written to the disk.

Snapshots: Snapshots are billed based on the size used. For example, you create a snapshot of a managed disk with provisioned capacity of 64 GiB and actual used data size of 10 GiB. In this case, the snapshot is billed only for the used data size of 10 GiB.

For more information on snapshots, see the section on snapshots in the [managed disk overview](#).

Outbound data transfers: [Outbound data transfers](#) (data going out of Azure data centers) incur billing for bandwidth usage.

Transactions: You're billed for the number of transactions performed on a standard managed disk. For standard SSDs, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of

throughput are considered multiple I/Os of size 256 kB. For Standard HDDs, each IO operation is considered a single transaction, whatever the I/O size.

For detailed information on pricing for managed disks (including transaction costs), see [Managed Disks Pricing](#).

Ultra disks VM reservation fee

Azure VMs have the capability to indicate if they're compatible with ultra disks. An ultra disk-compatible VM allocates dedicated bandwidth capacity between the compute VM instance and the block storage scale unit to optimize the performance and reduce latency. When you add this capability on the VM, it results in a reservation charge. The reservation charge is only imposed if you enabled ultra disk capability on the VM without an attached ultra disk. When an ultra disk is attached to the ultra disk compatible VM, the reservation charge wouldn't be applied. This charge is per vCPU provisioned on the VM.

 **Note**

For **constrained core VM sizes**, the reservation fee is based on the actual number of vCPUs and not the constrained cores. For Standard_E32-8s_v3, the reservation fee will be based on 32 cores.

Refer to the [Azure Disks pricing page](#) for ultra disk pricing details.

Azure disk reservation

Disk reservation provides you with a discount on the advance purchase of one year's of disk storage, reducing your total cost. When you purchase a disk reservation, you select a specific disk SKU in a target region. For example, you may choose five P30 (1 TiB) Premium SSDs in the Central US region for a one year term. The disk reservation experience is similar to Azure reserved VM instances. You can bundle VM and Disk reservations to maximize your savings. For now, Azure Disks Reservation offers one year commitment plan for Premium SSD SKUs from P30 (1 TiB) to P80 (32 TiB) in all production regions. For more information about reserved disks pricing, see [Azure Disks pricing page](#).

Next steps

See [Managed Disks pricing](#) to get started.

Server-side encryption of Azure Disk Storage

Article • 01/11/2024

Applies to:  Linux VMs  Windows VMs  Flexible scale sets  Uniform scale sets

Most Azure managed disks are encrypted with Azure Storage encryption, which uses server-side encryption (SSE) to protect your data and to help you meet your organizational security and compliance commitments. Azure Storage encryption automatically encrypts your data stored on Azure managed disks (OS and data disks) at rest by default when persisting it to the cloud. Disks with encryption at host enabled, however, aren't encrypted through Azure Storage. For disks with encryption at host enabled, the server hosting your VM provides the encryption for your data, and that encrypted data flows into Azure Storage.

Data in Azure managed disks is encrypted transparently using 256-bit [AES encryption](#), one of the strongest block ciphers available, and is FIPS 140-2 compliant. For more information about the cryptographic modules underlying Azure managed disks, see [Cryptography API: Next Generation](#)

Azure Storage encryption doesn't impact the performance of managed disks and there's no extra cost. For more information about Azure Storage encryption, see [Azure Storage encryption](#).

Note

Temporary disks are not managed disks and are not encrypted by SSE, unless you enable encryption at host.

About encryption key management

You can rely on platform-managed keys for the encryption of your managed disk, or you can manage encryption using your own keys. If you choose to manage encryption with your own keys, you can specify a *customer-managed key* to use for encrypting and decrypting all data in managed disks.

The following sections describe each of the options for key management in greater detail.

Platform-managed keys

By default, managed disks use platform-managed encryption keys. All managed disks, snapshots, images, and data written to existing managed disks are automatically encrypted-at-rest with platform-managed keys. Platform-managed keys are managed by Microsoft.

Customer-managed keys

You can choose to manage encryption at the level of each managed disk, with your own keys. When you specify a customer-managed key, that key is used to protect and control access to the key that encrypts your data. Customer-managed keys offer greater flexibility to manage access controls.

You must use one of the following Azure key stores to store your customer-managed keys:

- [Azure Key Vault](#)
- [Azure Key Vault Managed Hardware Security Module \(HSM\)](#)

You can either import [your RSA keys](#) to your Key Vault or generate new RSA keys in Azure Key Vault. Azure managed disks handles the encryption and decryption in a fully transparent fashion using envelope encryption. It encrypts data using an [AES](#) 256 based data encryption key (DEK), which is, in turn, protected using your keys. The Storage service generates data encryption keys and encrypts them with customer-managed keys using RSA encryption. The envelope encryption allows you to rotate (change) your keys periodically as per your compliance policies without impacting your VMs. When you rotate your keys, the Storage service re-encrypts the data encryption keys with the new customer-managed keys.

Managed disks and the Key Vault or managed HSM must be in the same Azure region, but they can be in different subscriptions. They must also be in the same Microsoft Entra tenant, unless you're using [Encrypt managed disks with cross-tenant customer-managed keys \(preview\)](#).

Full control of your keys

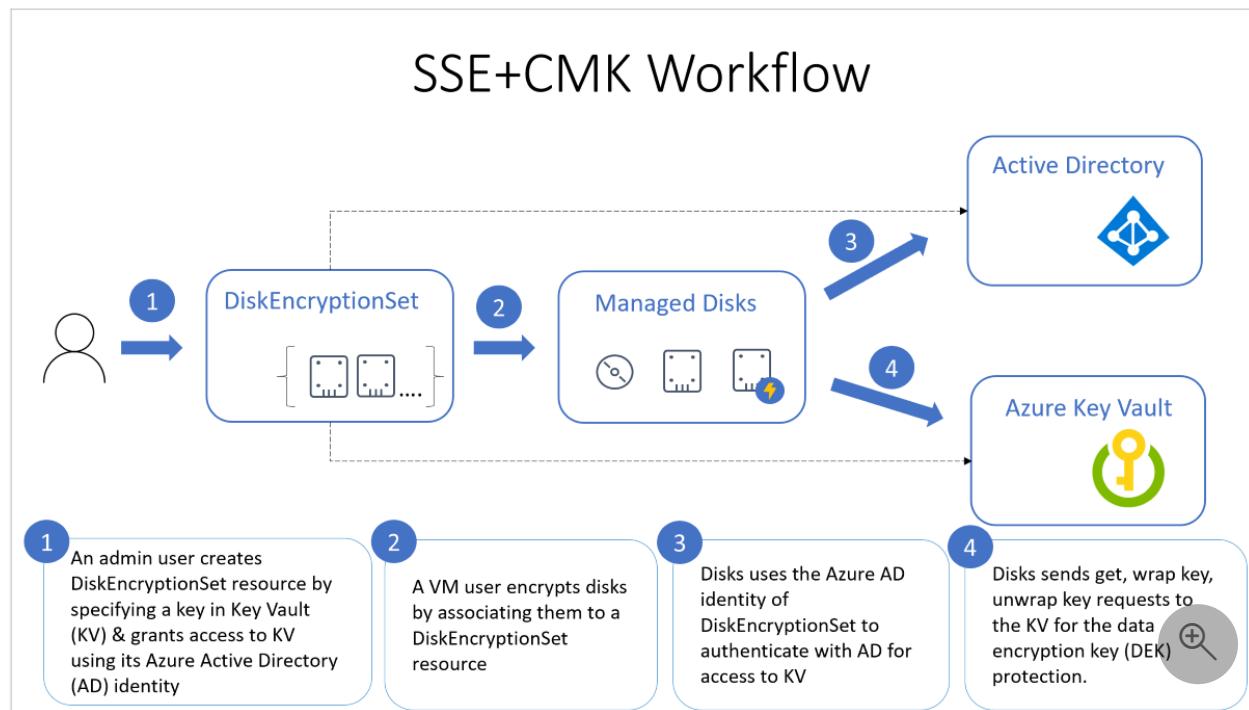
You must grant access to managed disks in your Key Vault or managed HSM to use your keys for encrypting and decrypting the DEK. This allows you full control of your data and keys. You can disable your keys or revoke access to managed disks at any time. You can also audit the encryption key usage with Azure Key Vault monitoring to ensure that only managed disks or other trusted Azure services are accessing your keys.

ⓘ Important

When a key is either disabled, deleted, or expired, any VMs with either OS or data disks using that key will automatically shut down. After the automated shut down, VMs won't boot until the key is enabled again, or you assign a new key.

Generally, disk I/O (read or write operations) start to fail one hour after a key is either disabled, deleted, or expired.

The following diagram shows how managed disks use Microsoft Entra ID and Azure Key Vault to make requests using the customer-managed key:



The following list explains the diagram in more detail:

1. An Azure Key Vault administrator creates key vault resources.
2. The key vault admin either imports their RSA keys to Key Vault or generate new RSA keys in Key Vault.
3. That administrator creates an instance of Disk Encryption Set resource, specifying an Azure Key Vault ID and a key URL. Disk Encryption Set is a new resource introduced for simplifying the key management for managed disks.
4. When a disk encryption set is created, a [system-assigned managed identity](#) is created in Microsoft Entra ID and associated with the disk encryption set.
5. The Azure key vault administrator then grants the managed identity permission to perform operations in the key vault.
6. A VM user creates disks by associating them with the disk encryption set. The VM user can also enable server-side encryption with customer-managed keys for

existing resources by associating them with the disk encryption set.

7. Managed disks use the managed identity to send requests to the Azure Key Vault.
8. For reading or writing data, managed disks sends requests to Azure Key Vault to encrypt (wrap) and decrypt (unwrap) the data encryption key in order to perform encryption and decryption of the data.

To revoke access to customer-managed keys, see [Azure Key Vault PowerShell](#) and [Azure Key Vault CLI](#). Revoking access effectively blocks access to all data in the storage account, as the encryption key is inaccessible by Azure Storage.

Automatic key rotation of customer-managed keys

Generally, if you're using customer-managed keys, you should enable automatic key rotation to the latest key version. Automatic key rotation helps ensure your keys are secure. A disk references a key via its disk encryption set. When you enable automatic rotation for a disk encryption set, the system will automatically update all managed disks, snapshots, and images referencing the disk encryption set to use the new version of the key within one hour. To learn how to enable customer-managed keys with automatic key rotation, see [Set up an Azure Key Vault and DiskEncryptionSet with automatic key rotation](#).

Note

Virtual Machines aren't rebooted during automatic key rotation.

If you can't enable automatic key rotation, you can use other methods to alert you before keys expire. This way, you can make sure to rotate your keys before expiration and keep business continuity. You can use either an [Azure Policy](#) or [Azure Event Grid](#) to send a notification when a key expires soon.

Restrictions

For now, customer-managed keys have the following restrictions:

- If this feature is enabled for a disk with incremental snapshots, it can't be disabled on that disk or its snapshots. To work around this, copy all the data to an entirely different managed disk that isn't using customer-managed keys. You can do that with either the [Azure CLI](#) or the [Azure PowerShell module](#).
- Only [software and HSM RSA keys](#) of sizes 2,048-bit, 3,072-bit and 4,096-bit are supported, no other keys or sizes.
 - [HSM](#) keys require the [premium](#) tier of Azure Key vaults.

- For Ultra Disks and Premium SSD v2 disks only: Snapshots created from disks that are encrypted with server-side encryption and customer-managed keys must be encrypted with the same customer-managed keys.
- Most resources related to your customer-managed keys (disk encryption sets, VMs, disks, and snapshots) must be in the same subscription and region.
 - Azure Key Vaults may be used from a different subscription but must be in the same region as your disk encryption set. As a preview, you can use Azure Key Vaults from [different Microsoft Entra tenants](#).
- Disks encrypted with customer-managed keys can only move to another resource group if the VM they are attached to is deallocated.
- Disks, snapshots, and images encrypted with customer-managed keys can't be moved between subscriptions.
- Managed disks currently or previously encrypted using Azure Disk Encryption can't be encrypted using customer-managed keys.
- Can only create up to 5000 disk encryption sets per region per subscription.
- For information about using customer-managed keys with shared image galleries, see [Preview: Use customer-managed keys for encrypting images](#).

Supported regions

Customer-managed keys are available in all regions that managed disks are available.

Important

Customer-managed keys rely on managed identities for Azure resources, a feature of Microsoft Entra ID. When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the subscription, resource group, or managed disk from one Microsoft Entra directory to another, the managed identity associated with managed disks isn't transferred to the new tenant, so customer-managed keys may no longer work. For more information, see [Transferring a subscription between Microsoft Entra directories](#).

To enable customer-managed keys for managed disks, see our articles covering how to enable it with either the [Azure PowerShell module](#), the [Azure CLI](#) or the [Azure portal](#).

See [Create a managed disk from a snapshot with CLI](#) for a code sample.

Encryption at host - End-to-end encryption for your VM data

When you enable encryption at host, that encryption starts on the VM host itself, the Azure server that your VM is allocated to. The data for your temporary disk and OS/data disk caches are stored on that VM host. After enabling encryption at host, all this data is encrypted at rest and flows encrypted to the Storage service, where it's persisted. Essentially, encryption at host encrypts your data from end-to-end. Encryption at host doesn't use your VM's CPU and doesn't impact your VM's performance.

Temporary disks and ephemeral OS disks are encrypted at rest with platform-managed keys when you enable end-to-end encryption. The OS and data disk caches are encrypted at rest with either customer-managed or platform-managed keys, depending on the selected disk encryption type. For example, if a disk is encrypted with customer-managed keys, then the cache for the disk is encrypted with customer-managed keys, and if a disk is encrypted with platform-managed keys then the cache for the disk is encrypted with platform-managed keys.

Restrictions

- Supported for 4k sector size Ultra Disks and Premium SSD v2.
- Only supported on 512e sector size Ultra Disks and Premium SSD v2 if they were created after 5/13/2023.
 - For disks created before this date, [snapshot your disk](#) and create a new disk using the snapshot.
- Can't be enabled if Azure Disk Encryption (guest-VM encryption using bitlocker/DM-Crypt) is enabled on your virtual machines (VMs) or virtual machine scale sets.
- Azure Disk Encryption can't be enabled on disks that have encryption at host enabled.
- The encryption can be enabled on existing virtual machine scale sets. However, only new VMs created after enabling the encryption are automatically encrypted.
- Existing VMs must be deallocated and reallocated in order to be encrypted.

Regional availability

Except for Ultra Disks and Premium SSD v2 managed disks, encryption at host is available in all regions.

For Ultra Disks and Premium SSD v2 managed disks, encryption at host is currently available in every region except for the following:

- Canada East
- West Europe
- Japan West
- South Central US
- West US 3

Supported VM sizes

The complete list of supported VM sizes can be pulled programmatically. To learn how to retrieve them programmatically, refer to the finding supported VM sizes section of either the [Azure PowerShell module](#) or [Azure CLI](#) articles.

To enable end-to-end encryption using encryption at host, see our articles covering how to enable it with either the [Azure PowerShell module](#), the [Azure CLI](#), or the [Azure portal](#).

Double encryption at rest

High security sensitive customers who are concerned of the risk associated with any particular encryption algorithm, implementation, or key being compromised can now opt for extra layer of encryption using a different encryption algorithm/mode at the infrastructure layer using platform managed encryption keys. This new layer can be applied to persisted OS and data disks, snapshots, and images, all of which will be encrypted at rest with double encryption.

Restrictions

Double encryption at rest isn't currently supported with either Ultra Disks or Premium SSD v2 disks.

Supported regions

Double encryption is available in all regions that managed disks are available.

To enable double encryption at rest for managed disks, see our articles covering how to enable it with either the [Azure PowerShell module](#), the [Azure CLI](#) or the [Azure portal](#).

Server-side encryption versus Azure disk encryption

Azure Disk Encryption leverages either the [DM-Crypt](#) feature of Linux or the [BitLocker](#) feature of Windows to encrypt managed disks with customer-managed keys within the guest VM. Server-side encryption with customer-managed keys improves on ADE by enabling you to use any OS types and images for your VMs by encrypting data in the Storage service.

Important

Customer-managed keys rely on managed identities for Azure resources, a feature of Microsoft Entra ID. When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the subscription, resource group, or managed disk from one Microsoft Entra directory to another, the managed identity associated with managed disks is not transferred to the new tenant, so customer-managed keys may no longer work. For more information, see [Transferring a subscription between Microsoft Entra directories](#).

Next steps

- Enable end-to-end encryption using encryption at host with either the [Azure PowerShell module](#), the [Azure CLI](#), or the [Azure portal](#).
- Enable double encryption at rest for managed disks with either the [Azure PowerShell module](#), the [Azure CLI](#) or the [Azure portal](#).
- Enable customer-managed keys for managed disks with either the [Azure PowerShell module](#), the [Azure CLI](#) or the [Azure portal](#).
- Explore the [Azure Resource Manager templates](#) for creating encrypted disks with customer-managed keys
- [What is Azure Key Vault?](#)

Azure Disk Encryption for Linux VMs

Article • 08/03/2023

Applies to:  Linux VMs  Flexible scale sets

Azure Disk Encryption helps protect and safeguard your data to meet your organizational security and compliance commitments. It uses the [DM-Crypt](#) feature of Linux to provide volume encryption for the OS and data disks of Azure virtual machines (VMs), and is integrated with [Azure Key Vault](#) to help you control and manage the disk encryption keys and secrets.

Azure Disk Encryption is zone resilient, the same way as Virtual Machines. For details, see [Azure Services that support Availability Zones](#).

If you use [Microsoft Defender for Cloud](#), you're alerted if you have VMs that aren't encrypted. The alerts show as High Severity and the recommendation is to encrypt these VMs.

VIRTUAL MACHINES RECOMMENDATIONS		TOTAL																			
Missing disk encryption		2 of 2 VMs																			
Virtual machines																					
<table><thead><tr><th>NAME</th><th>ONBOARDING</th><th>SYSTEM UPDATES</th><th>ANTIMALWARE</th><th>BASELINE</th><th>DISK ENCRYPTION</th></tr></thead><tbody><tr><td> ASC-VM1</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td> ASC-VM2</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>				NAME	ONBOARDING	SYSTEM UPDATES	ANTIMALWARE	BASELINE	DISK ENCRYPTION	 ASC-VM1						 ASC-VM2					
NAME	ONBOARDING	SYSTEM UPDATES	ANTIMALWARE	BASELINE	DISK ENCRYPTION																
 ASC-VM1																					
 ASC-VM2																					

Warning

- If you have previously used Azure Disk Encryption with Azure AD to encrypt a VM, you must continue to use this option to encrypt your VM. See [Azure Disk Encryption with Azure AD \(previous release\)](#) for details.
- Certain recommendations might increase data, network, or compute resource usage, resulting in additional license or subscription costs. You must have a valid active Azure subscription to create resources in Azure in the supported regions.

You can learn the fundamentals of Azure Disk Encryption for Linux in just a few minutes with the [Create and encrypt a Linux VM with Azure CLI quickstart](#) or the [Create and](#)

encrypt a Linux VM with Azure PowerShell quickstart.

Supported VMs and operating systems

Supported VMs

Linux VMs are available in a [range of sizes](#). Azure Disk Encryption is supported on Generation 1 and Generation 2 VMs. Azure Disk Encryption is also available for VMs with premium storage.

See [Azure VM sizes with no local temporary disk](#).

Azure Disk Encryption is also not available on [Basic, A-series VMs](#), or on virtual machines that do not meet these minimum memory requirements:

Memory requirements

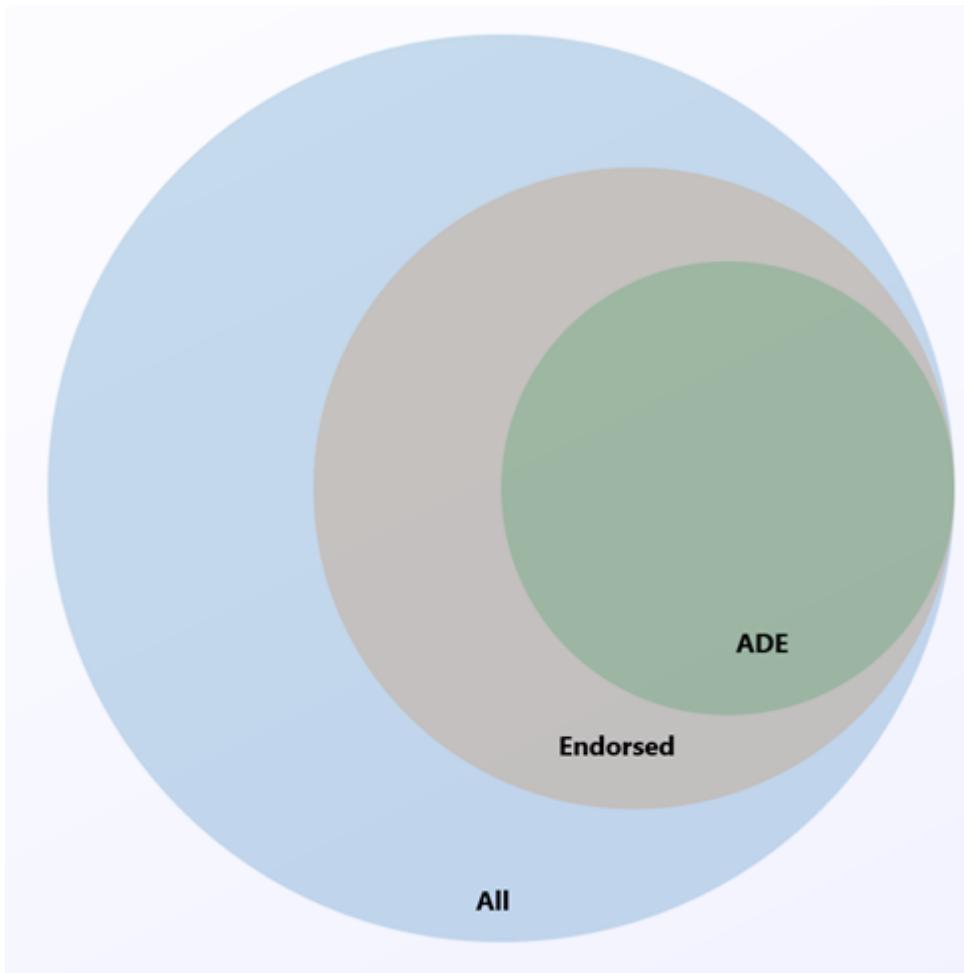
Virtual machine	Minimum memory requirement
Linux VMs when only encrypting data volumes	2 GB
Linux VMs when encrypting both data and OS volumes, and where the root (/) file system usage is 4GB or less	8 GB
Linux VMs when encrypting both data and OS volumes, and where the root (/) file system usage is greater than 4GB	The root file system usage * 2. For instance, a 16 GB of root file system usage requires at least 32GB of RAM

Once the OS disk encryption process is complete on Linux virtual machines, the VM can be configured to run with less memory.

For more exceptions, see [Azure Disk Encryption: Unsupported scenarios](#).

Supported operating systems

Azure Disk Encryption is supported on a subset of the [Azure-endorsed Linux distributions](#), which is itself a subset of all Linux server possible distributions.



Linux server distributions that are not endorsed by Azure do not support Azure Disk Encryption; of those that are endorsed, only the following distributions and versions support Azure Disk Encryption:

Publisher	Offer	SKU	URN	Volume type supported for encryption
Canonical	Ubuntu	22.04-LTS	Canonical:0001-com-ubuntu-server-focal:22_04-lts:latest	OS and data disk
Canonical	Ubuntu	22.04-LTS Gen2	Canonical:0001-com-ubuntu-server-focal:22_04-lts-gen2:latest	OS and data disk
Canonical	Ubuntu	20.04-LTS	Canonical:0001-com-ubuntu-server-focal:20_04-lts:latest	OS and data disk
Canonical	Ubuntu	20.04-DAILY-LTS	Canonical:0001-com-ubuntu-server-focal-daily:20_04-daily-lts:latest	OS and data disk
Canonical	Ubuntu	20.04-LTS Gen2	Canonical:0001-com-ubuntu-server-focal:20_04-lts-gen2:latest	OS and data disk

Publisher	Offer	SKU	URN	Volume type supported for encryption
Canonical	Ubuntu	20.04-DAILY-LTS Gen2	Canonical:0001-com-ubuntu-server-focal-daily:20_04-daily-lts-gen2:latest	OS and data disk
Canonical	Ubuntu	18.04-LTS	Canonical:UbuntuServer:18.04-LTS:latest	OS and data disk
Canonical	Ubuntu 18.04	18.04-DAILY-LTS	Canonical:UbuntuServer:18.04-DAILY-LTS:latest	OS and data disk
MicrosoftCBLMariner	cbl-mariner	cbl-mariner-2	MicrosoftCBLMariner:cbl-mariner:cbl-mariner-2:latest*	OS and data disk
MicrosoftCBLMariner	cbl-mariner	cbl-mariner-2-gen2	MicrosoftCBLMariner:cbl-mariner:cbl-mariner-2-gen2:latest*	OS and data disk
OpenLogic	CentOS 8-LVM	8-LVM	OpenLogic:CentOS-LVM:8-LVM:latest	OS and data disk
OpenLogic	CentOS 8.4	8_4	OpenLogic:CentOS:8_4:latest	OS and data disk
OpenLogic	CentOS 8.3	8_3	OpenLogic:CentOS:8_3:latest	OS and data disk
OpenLogic	CentOS 8.2	8_2	OpenLogic:CentOS:8_2:latest	OS and data disk
OpenLogic	CentOS 8.1	8_1	OpenLogic:CentOS:8_1:latest	OS and data disk
OpenLogic	CentOS 7-LVM	7-LVM	OpenLogic:CentOS-LVM:7-LVM:7.9.2021020400	OS and data disk
OpenLogic	CentOS 7.9	7_9	OpenLogic:CentOS:7_9:latest	OS and data disk
OpenLogic	CentOS 7.8	7_8	OpenLogic:CentOS:7_8:latest	OS and data disk
OpenLogic	CentOS 7.7	7.7	OpenLogic:CentOS:7.7:latest	OS and data disk

Publisher	Offer	SKU	URN	Volume type supported for encryption
OpenLogic	CentOS 7.6	7.6	OpenLogic:CentOS:7.6:latest	OS and data disk
OpenLogic	CentOS 7.5	7.5	OpenLogic:CentOS:7.5:latest	OS and data disk
OpenLogic	CentOS 7.4	7.4	OpenLogic:CentOS:7.4:latest	OS and data disk
OpenLogic	CentOS 6.8	6.8	OpenLogic:CentOS:6.8:latest	Data disk only
Oracle	Oracle Linux 8.6	8.6	Oracle:Oracle-Linux:ol86-lvm:latest	OS and data disk (see note below)
Oracle	Oracle Linux 8.6 Gen 2	8.6	Oracle:Oracle-Linux:ol86-lvm-gen2:latest	OS and data disk (see note below)
Oracle	Oracle Linux 8.5	8.5	Oracle:Oracle-Linux:ol85-lvm:latest	OS and data disk (see note below)
Oracle	Oracle Linux 8.5 Gen 2	8.5	Oracle:Oracle-Linux:ol85-lvm-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 9.2	9.2	RedHat:RHEL:9_2:latest	OS and data disk (see note below)
RedHat	RHEL 9.2 Gen 2	9.2	RedHat:RHEL:92-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 9.1	9.1	RedHat:RHEL:9_1:latest	OS and data disk (see note below)
RedHat	RHEL 9.1 Gen 2	9.1	RedHat:RHEL:91-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 9.0	9.0	RedHat:RHEL:9_0:latest	OS and data disk (see

Publisher	Offer	SKU	URN	Volume type supported for encryption
				note below)
RedHat	RHEL 9.0 Gen 2	9.0	RedHat:RHEL:90-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 9-lvm	9-lvm	RedHat:RHEL:9-lvm:latest	OS and data disk (see note below)
RedHat	RHEL 9-lvm Gen 2	9-lvm-gen2	RedHat:RHEL:9-lvm-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 8.8	8.8	RedHat:RHEL:8_8:latest	OS and data disk (see note below)
RedHat	RHEL 8.8 Gen 2	8.8	RedHat:RHEL:88-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 8.7	8.7	RedHat:RHEL:8_7:latest	OS and data disk (see note below)
RedHat	RHEL 8.7 Gen 2	8.7	RedHat:RHEL:87-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 8.6	8.6	RedHat:RHEL:8_6:latest	OS and data disk (see note below)
RedHat	RHEL 8.6 Gen 2	8.6	RedHat:RHEL:86-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 8.5	8.5	RedHat:RHEL:8_5:latest	OS and data disk (see note below)
RedHat	RHEL 8.5 Gen 2	8.5	RedHat:RHEL:85-gen2:latest	OS and data disk (see note below)

Publisher	Offer	SKU	URN	Volume type supported for encryption
RedHat	RHEL 8.4	8.4	RedHat:RHEL:8.4:latest	OS and data disk (see note below)
RedHat	RHEL 8.3	8.3	RedHat:RHEL:8.3:latest	OS and data disk (see note below)
RedHat	RHEL 8-LVM	8-LVM	RedHat:RHEL:8-LVM:latest	OS and data disk (see note below)
RedHat	RHEL 8-LVM Gen 2	8-lvm-gen2	RedHat:RHEL:8-lvm-gen2:latest	OS and data disk (see note below)
RedHat	RHEL 8.2	8.2	RedHat:RHEL:8.2:latest	OS and data disk (see note below)
RedHat	RHEL 8.1	8.1	RedHat:RHEL:8.1:latest	OS and data disk (see note below)
RedHat	RHEL 7-LVM	7-LVM	RedHat:RHEL:7-LVM:7.9.2020111202	OS and data disk (see note below)
RedHat	RHEL 7.9	7_9	RedHat:RHEL:7_9:latest	OS and data disk (see note below)
RedHat	RHEL 7.8	7.8	RedHat:RHEL:7.8:latest	OS and data disk (see note below)
RedHat	RHEL 7.7	7.7	RedHat:RHEL:7.7:latest	OS and data disk (see note below)
RedHat	RHEL 7.6	7.6	RedHat:RHEL:7.6:latest	OS and data disk (see note below)

Publisher	Offer	SKU	URN	Volume type supported for encryption
RedHat	RHEL 7.5	7.5	RedHat:RHEL:7.5:latest	OS and data disk (see note below)
RedHat	RHEL 7.4	7.4	RedHat:RHEL:7.4:latest	OS and data disk (see note below)
RedHat	RHEL 6.8	6.8	RedHat:RHEL:6.8:latest	Data disk (see note below)
RedHat	RHEL 6.7	6.7	RedHat:RHEL:6.7:latest	Data disk (see note below)
SUSE	openSUSE 42.3	42.3	SUSE:openSUSE-Leap:42.3:latest	Data disk only
SUSE	SLES 12-SP4	12-SP4	SUSE:SLES:12-SP4:latest	Data disk only
SUSE	SLES HPC 12-SP3	12-SP3	SUSE:SLES-HPC:12-SP3:latest	Data disk only

* For image versions greater than or equal to May 2023.

ⓘ Note

RHEL:

- The new Azure Disk Encryption implementation is supported for RHEL OS and data disk for RHEL7 Pay-As-You-Go images.
- ADE is also supported for RHEL Bring-Your-Own-Subscription Gold Images, but only **after** the subscription has been registered . For more information, see [Red Hat Enterprise Linux Bring-Your-Own-Subscription Gold Images in Azure](#)

All distros:

- ADE support for a particular offer type does not extend beyond the end-of-life date provided by the publisher.
- The legacy ADE solution (using AAD credentials) is not recommended for new VMs and is not compatible with RHEL versions later than RHEL 7.8 or with Python 3 as default.

Additional VM requirements

Azure Disk Encryption requires the dm-crypt and vfat modules to be present on the system. Removing or disabling vfat from the default image will prevent the system from reading the key volume and obtaining the key needed to unlock the disks on subsequent reboots. System hardening steps that remove the vfat module from the system or enforce expanding the OS mountpoints/folders on data drives are not compatible with Azure Disk Encryption.

Before enabling encryption, the data disks to be encrypted must be properly listed in /etc/fstab. Use the "nofail" option when creating entries, and choose a persistent block device name (as device names in the "/dev/sdX" format may not be associated with the same disk across reboots, particularly after encryption; for more detail on this behavior, see: [Troubleshoot Linux VM device name changes](#)).

Make sure the /etc/fstab settings are configured properly for mounting. To configure these settings, run the mount -a command or reboot the VM and trigger the remount that way. Once that is complete, check the output of the lsblk command to verify that the drive is still mounted.

- If the /etc/fstab file doesn't mount the drive properly before enabling encryption, Azure Disk Encryption won't be able to mount it properly.
- The Azure Disk Encryption process will move the mount information out of /etc/fstab and into its own configuration file as part of the encryption process. Don't be alarmed to see the entry missing from /etc/fstab after data drive encryption completes.
- Before starting encryption, be sure to stop all services and processes that could be writing to mounted data disks and disable them, so that they do not restart automatically after a reboot. These could keep files open on these partitions, preventing the encryption procedure to remount them, causing failure of the encryption.
- After reboot, it will take time for the Azure Disk Encryption process to mount the newly encrypted disks. They won't be immediately available after a reboot. The process needs time to start, unlock, and then mount the encrypted drives before

being available for other processes to access. This process may take more than a minute after reboot depending on the system characteristics.

Here is an example of the commands used to mount the data disks and create the necessary /etc/fstab entries:

Bash

```
sudo UUID0=$(blkid -s UUID -o value /dev/sda1)
sudo UUID1=$(blkid -s UUID -o value /dev/sda2)
sudo mkdir /data0
sudo mkdir /data1
sudo echo "UUID=$UUID0 /data0 ext4 defaults,nofail 0 0" >>/etc/fstab
sudo echo "UUID=$UUID1 /data1 ext4 defaults,nofail 0 0" >>/etc/fstab
sudo mount -a
```

Networking requirements

To enable the Azure Disk Encryption feature, the Linux VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the Linux VM must be able to connect to an Azure Active Directory endpoint, [login.microsoftonline.com].
- To write the encryption keys to your key vault, the Linux VM must be able to connect to the key vault endpoint.
- The Linux VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.
- If your security policy limits access from Azure VMs to the Internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see [Azure Key Vault behind a firewall](#).

Encryption key storage requirements

Azure Disk Encryption requires an Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For details, see [Creating and configuring a key vault for Azure Disk Encryption](#).

Terminology

The following table defines some of the common terms used in Azure disk encryption documentation:

Terminology	Definition
Azure Key Vault	Key Vault is a cryptographic, key management service that's based on Federal Information Processing Standards (FIPS) validated hardware security modules. These standards help to safeguard your cryptographic keys and sensitive secrets. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption .
Azure CLI	The Azure CLI is optimized for managing and administering Azure resources from the command line.
DM-Crypt	DM-Crypt is the Linux-based, transparent disk-encryption subsystem that's used to enable disk encryption on Linux VMs.
Key encryption key (KEK)	The asymmetric key (RSA 2048) that you can use to protect or wrap the secret. You can provide a hardware security module (HSM)-protected key or software-protected key. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption .
PowerShell cmdlets	For more information, see Azure PowerShell cmdlets .

Next steps

- [Quickstart - Create and encrypt a Linux VM with Azure CLI](#)
- [Quickstart - Create and encrypt a Linux VM with Azure PowerShell](#)
- [Azure Disk Encryption scenarios on Linux VMs](#)
- [Azure Disk Encryption prerequisites CLI script](#)
- [Azure Disk Encryption prerequisites PowerShell script](#)
- [Creating and configuring a key vault for Azure Disk Encryption](#)

Azure Disk Encryption for Windows VMs

Article • 01/05/2023

Applies to:  Windows VMs  Flexible scale sets

Azure Disk Encryption helps protect and safeguard your data to meet your organizational security and compliance commitments. It uses the [BitLocker](#) feature of Windows to provide volume encryption for the OS and data disks of Azure virtual machines (VMs), and is integrated with [Azure Key Vault](#) to help you control and manage the disk encryption keys and secrets.

Azure Disk Encryption is zone resilient, the same way as Virtual Machines. For details, see [Azure Services that support Availability Zones](#).

If you use [Microsoft Defender for Cloud](#), you're alerted if you have VMs that aren't encrypted. The alerts show as High Severity and the recommendation is to encrypt these VMs.

VIRTUAL MACHINES RECOMMENDATIONS		TOTAL				
Missing disk encryption		2 of 2 VMs 				
Virtual machines						
NAME	ONBOARDING	SYSTEM UPDATES	ANTIMALWARE	BASELINE	DISK ENCRYPTION	
 ASC-VM1						
 ASC-VM2						

Warning

- If you have previously used Azure Disk Encryption with Azure AD to encrypt a VM, you must continue use this option to encrypt your VM. See [Azure Disk Encryption with Azure AD \(previous release\)](#) for details.
- Certain recommendations might increase data, network, or compute resource usage, resulting in additional license or subscription costs. You must have a valid active Azure subscription to create resources in Azure in the supported regions.
- Do not use BitLocker to manually decrypt a VM or disk that was encrypted through Azure Disk Encryption.

You can learn the fundamentals of Azure Disk Encryption for Windows in just a few minutes with the [Create and encrypt a Windows VM with Azure CLI quickstart](#) or the [Create and encrypt a Windows VM with Azure PowerShell quickstart](#).

Supported VMs and operating systems

Supported VMs

Windows VMs are available in a [range of sizes](#). Azure Disk Encryption is supported on Generation 1 and Generation 2 VMs. Azure Disk Encryption is also available for VMs with premium storage.

Azure Disk Encryption is not available on [Basic, A-series VMs](#), or on virtual machines with a less than 2 GB of memory. For more exceptions, see [Azure Disk Encryption: Unsupported scenarios](#).

Supported operating systems

- Windows client: Windows 8 and later.
- Windows Server: Windows Server 2008 R2 and later.
- Windows 10 Enterprise multi-session and later.

Note

Windows Server 2022 and Windows 11 do not support an RSA 2048 bit key. For more information, see [FAQ: What size should I use for my key encryption key?](#)

Windows Server 2008 R2 requires the .NET Framework 4.5 to be installed for encryption; install it from Windows Update with the optional update Microsoft .NET Framework 4.5.2 for Windows Server 2008 R2 x64-based systems ([KB2901983](#)).

Windows Server 2012 R2 Core and Windows Server 2016 Core requires the bdehdcfg component to be installed on the VM for encryption.

Networking requirements

To enable Azure Disk Encryption, the VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the Windows VM must be able to connect to an Azure Active Directory endpoint, [login.microsoftonline.com].
- To write the encryption keys to your key vault, the Windows VM must be able to connect to the key vault endpoint.
- The Windows VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.
- If your security policy limits access from Azure VMs to the Internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see [Azure Key Vault behind a firewall](#).

Group Policy requirements

Azure Disk Encryption uses the BitLocker external key protector for Windows VMs. For domain joined VMs, don't push any group policies that enforce TPM protectors. For information about the group policy for "Allow BitLocker without a compatible TPM," see [BitLocker Group Policy Reference](#).

BitLocker policy on domain joined virtual machines with custom group policy must include the following setting: [Configure user storage of BitLocker recovery information - > Allow 256-bit recovery key](#). Azure Disk Encryption will fail when custom group policy settings for BitLocker are incompatible. On machines that didn't have the correct policy setting, apply the new policy, and force the new policy to update (gpupdate.exe /force). Restarting may be required.

Microsoft BitLocker Administration and Monitoring (MBAM) group policy features aren't compatible with Azure Disk Encryption.

Warning

Azure Disk Encryption **does not store recovery keys**. If the **Interactive logon: Machine account lockout threshold** security setting is enabled, machines can only be recovered by providing a recovery key via the serial console. Instructions for ensuring the appropriate recovery policies are enabled can be found in the [Bitlocker recovery guide plan](#).

Azure Disk Encryption will fail if domain level group policy blocks the AES-CBC algorithm, which is used by BitLocker.

Encryption key storage requirements

Azure Disk Encryption requires an Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For details, see [Creating and configuring a key vault for Azure Disk Encryption](#).

Terminology

The following table defines some of the common terms used in Azure disk encryption documentation:

Terminology	Definition
Azure Key Vault	Key Vault is a cryptographic, key management service that's based on Federal Information Processing Standards (FIPS) validated hardware security modules. These standards help to safeguard your cryptographic keys and sensitive secrets. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption .
Azure CLI	The Azure CLI is optimized for managing and administering Azure resources from the command line.
BitLocker	BitLocker is an industry-recognized Windows volume encryption technology that's used to enable disk encryption on Windows VMs.
Key encryption key (KEK)	The asymmetric key (RSA 2048) that you can use to protect or wrap the secret. You can provide a hardware security module (HSM)-protected key or software-protected key. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption .
PowerShell cmdlets	For more information, see Azure PowerShell cmdlets .

Next steps

- [Quickstart - Create and encrypt a Windows VM with Azure CLI](#)
- [Quickstart - Create and encrypt a Windows VM with Azure PowerShell](#)
- [Azure Disk Encryption scenarios on Windows VMs](#)
- [Azure Disk Encryption prerequisites CLI script](#)
- [Azure Disk Encryption prerequisites PowerShell script](#)
- [Creating and configuring a key vault for Azure Disk Encryption](#)

Azure premium storage: Design for high performance

Article • 11/23/2023

Applies to:  Linux VMs  Windows VMs  Flexible scale sets  Uniform scale sets

This article provides guidelines for building high-performance applications by using Azure premium storage. You can use the instructions provided in this document combined with performance best practices applicable to technologies used by your application. To illustrate the guidelines, we use SQL Server running on premium storage as an example throughout this document.

While we address performance scenarios for the storage layer in this article, you need to optimize the application layer. For example, if you're hosting a SharePoint Farm on premium storage, you can use the SQL Server examples from this article to optimize the database server. You can also optimize the SharePoint Farm's web server and application server to get the most performance.

This article helps to answer the following common questions about optimizing application performance on premium storage:

- How can you measure your application performance?
- Why aren't you seeing expected high performance?
- Which factors influence your application performance on premium storage?
- How do these factors influence performance of your application on premium storage?
- How can you optimize for input/output operations per second (IOPS), bandwidth, and latency?

We provide these guidelines specifically for premium storage because workloads running on premium storage are highly performance sensitive. We provide examples where appropriate. You can also apply some of these guidelines to applications running on infrastructure as a service (IaaS) VMs with standard storage disks.

Note

Sometimes what appears to be a disk performance issue is actually a network bottleneck. In these situations, you should optimize your **network performance**.

If you're looking to benchmark your disk, see the following articles:

- For Linux: [Benchmark your application on Azure Disk Storage](#)
- For Windows: [Benchmark a disk](#)

If your VM supports accelerated networking, make sure it's enabled. If it's not enabled, you can enable it on already deployed VMs on both [Windows](#) and [Linux](#).

Before you begin, if you're new to premium storage, first read [Select an Azure disk type for IaaS VMs](#) and [Scalability targets for premium page blob storage accounts](#).

Application performance indicators

We assess whether an application is performing well or not by using performance indicators like:

- How fast an application is processing a user request.
- How much data an application is processing per request.
- How many requests an application is processing in a specific period of time.
- How long a user has to wait to get a response after submitting their request.

The technical terms for these performance indicators are IOPS, throughput or bandwidth, and latency.

In this section, we discuss the common performance indicators in the context of premium storage. In the section [Performance application checklist for disks](#), you learn how to measure these performance indicators for your application. Later in [Optimize application performance](#), you learn about the factors that affect these performance indicators and recommendations to optimize them.

IOPS

IOPS is the number of requests that your application is sending to storage disks in one second. An input/output operation could be read or write, sequential, or random. Online transaction processing (OLTP) applications like an online retail website need to process many concurrent user requests immediately. The user requests are insert- and update-intensive database transactions, which the application must process quickly. For this reason, OLTP applications require very high IOPS.

OLTP applications handle millions of small and random I/O requests. If you have such an application, you must design the application infrastructure to optimize for IOPS. For more information on all the factors to consider to get high IOPS, see [Optimize application performance](#).

When you attach a premium storage disk to your high-scale VM, Azure provisions for you a guaranteed number of IOPS according to the disk specification. For example, a P50 disk provisions 7,500 IOPS. Each high-scale VM size also has a specific IOPS limit that it can sustain. For example, a Standard GS5 VM has an 80,000 IOPS limit.

Throughput

Throughput, or bandwidth, is the amount of data that your application is sending to the storage disks in a specified interval. If your application is performing input/output operations with large I/O unit sizes, it requires high throughput. Data warehouse applications tend to issue scan-intensive operations that access large portions of data at a time and commonly perform bulk operations. In other words, such applications require higher throughput. If you have such an application, you must design its infrastructure to optimize for throughput. In the next section, we discuss the factors you must tune to achieve this optimization.

When you attach a premium storage disk to a high-scale VM, Azure provisions throughput according to that disk specification. For example, a P50 disk provisions 250 MB/sec disk throughput. Each high-scale VM size also has a specific throughput limit that it can sustain. For example, Standard GS5 VM has a maximum throughput of 2,000 MB/sec.

There's a relation between throughput and IOPS, as shown in the following formula.

$$\text{IOPS} \times \text{IO Size} = \text{Throughput}$$

It's important to determine the optimal throughput and IOPS values that your application requires. As you try to optimize one, the other is also affected. For more information about optimizing IOPS and throughput, see [Optimize application performance](#).

Latency

Latency is the time it takes an application to receive a single request, send it to storage disks, and send the response to the client. Latency is a critical measure of an application's performance in addition to IOPS and throughput. The latency of a premium storage disk is the time it takes to retrieve the information for a request and communicate it back to your application. Premium storage provides consistently low latencies. Premium disks are designed to provide single-digit millisecond latencies for most I/O operations. If you enable **ReadOnly** host caching on premium storage disks, you can get much lower read latency. For more information on disk caching, see [Disk caching](#).

When you optimize your application to get higher IOPS and throughput, it affects the latency of your application. After you tune the application performance, always evaluate the latency of the application to avoid unexpected high latency behavior.

The following control plane operations on managed disks might involve movement of the disk from one storage location to another. This movement is orchestrated via the background copy of data, which can take several hours to complete. Typically, the time is less than 24 hours depending on the amount of data in the disks. During that time, your application can experience higher than usual read latency because some reads can get redirected to the original location and take longer to complete.

There's no effect on write latency during this period. For Premium SSD v2 and Ultra Disks, if the disk has a 4K sector size, it experiences higher read latency. If the disk has a 512e sector size, it experiences both higher read and write latency.

Control plane operations are used to:

- Update the storage type.
- Detach and attach a disk from one VM to another.
- Create a managed disk from a VHD.
- Create a managed disk from a snapshot.
- Convert unmanaged disks to managed disks.

Performance application checklist for disks

The first step in designing high-performance applications running on premium storage is understanding the performance requirements of your application. After you gather performance requirements, you can optimize your application to achieve the most optimal performance.

In the previous section, we explained the common performance indicators: IOPS, throughput, and latency. You must identify which of these performance indicators are critical to your application to deliver the desired user experience. For example, high IOPS matters most to OLTP applications processing millions of transactions in a second. High throughput is critical for data warehouse applications processing large amounts of data in a second. Extremely low latency is crucial for real-time applications like live video-streaming websites.

Next, measure the maximum performance requirements of your application throughout its lifetime. Use the following sample checklist as a start. Record the maximum performance requirements during normal, peak, and off-hour workload periods. By identifying requirements for all workload levels, you can determine the overall performance requirement of your application.

For example, the normal workload of an e-commerce website is the transactions it serves during most days in a year. The peak workload of the website is the transactions it serves during holiday seasons or special sale events. The peak workload is typically experienced for a limited period but can require your application to scale two or more times its normal operation. Find out the 50 percentile, 90 percentile, and 99 percentile requirements. This information helps filter out any outliers in the performance requirements, and you can focus your efforts on optimizing for the right values.

Application performance requirements checklist

Performance requirements	50 percentile	90 percentile	99 percentile
Maximum transactions per second			
% Read operations			
% Write operations			
% Random operations			
% Sequential operations			
I/O request size			
Average throughput			
Maximum throughput			
Minimum latency			
Average latency			
Maximum CPU			
Average CPU			
Maximum memory			
Average memory			
Queue depth			

Note

Consider scaling these numbers based on expected future growth of your application. It's a good idea to plan for growth ahead of time because it could be harder to change the infrastructure for improving performance later.

If you have an existing application and want to move to premium storage, first build the preceding checklist for the existing application. Then, build a prototype of your application on premium storage and design the application based on guidelines described in [Optimize application performance](#). The next article describes the tools you can use to gather the performance measurements.

Counters to measure application performance requirements

The best way to measure performance requirements of your application is to use [PerfMon](#)-monitoring tools provided by the operating system of the server. You can use [PerfMon](#) for Windows and [iostat](#) for Linux. These tools capture counters corresponding to each measure explained in the preceding section. You must capture the values of these counters when your application is running its normal, peak, and off-hour workloads.

The [PerfMon](#) counters are available for processor, memory, and each logical disk and physical disk of your server. When you use premium storage disks with a VM, the physical disk counters are for each premium storage disk, and logical disk counters are for each volume created on the premium storage disks. You must capture the values for the disks that host your application workload. If there's a one-to-one mapping between logical and physical disks, you can refer to physical disk counters. Otherwise, refer to the logical disk counters.

On Linux, the [iostat](#) command generates a CPU and disk utilization report. The disk utilization report provides statistics per physical device or partition. If you have a database server with its data and logs on separate disks, collect this data for both disks. The following table describes counters for disks, processors, and memory.

Counter	Description	PerfMon	iostat
IOPS or transactions/sec	Number of I/O requests issued to the storage disk/sec	Disk reads/sec Disk writes/sec	tps r/s w/s
Disk reads and writes	% of read and write operations performed on the disk	% Disk read time % Disk write time	r/s w/s
Throughput	Amount of data read from or written to the disk/sec	Disk read bytes/sec Disk write bytes/sec	kB_read/s kB_wrtn/s
Latency	Total time to complete a disk I/O request	Average disk sec/read Average disk sec/write	await svctm
I/O size	The size of I/O request issues to the storage disks	Average disk bytes/read Average disk bytes/write	avgrq-sz
Queue depth	Number of outstanding I/O requests waiting to be read from or written to the storage disk	Current disk queue length	avgqu-sz
Maximum memory	Amount of memory required to run the application smoothly	% Committed bytes in use	Use vmstat
Maximum CPU	Amount of CPU required to run the application smoothly	% Processor time	%util

Learn more about [iostat](#) and [PerfMon](#).

Optimize application performance

The main factors that influence performance of an application running on premium storage are the nature of I/O requests, VM size, disk size, number of disks, disk caching, multithreading, and queue depth. You can control some of these factors with knobs provided by the system.

Most applications might not give you an option to alter the I/O size and queue depth directly. For example, if you're using SQL Server, you can't choose the I/O size and queue depth. SQL Server chooses the optimal I/O size and queue depth values to get the most performance. It's important to understand the effects of both types of factors on your application performance so that you can provision appropriate resources to meet performance needs.

Throughout this section, refer to the application requirements checklist that you created to identify how much you need to optimize your application performance. Based on the checklist, you can determine which factors from this section you need to tune.

To witness the effects of each factor on your application performance, run benchmarking tools on your application setup. For steps to run common benchmarking tools on Windows and Linux VMs, see the benchmarking articles at the end of this document.

Optimize IOPS, throughput, and latency at a glance

The following table summarizes performance factors and the steps necessary to optimize IOPS, throughput, and latency. The sections following this summary describe each factor in more depth.

For more information on VM sizes and on the IOPS, throughput, and latency available for each type of VM, see [Sizes for virtual machines in Azure](#).

Performance factors	IOPS	Throughput	Latency
Example scenario	Enterprise OLTP application requiring very high transactions per second rate.	Enterprise Data warehousing application processing large amounts of data.	Near real-time applications requiring instant responses to user requests, like online gaming.
Performance factors			
I/O size	Smaller I/O size yields higher IOPS.	Larger I/O size yields higher throughput.	
VM size	Use a VM size that offers IOPS greater than your application requirement.	Use a VM size with a throughput limit greater than your application requirement.	Use a VM size that offers scale limits greater than your application requirement.
Disk size	Use a disk size that offers IOPS greater than your application requirement.	Use a disk size with a throughput limit greater than your application requirement.	Use a disk size that offers scale limits greater than your application requirement.
VM and disk scale limits	IOPS limit of the VM size chosen should be greater than the total IOPS driven by the storage disks attached to it.	Throughput limit of the VM size chosen should be greater than the total throughput driven by the premium storage disks attached to it.	Scale limits of the VM size chosen must be greater than the total scale limits of the attached premium storage disks.
Disk caching	Enable ReadOnly cache on premium storage disks with read-heavy operations to get higher read IOPS.		Enable ReadOnly cache on premium storage disks with read-heavy operations to get very low read latencies.
Disk striping	Use multiple disks and stripe them together to get a combined higher IOPS and throughput limit. The combined limit per VM should be higher than the combined limits of attached premium disks.		
Stripe size	Smaller stripe size for random small I/O pattern seen in OLTP applications. For example, use a 64-KB stripe size for a SQL Server OLTP application.	Larger stripe size for sequential large I/O pattern seen in data warehouse applications. For example, use a 256-KB stripe size for a SQL Server data warehouse application.	
Multithreading	Use multithreading to push a higher number of requests to premium storage to lead to higher IOPS and throughput. For example, on SQL Server, set a high MAXDOP value to allocate more CPUs to SQL Server.		
Queue depth	Larger queue depth yields higher IOPS.	Larger queue depth yields higher throughput.	Smaller queue depth yields lower latencies.

Nature of I/O requests

An I/O request is a unit of input/output operation that your application is performing. Identifying the nature of I/O requests, random or sequential, read or write, small or large, helps you determine the performance requirements of your application. It's important to understand the nature of I/O requests to make the right decisions when you design your application infrastructure. I/Os must be distributed evenly to achieve the best performance possible.

I/O size is one of the more important factors. The I/O size is the size of the input/output operation request generated by your application. The I/O size affects performance significantly, especially on the IOPS and bandwidth that the application can achieve. The following formula shows the relationship between IOPS, I/O size, and bandwidth/throughput.



Some applications allow you to alter their I/O size, while some applications don't. For example, SQL Server determines the optimal I/O size itself and doesn't provide users with any knobs to change it. On the other hand, Oracle provides a parameter called [DB_BLOCK_SIZE](#), which you can use to configure the I/O request size of the database.

If you're using an application, which doesn't allow you to change the I/O size, use the guidelines in this article to optimize the performance KPI that's most relevant to your application. For example:

- An OLTP application generates millions of small and random I/O requests. To handle these types of I/O requests, you must design your application infrastructure to get higher IOPS.
- A data warehousing application generates large and sequential I/O requests. To handle these types of I/O requests, you must design your application infrastructure to get higher bandwidth or throughput.

If you're using an application that allows you to change the I/O size, use this rule of thumb for the I/O size in addition to other performance guidelines:

- Smaller I/O size to get higher IOPS. For example, 8 KB for an OLTP application.
- Larger I/O size to get higher bandwidth/throughput. For example, 1,024 KB for a data warehouse application.

Here's an example of how you can calculate the IOPS and throughput/bandwidth for your application.

Consider an application that uses a P30 disk. The maximum IOPS and throughput/bandwidth a P30 disk can achieve is 5,000 IOPS and 200 MB/sec, respectively. If your application requires the maximum IOPS from the P30 disk and you use a smaller I/O size, like 8 KB, the resulting bandwidth you can get is 40 MB/sec. If your application requires the maximum throughput/bandwidth from a P30 disk and you use a larger I/O size, like 1,024 KB, the resulting IOPS is less, such as 200 IOPS.

Tune the I/O size so that it meets both your application's IOPS and throughput/bandwidth requirement. The following table summarizes the different I/O sizes and their corresponding IOPS and throughput for a P30 disk.

Application requirement	I/O size	IOPS	Throughput/Bandwidth
Maximum IOPS	8 KB	5,000	40 MB/sec
Maximum throughput	1,024 KB	200	200 MB/sec
Maximum throughput + high IOPS	64 KB	3,200	200 MB/sec
Maximum IOPS + high throughput	32 KB	5,000	160 MB/sec

To get IOPS and bandwidth higher than the maximum value of a single premium storage disk, use multiple premium disks striped together. For example, stripe two P30 disks to get a combined IOPS of 10,000 IOPS or a combined throughput of 400 MB/sec. As explained in the next section, you must use a VM size that supports the combined disk IOPS and throughput.

Note

As you increase either IOPS or throughput, the other also increases. Make sure you don't hit throughput or IOPS limits of the disk or VM when you increase either one.

To witness the effects of I/O size on application performance, you can run benchmarking tools on your VM and disks. Create multiple test runs and use different I/O size for each run to see the effect. For more information, see the benchmarking articles at the end of this document.

High-scale VM sizes

When you start designing an application, one of the first things to do is choose a VM to host your application. Premium storage comes with high-scale VM sizes that can run applications requiring higher compute power and a high local disk I/O performance. These VMs provide faster processors, a higher memory-to-core ratio, and a solid-state drive (SSD) for the local disk. Examples of high-scale VMs supporting premium storage are the DS and GS series VMs.

High-scale VMs are available in different sizes with a different number of CPU cores, memory, OS, and temporary disk size. Each VM size also has a maximum number of data disks that you can attach to the VM. The chosen VM size affects how much processing, memory, and storage capacity are available for your application. It also affects the compute and storage cost. For example, the following specifications are for the largest VM size in a DS series and a GS series.

VM size	CPU cores	Memory	VM disk sizes	Maximum data disks	Cache size	IOPS	Bandwidth cache I/O limits
Standard_DS14	16	112 GB	OS = 1,023 GB Local SSD = 224 GB	32	576 GB	50,000 IOPS 512 MB/sec	4,000 IOPS and 33 MB/sec
Standard_GS5	32	448 GB	OS = 1,023 GB Local SSD = 896 GB	64	4224 GB	80,000 IOPS 2,000 MB/sec	5,000 IOPS and 50 MB/sec

To view a complete list of all available Azure VM sizes, see [Sizes for virtual machines in Azure](#). Choose a VM size that can meet and scale to your desired application performance requirements. Also take into account the following important considerations when you choose VM sizes.

Scale limits

The maximum IOPS limits per VM and per disk are different and independent of each other. Make sure that the application is driving IOPS within the limits of the VM and the premium disks attached to it. Otherwise, application performance experiences throttling.

As an example, suppose an application requirement is a maximum of 4,000 IOPS. To achieve this level, you provision a P30 disk on a DS1 VM. The P30 disk can deliver up to 5,000 IOPS. However, the DS1 VM is limited to 3,200 IOPS. So, the application performance is constrained by the VM limit at 3,200 IOPS and performance is degraded. To prevent this situation, choose a VM and disk size that both meet application requirements.

Cost of operation

In many cases, it's possible that your overall cost of operation using premium storage is lower than using standard storage.

For example, consider an application requiring 16,000 IOPS. To achieve this performance, you need a Standard_D14 Azure IaaS VM, which can give a maximum IOPS of 16,000 by using 32 standard storage 1-TB disks. Each 1-TB standard storage disk can achieve a maximum of 500 IOPS.

- The estimated cost of this VM per month is \$1,570.
- The monthly cost of 32 standard storage disks is \$1,638.
- The estimated total monthly cost is \$3,208.

If you hosted the same application on premium storage, you need a smaller VM size and fewer premium storage disks, reducing the overall cost. A Standard_DS13 VM can meet the 16,000 IOPS requirement by using four P30 disks. The DS13 VM has a maximum IOPS of 25,600, and each P30 disk has a maximum IOPS of 5,000. Overall, this configuration can achieve $5,000 \times 4 = 20,000$ IOPS.

- The estimated cost of this VM per month is \$1,003.
- The monthly cost of four P30 premium storage disks is \$544.34.
- The estimated total monthly cost is \$1,544.

The following table summarizes the cost breakdown of this scenario for standard and premium storage.

Monthly cost	Standard	Premium
Cost of VM per month	\$1,570.58 (Standard_D14)	\$1,003.66 (Standard_DS13)
Cost of disks per month	\$1,638.40 (32 x 1-TB disks)	\$544.34 (4 x P30 disks)
Overall cost per month	\$3,208.98	\$1,544.34

Linux distros

With premium storage, you get the same level of performance for VMs running Windows and Linux. We support many flavors of Linux distros. For more information, see [Linux distributions endorsed on Azure](#).

Different distros are better suited for different types of workloads. You see different levels of performance depending on the distro on which your workload is running. Test the Linux distros with your application and choose the one that works best.

When you run Linux with premium storage, check the latest updates about required drivers to ensure high performance.

Premium storage disk sizes

Premium storage offers various sizes so you can choose one that best suits your needs. Each disk size has a different scale limit for IOPS, bandwidth, and storage. Choose the right premium storage disk size depending on the application requirements and the high-scale VM size. The following table shows the disks sizes and their capabilities. P4, P6, P15, P60, P70, and P80 sizes are currently only supported for managed disks.

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base provisioned IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
**Expanded provisioned IOPS per disk	N/A	8,000	16,000	20,000	20,000	20,000	20,000							
Base provisioned Throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	750 MB/s	900 MB/s
**Expanded provisioned throughput per disk	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	900 MB/s	900 MB/s							
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,000*	30,000*
Max burst throughput per disk	170 MB/s	1,000 MB/s*												
Max burst duration	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*							
Eligible for reservation	No	Yes, up to one year												

*Applies only to disks with on-demand bursting enabled.

** Only applies to disks with performance plus (preview) enabled.

How many disks you choose depends on the disk size chosen. You could use a single P50 disk or multiple P10 disks to meet your application requirement. Take into account considerations listed here when you're making the choice.

Scale limits (IOPS and throughput)

The IOPS and throughput limits of each premium disk size is different and independent from the VM scale limits. Make sure that the total IOPS and throughput from the disks are within scale limits of the chosen VM size.

For example, if an application requirement is a maximum of 250 MB/sec throughput and you're using a DS4 VM with a single P30 disk, the DS4 VM can give up to 256 MB/sec throughput. However, a single P30 disk has a throughput limit of 200 MB/sec. So, the application is constrained at 200 MB/sec because of the disk limit. To overcome this limit, provision more than one data disk to the VM or resize your disks to P40 or P50.

⚠ Note

Reads served by the cache aren't included in the disk IOPS and throughput, so they aren't subject to disk limits. Cache has its separate IOPS and throughput limit per VM.

For example, initially your reads and writes are 60 MB/sec and 40 MB/sec, respectively. Over time, the cache warms up and serves more and more of the reads from the cache. Then, you can get higher write throughput from the disk.

Number of disks

Determine the number of disks you need by assessing application requirements. Each VM size also has a limit on the number of disks that you can attach to the VM. Typically, this amount is twice the number of cores. Ensure that the VM size you choose can support the number of disks needed.

Remember, the premium storage disks have higher performance capabilities compared to standard storage disks. If you're migrating your application from an Azure IaaS VM using standard storage to premium storage, you likely need fewer premium disks to achieve the same or higher performance for your application.

Disk caching

High-scale VMs that use premium storage have a multilayer caching technology called **BlobCache**. **BlobCache** uses a combination of the host RAM and local SSD for caching. This cache is available for the premium storage persistent disks and the VM local disks. By default, this cache setting is set to **ReadWrite** for OS disks and **ReadOnly** for data disks hosted on premium storage. With disk caching enabled on the premium storage disks, the high-scale VMs can achieve extremely high levels of performance that exceed the underlying disk performance.

⚠ Warning

Disk caching isn't supported for disks 4 TiB and larger. If multiple disks are attached to your VM, each disk that's smaller than 4 TiB supports caching.

Changing the cache setting of an Azure disk detaches and reattaches the target disk. If it's the operating system disk, the VM is restarted. Stop all applications and services that might be affected by this disruption before you change the disk cache setting. Not following those recommendations could lead to data corruption.

To learn more about how **BlobCache** works, see the [Inside Azure premium storage](#) blog post.

It's important to enable caching on the right set of disks. Whether you should enable disk caching on a premium disk or not depends on the workload pattern that disk is handling. The following table shows the default cache settings for OS and data disks.

Disk type	Default cache setting
OS disk	ReadWrite
Data disk	ReadOnly

We recommend the following disk cache settings for data disks.

Disk caching setting	Recommendation for when to use this setting
None	Configure host-cache as None for write-only and write-heavy disks.
ReadOnly	Configure host-cache as ReadOnly for read-only and read-write disks.

Disk caching setting	Recommendation for when to use this setting
ReadWrite	Configure host-cache as ReadWrite only if your application properly handles writing cached data to persistent disks when needed.

ReadOnly

By configuring **ReadOnly** caching on premium storage data disks, you can achieve low read latency and get very high read IOPS and throughput for your application for two reasons:

1. Reads performed from cache, which is on the VM memory and local SSD, are faster than reads from the data disk, which is on Azure Blob Storage.
2. Premium storage doesn't count the reads served from the cache toward the disk IOPS and throughput. For this reason, your application can achieve higher total IOPS and throughput.

ReadWrite

By default, the OS disks have **ReadWrite** caching enabled. We recently added support for **ReadWrite** caching on data disks too. If you're using **ReadWrite** caching, you must have a proper way to write the data from cache to persistent disks. For example, SQL Server handles writing cached data to the persistent storage disks on its own. Using **ReadWrite** cache with an application that doesn't handle persisting the required data can lead to data loss, if the VM crashes.

None

Currently, **None** is only supported on data disks. It isn't supported on OS disks. If you set **None** on an OS disk, it overrides this setting internally and sets it to **ReadOnly**.

As an example, you can apply these guidelines to SQL Server running on premium storage by following these steps:

1. Configure the **ReadOnly** cache on premium storage disks hosting data files.
 - a. The fast reads from cache lower the SQL Server query time because data pages are retrieved faster from the cache compared to directly from the data disks.
 - b. Serving reads from cache means there's more throughput available from premium data disks. SQL Server can use this extra throughput toward retrieving more data pages and other operations like backup/restore, batch loads, and index rebuilds.
2. Configure the **None** cache on premium storage disks hosting the log files.
 - a. Log files have primarily write-heavy operations, so they don't benefit from the **ReadOnly** cache.

Optimize performance on Linux VMs

For all Premium SSDs or Ultra Disks, you might be able to disable *barriers* for file systems on the disk to improve performance when it's known that there are no caches that could lose data. If Azure disk caching is set to **ReadOnly** or **None**, you can disable barriers. But if caching is set to **ReadWrite**, barriers should remain enabled to ensure write durability. Barriers are typically enabled by default, but you can disable barriers by using one of the following methods depending on the file system type:

- **reiserFS**: Use the **barrier=none** mount option to disable barriers. To explicitly enable barriers, use **barrier=flush**.
- **ext3/ext4**: Use the **barrier=0** mount option to disable barriers. To explicitly enable barriers, use **barrier=1**.
- **XFS**: Use the **nobarrier** mount option to disable barriers. To explicitly enable barriers, use **barrier**. As of version 4.10 of the mainline Linux kernel, the design of the XFS file system always ensures durability. Disabling barriers has no effect and the **nobarrier** option is deprecated. However, some Linux distributions might have backported the changes to a distribution release with an earlier kernel version. Check with your distribution vendor for the status in the distribution and version that you're running.

Disk striping

When a high-scale VM is attached with several premium storage persistent disks, the disks can be striped together to aggregate their IOPs, bandwidth, and storage capacity.

On Windows, you can use Storage Spaces to stripe disks together. You must configure one column for each disk in a pool. Otherwise, the overall performance of striped volume can be lower than expected because of uneven distribution of traffic across the disks.

By using the Server Manager UI, you can set the total number of columns up to 8 for a striped volume. When you're attaching more than eight disks, use PowerShell to create the volume. By using PowerShell, you can set the number of columns equal to the number of disks. For example, if there are 16 disks in a single stripe set, specify 16 columns in the `NumberOfColumns` parameter of the `New-VirtualDisk` PowerShell cmdlet.

On Linux, use the MDADM utility to stripe disks together. For steps on how to stripe disks on Linux, see [Configure Software RAID on Linux](#).

Stripe size

An important configuration in disk striping is the stripe size. The stripe size or block size is the smallest chunk of data that an application can address on a striped volume. The stripe size you configure depends on the type of application and its request pattern. If you choose the wrong stripe size, it could lead to I/O misalignment, which leads to degraded performance of your application.

For example, if an I/O request generated by your application is bigger than the disk stripe size, the storage system writes it across stripe unit boundaries on more than one disk. When it's time to access that data, it has to seek across more than one stripe unit to complete the request. The cumulative effect of such behavior can lead to substantial performance degradation. On the other hand, if the I/O request size is smaller than the stripe size, and if it's random in nature, the I/O requests might add up on the same disk, causing a bottleneck and ultimately degrading the I/O performance.

Depending on the type of workload your application is running, choose an appropriate stripe size. For random small I/O requests, use a smaller stripe size. For large sequential I/O requests, use a larger stripe size. Find out the stripe size recommendations for the application you'll be running on premium storage. For SQL Server, configure a stripe size of 64 KB for OLTP workloads and 256 KB for data warehousing workloads. For more information, see [Performance best practices for SQL Server on Azure VMs](#).

 **Note**

You can stripe together a maximum of 32 premium storage disks on a DS series VM and 64 premium storage disks on a GS series VM.

Multithreading

Azure designed the premium storage platform to be massively parallel. For this reason, a multithreaded application achieves higher performance than a single-threaded application. A multithreaded application splits up its tasks across multiple threads and increases efficiency of its execution by utilizing the VM and disk resources to the maximum.

For example, if your application is running on a single core VM using two threads, the CPU can switch between the two threads to achieve efficiency. While one thread is waiting on a disk I/O to complete, the CPU can switch to the other thread. In this way, two threads can accomplish more than a single thread would. If the VM has more than one core, it further decreases running time because each core can run tasks in parallel.

You might not be able to change the way an off-the-shelf application implements single threading or multithreading. For example, SQL Server is capable of handling multi-CPU and multicore. However, SQL Server decides under what conditions it uses one or more threads to process a query. It can run queries and build indexes by using multithreading. For a query that involves joining large tables and sorting data before returning to the user, SQL Server likely uses multiple threads. A user can't control whether SQL Server runs a query by using a single thread or multiple threads.

There are configuration settings that you can alter to influence the multithreading or parallel processing of an application. For example, for SQL Server it's the `max degree of parallelism` configuration. This setting called MAXDOP allows you to configure the maximum number of processors SQL Server can use when parallel processing. You can configure MAXDOP for individual queries or index operations. This capability is beneficial when you want to balance resources of your system for a performance critical application.

For example, say your application that's using SQL Server is running a large query and an index operation at the same time. Let's assume that you wanted the index operation to be more performant compared to the large query. In such a case, you can set the

MAXDOP value of the index operation to be higher than the MAXDOP value for the query. This way, SQL Server has more processors than it can use for the index operation compared to the number of processors it can dedicate to the large query. Remember, you don't control the number of threads that SQL Server uses for each operation. You can control the maximum number of processors being dedicated for multithreading.

Learn more about [degrees of parallelism](#) in SQL Server. Find out how such settings influence multithreading in your application and their configurations to optimize performance.

Queue depth

The queue depth or queue length or queue size is the number of pending I/O requests in the system. The value of queue depth determines how many I/O operations your application can line up, which the storage disks process. It affects all three application performance indicators discussed in this article: IOPS, throughput, and latency.

Queue depth and multithreading are closely related. The queue depth value indicates how much multithreading can be achieved by the application. If the queue depth is large, the application can run more operations concurrently, in other words, more multithreading. If the queue depth is small, even though the application is multithreaded, it won't have enough requests lined up for concurrent execution.

Typically, off-the-shelf applications don't allow you to change queue depth, because if it's set incorrectly, it does more harm than good. Applications set the right value of queue depth to get the optimal performance. It's important to understand this concept so that you can troubleshoot performance issues with your application. You can also observe the effects of queue depth by running benchmarking tools on your system.

Some applications provide settings to influence the queue depth. For example, the MAXDOP setting in SQL Server explained in the previous section. MAXDOP is a way to influence queue depth and multithreading, although it doesn't directly change the queue depth value of SQL Server.

High queue depth

A high queue depth lines up more operations on the disk. The disk knows the next request in its queue ahead of time. So, the disk can schedule operations ahead of time and process them in an optimal sequence. Because the application is sending more requests to the disk, the disk can process more parallel I/Os. Ultimately, the application can achieve higher IOPS. Because the application is processing more requests, the total throughput of the application also increases.

Typically, an application can achieve maximum throughput with 8 to 16+ outstanding I/Os per attached disk. If a queue depth is one, the application isn't pushing enough I/Os to the system, and it processes a smaller amount in a given period. In other words, less throughput.

For example, in SQL Server, setting the MAXDOP value for a query to 4 informs SQL Server that it can use up to four cores to run the query. SQL Server determines the best queue depth value and the number of cores for the query execution.

Optimal queue depth

A very high queue depth value also has its drawbacks. If the queue depth value is too high, the application tries to drive very high IOPS. Unless the application has persistent disks with sufficient provisioned IOPS, a very high queue depth value can negatively affect application latencies. The following formula shows the relationship between IOPS, latency, and queue depth.

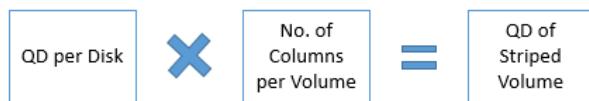


You shouldn't configure queue depth to any high value, but to an optimal value, which can deliver enough IOPS for the application without affecting latencies. For example, if the application latency needs to be 1 millisecond, the queue depth required to achieve 5,000 IOPS is $QD = 5,000 \times 0.001 = 5$.

Queue depth for striped volume

For a striped volume, maintain a high-enough queue depth so that every disk has a peak queue depth individually. For example, consider an application that pushes a queue depth of 2 and there are four disks in the stripe. The two I/O requests go to two disks

and the remaining two disks are idle. Therefore, configure the queue depth so that all the disks can be busy. The following formula shows how to determine the queue depth of striped volumes.



Throttling

Premium storage provisions a specified number of IOPS and throughput depending on the VM sizes and disk sizes you choose. Anytime your application tries to drive IOPS or throughput above these limits of what the VM or disk can handle, premium storage throttles it. The result is degraded performance in your application, which can mean higher latency, lower throughput, or lower IOPS.

If premium storage doesn't throttle, your application could completely fail by exceeding what its resources are capable of achieving. To avoid performance issues because of throttling, always provision sufficient resources for your application. Take into consideration what we discussed in the previous VM sizes and disk sizes sections. Benchmarking is the best way to figure out what resources you need to host your application.

Next steps

If you're looking to benchmark your disk, see the following articles:

- For Linux: [Benchmark your application on Azure Disk Storage](#)
- For Windows: [Benchmark a disk](#)

Learn more about the available disk types:

- For Linux: [Select a disk type](#)
- For Windows: [Select a disk type](#)

For SQL Server users, see the articles on performance best practices for SQL Server:

- [Performance best practices for SQL Server in Azure VMs](#)
- [Azure premium storage provides highest performance for SQL Server in Azure VM](#)

Scalability and performance targets for VM disks

Article • 05/23/2023

Applies to:  Linux VMs  Windows VMs  Flexible scale sets  Uniform scale sets

You can attach a number of data disks to an Azure virtual machine (VM). Based on the scalability and performance targets for a VM's data disks, you can determine the number and type of disk that you need to meet your performance and capacity requirements.

Important

For optimal performance, limit the number of highly utilized disks attached to the virtual machine to avoid possible throttling. If all attached disks aren't highly utilized at the same time, the virtual machine can support a larger number of disks. Additionally, when creating a managed disk from an existing managed disk, only 49 disks can be created concurrently. More disks can be created after some of the initial 49 have been created.

For Azure managed disks:

The following table illustrates the default and maximum limits of the number of resources per region per subscription. The limits remain the same irrespective of disks encrypted with either platform-managed keys or customer-managed keys. There is no limit for the number of Managed Disks, snapshots and images per resource group.

Resource	Limit
Standard managed disks	50,000
Standard SSD managed disks	50,000
Premium SSD managed disks	50,000
Premium SSD v2 managed disks	1,000
Premium SSD v2 managed disks capacity ²	32,768
Ultra disks	1,000
Ultra disk capacity ²	32,768
Standard_LRS snapshots ¹	75,000
Standard_ZRS snapshots ¹	75,000
Managed image	50,000

¹An individual disk can have 500 incremental snapshots.

²This is the default max but higher capacities are supported by request. To request an increase in capacity, request a quota increase or contact Azure Support.

For standard storage accounts:

A Standard storage account has a maximum total request rate of 20,000 IOPS. The total IOPS across all of your virtual machine disks in a Standard storage account should not exceed this limit.

For unmanaged disks, you can roughly calculate the number of highly utilized disks supported by a single standard storage account based on the request rate limit. For example, for a Basic tier VM, the maximum number of highly utilized disks is about 66, which is 20,000/300 IOPS per disk. The maximum number of highly utilized disks for a Standard tier VM is about 40, which is 20,000/500 IOPS per disk.

For premium storage accounts:

A premium storage account has a maximum total throughput rate of 50 Gbps. The total throughput across all of your VM disks should not exceed this limit.

See [VM sizes](#) for more details.

Managed virtual machine disks

Standard HDD managed disks

Standard Disk Type	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to 500	Up to 500	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000					
*Expanded IOPS per disk	N/A	N/A	N/A	N/A	N/A	Up to 1,500	Up to 3,000				
Base throughput per disk	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 300 MB/s	Up to 500 MB/s	Up to 500 MB/s					
*Expanded throughput per disk	N/A	N/A	N/A	N/A	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 500 MB/s			

* Only applies to disks with performance plus (preview) enabled.

Standard SSD managed disks

Standard SSD sizes	E1	E2	E3	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to 500	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000									
*Expanded IOPS per disk	N/A	Up to 1,500	Up to 3,000	Up to 6,000	Up to 6,000	Up to 6,000	Up to 6,000							
Base throughput per disk	Up to 60 MB/s	Up to 60 MB/s	Up to 400 MB/s	Up to 600 MB/s	Up to 750 MB/s									
*Expanded throughput per disk	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 600 MB/s	Up to 750 MB/s	Up to 750 MB/s	Up to 750 MB/s							
Max burst IOPS per disk	600	600	600	600	600	600	600	600	600	1000				
Max burst throughput per disk	150 MB/s	250 MB/s												
Max burst duration	30 min	30 min												

* Only applies to disks with performance plus (preview) enabled.

Premium SSD managed disks: Per-disk limits

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Base provisioned IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
**Expanded provisioned IOPS per disk	N/A	8,000	16,000	20,000	20,000	20,000								
Base provisioned Throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	750 MB/s	900 MB/s
**Expanded provisioned Throughput per disk	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	900 MB/s	900 MB/s							
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,000*	30,000*
Max burst throughput per disk	170 MB/s	1,000 MB/s*												
Max burst duration	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*							
Eligible for reservation	No	Yes, up to one year												

*Applies only to disks with on-demand bursting enabled.

** Only applies to disks with performance plus (preview) enabled.

Premium SSD managed disks: Per-VM limits

Resource	Limit
Maximum IOPS Per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/s with GS5 VM

Unmanaged virtual machine disks

Standard unmanaged virtual machine disks: Per-disk limits

VM tier	Basic tier VM	Standard tier VM
Disk size	4,095 GB	4,095 GB
Maximum 8-KB IOPS per persistent disk	300	500
Maximum number of disks that perform the maximum IOPS	66	40

Premium unmanaged virtual machine disks: Per-account limits

Resource	Limit
Total disk capacity per account	35 TB
Total snapshot capacity per account	10 TB

Resource	Limit
Maximum bandwidth per account (ingress + egress) ¹	<=50 Gbps

¹Ingress refers to all data from requests that are sent to a storage account. Egress refers to all data from responses that are received from a storage account.

Premium unmanaged virtual machine disks: Per-disk limits

Premium storage disk type	P10	P20	P30	P40	P50
Disk size	128 GiB	512 GiB	1,024 GiB (1 TB)	2,048 GiB (2 TB)	4,095 GiB (4 TB)
Maximum IOPS per disk	500	2,300	5,000	7,500	7,500
Maximum throughput per disk	100 MB/sec	150 MB/sec	200 MB/sec	250 MB/sec	250 MB/sec
Maximum number of disks per storage account	280	70	35	17	8

Premium unmanaged virtual machine disks: Per-VM limits

Resource	Limit
Maximum IOPS per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/sec with GS5 VM

See also

[Azure subscription and service limits, quotas, and constraints](#)

Create an incremental snapshot for managed disks

Article • 11/17/2023

Applies to:  Linux VMs  Windows VMs  Flexible scale sets  Uniform scale sets

Incremental snapshots are point-in-time backups for managed disks that, when taken, consist only of the changes since the last snapshot. The first incremental snapshot is a full copy of the disk. The subsequent incremental snapshots occupy only delta changes to disks since the last snapshot. When you restore a disk from an incremental snapshot, the system reconstructs the full disk that represents the point in time backup of the disk when the incremental snapshot was taken. This capability for managed disk snapshots potentially allows them to be more cost-effective, since, unless you choose to, you don't have to store the entire disk with each individual snapshot. Just like full snapshots, incremental snapshots can be used to either create a full managed disk or a full snapshot. Both full snapshots and incremental snapshots can be used immediately after being taken. In other words, once you take either snapshot, you can immediately read the underlying data and use it to restore disks.

There are a few differences between an incremental snapshot and a full snapshot. Incremental snapshots will always use standard HDD storage, irrespective of the storage type of the disk, whereas full snapshots can use premium SSDs. If you're using full snapshots on Premium Storage to scale up VM deployments, we recommend you use custom images on standard storage in the [Azure Compute Gallery](#). It will help you achieve a more massive scale with lower cost. Additionally, incremental snapshots potentially offer better reliability with [zone-redundant storage](#) (ZRS). If ZRS is available in the selected region, an incremental snapshot will use ZRS automatically. If ZRS isn't available in the region, then the snapshot will default to [locally-redundant storage](#) (LRS). You can override this behavior and select one manually but, we don't recommend that.

Incremental snapshots are billed for the used size only. You can find the used size of your snapshots by looking at the [Azure usage report](#). For example, if the used data size of a snapshot is 10 GiB, the **daily** usage report will show $10 \text{ GiB}/(31 \text{ days}) = 0.3226$ as the consumed quantity.

Restrictions

- Incremental snapshots currently can't be moved between subscriptions.

- You can currently only generate SAS URIs of up to five snapshots of a particular snapshot family at any given time.
- You can't create an incremental snapshot for a particular disk outside of that disk's subscription.
- Incremental snapshots can't be moved to another resource group. But, they can be copied to another resource group or region.
- Up to seven incremental snapshots per disk can be created every five minutes.
- A total of 500 incremental snapshots can be created for a single disk.
- You can't get the changes between snapshots taken before and after you changed the size of the parent disk across 4-TB boundary. For example, You took an incremental snapshot `snapshot-a` when the size of a disk was 2 TB. Now you increased the size of the disk to 6 TB and then took another incremental snapshot `snapshot-b`. You can't get the changes between `snapshot-a` and `snapshot-b`. You have to download the full copy of `snapshot-b` created after the resize. Subsequently, you can get the changes between `snapshot-b` and snapshots created after `snapshot-b`.
- When you create a managed disk from a snapshot, it starts a background copy process. You can attach a disk to a VM while this process is running but you'll experience [performance impact](#). You can use `CompletionPercent` property to [check the status of the background copy](#) for Ultra Disks and Premium SSD v2 disks.

Incremental snapshots of Premium SSD v2 and Ultra Disks

Incremental snapshots of Premium SSD v2 and Ultra Disks have the following extra restrictions:

- Snapshots with a 512 logical sector size are stored as VHD, and can be used to create any disk type. Snapshots with a 4096 logical sector size are stored as VHDX and can only be used to create Ultra Disks and Premium SSD v2 disks, they can't be used to create other disk types. To determine which sector size your snapshot is, see [check sector size](#).
- Up to five disks may be simultaneously created from a snapshot of a Premium SSD v2 or an Ultra Disk.
- When an incremental snapshot of either a Premium SSD v2 or an Ultra Disk is created, a background copy process for that disk is started. While a background copy is ongoing, you can have up to three total snapshots pending. The process must complete before any more snapshots of that disk can be created.
- Incremental snapshots of a Premium SSD v2 or an Ultra disk can't be used immediately after they're created. The background copy must complete before you

can create a disk from the snapshot. See [Check snapshot status](#) for details.

- Taking increment snapshots of a Premium SSD v2 or an Ultra disk while the CompletionPercent property of the disk hasn't reached 100 isn't supported.
- When you attach a Premium SSD v2 or Ultra disk created from snapshot to a running Virtual Machine while CompletionPercnet property hasn't reached 100, the disk suffers performance impact. Specifically, if the disk has a 4k sector size, it may experience slower read. If the disk has a 512e sector size, it may experience slower read and write. To track the progress of this background copy process, see the the check disk status section of either the Azure [PowerShell sample](#) or the [Azure CLI sample](#).

ⓘ Note

Normally, when you take an incremental snapshot, and there aren't any changes, the size of that snapshot is 0 MiB. Currently, empty snapshots of disks with a 4096 logical sector size instead have a size of 6 MiB, when they'd normally be 0 MiB.

Create incremental snapshots

Azure CLI

You can use the Azure CLI to create an incremental snapshot. You need the latest version of the Azure CLI. See the following articles to learn how to either [install](#) or [update](#) the Azure CLI.

The following script creates an incremental snapshot of a particular disk:

Azure CLI

```
# Declare variables
diskName="yourDiskNameHere"
resourceGroupName="yourResourceGroupNameHere"
snapshotName="desiredSnapshotNameHere"

# Get the disk you need to backup
yourDiskID=$(az disk show -n $diskName -g $resourceGroupName --query
"id" --output tsv)

# Create the snapshot
az snapshot create -g $resourceGroupName -n $snapshotName --source
$yourDiskID --incremental true
```

You can identify incremental snapshots from the same disk with the `SourceResourceId` property of snapshots. `SourceResourceId` is the Azure Resource Manager resource ID of the parent disk.

You can use `SourceResourceId` to create a list of all snapshots associated with a particular disk. Replace `yourResourceGroupNameHere` with your value and then you can use the following example to list your existing incremental snapshots:

Azure CLI

```
# Declare variables and create snapshot list
subscriptionId="yourSubscriptionId"
resourceGroupName="yourResourceGroupNameHere"
diskName="yourDiskNameHere"

az account set --subscription $subscriptionId

diskId=$(az disk show -n $diskName -g $resourceGroupName --query [id] -o
tsv)

az snapshot list --query "[?creationData.sourceResourceId=='$diskId' &&
incremental]" -g $resourceGroupName --output table
```

Check snapshot status

Incremental snapshots of Premium SSD v2 or Ultra Disks can't be used to create new disks until the background process copying the data into the snapshot has completed.

You can use either the [CLI](#) or [PowerShell](#) sections to check the status of the background copy from a disk to a snapshot.

Important

You can't use the following sections to get the status of the background copy process for disk types other than Ultra Disk or Premium SSD v2. Snapshots of other disk types always report 100%.

CLI

You have two options for getting the status of snapshots. You can either get a [list of all incremental snapshots associated with a specific disk](#), and their respective status, or you can get the [status of an individual snapshot](#).

CLI - List incremental snapshots

The following script returns a list of all snapshots associated with a particular disk. The value of the `CompletionPercent` property of any snapshot must be 100 before it can be used. Replace `yourResourceGroupNameHere`, `yourSubscriptionId`, and `yourDiskNameHere` with your values then run the script:

Azure CLI

```
# Declare variables and create snapshot list
subscriptionId="yourSubscriptionId"
resourceGroupName="yourResourceGroupNameHere"
diskName="yourDiskNameHere"
az account set --subscription $subscriptionId
diskId=$(az disk show -n $diskName -g $resourceGroupName --query [id] -o
tsv)
az snapshot list --query "[?creationData.sourceResourceId=='$diskId' &&
incremental]" -g $resourceGroupName --output table
```

CLI - Individual snapshot

You can also check the status of an individual snapshot by checking the `CompletionPercent` property. Replace `$sourceSnapshotName` with the name of your snapshot then run the following command. The value of the property must be 100 before you can use the snapshot for restoring disk or generate a SAS URI for downloading the underlying data.

Azure CLI

```
az snapshot show -n $sourceSnapshotName -g $resourceGroupName --query
[completionPercent] -o tsv
```

PowerShell

You have two options for getting the status of snapshots. You can either get a [list of all incremental snapshots associated with a particular disk](#) and their respective status, or you can get the [status of an individual snapshot](#).

PowerShell - List incremental snapshots

The following script returns a list of all incremental snapshots associated with a particular disk that haven't completed their background copy. Replace `yourResourceGroupNameHere` and `yourDiskNameHere`, then run the script.

Azure PowerShell

```
$resourceGroupName = "yourResourceGroupNameHere"
$snapshots = Get-AzSnapshot -ResourceGroupName $resourceGroupName
$diskName = "yourDiskNameHere"
$yourDisk = Get-AzDisk -DiskName $diskName -ResourceGroupName
$resourceGroupName
$incrementalSnapshots = New-Object System.Collections.ArrayList
foreach ($snapshot in $snapshots)
{
    if($snapshot.Incremental -and $snapshot.CreationData.SourceResourceId -eq $yourDisk.Id -and $snapshot.CreationData.SourceUniqueId -eq $yourDisk.UniqueId)
    {
        $targetSnapshot=Get-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName $snapshotName
        {
            if($targetSnapshot.CompletionPercent -lt 100)
            {
                $incrementalSnapshots.Add($targetSnapshot)
            }
        }
    }
}
$incrementalSnapshots
```

PowerShell - individual snapshots

You can check the `CompletionPercent` property of an individual snapshot to get its status. Replace `yourResourceGroupNameHere` and `yourSnapshotName` then run the script. The value of the property must be 100 before you can use the snapshot for restoring disk or generate a SAS URI for downloading the underlying data.

Azure PowerShell

```
$resourceGroupName = "yourResourceGroupNameHere"
$snapshotName = "yourSnapshotName"
$targetSnapshot=Get-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName $snapshotName
$targetSnapshot.CompletionPercent
```

Check sector size

Snapshots with a 4096 logical sector size can only be used to create Premium SSD v2 or Ultra Disks. They can't be used to create other disk types. Snapshots of disks with 4096 logical sector size are stored as VHDX, whereas snapshots of disks with 512 logical

sector size are stored as VHD. Snapshots inherit the logical sector size from the parent disk.

To determine whether or your Premium SSD v2 or Ultra Disk snapshot is a VHDX or a VHD, get the `LogicalSectorSize` property of the snapshot.

The following command displays the logical sector size of a snapshot:

Azure CLI

```
az snapshot show -g resourcegroupname -n snapshotname --query  
[creationData.logicalSectorSize] -o tsv
```

Next steps

See the following articles to create disks from your snapshots using the [Azure CLI](#) or [Azure PowerShell module](#).

See [Copy an incremental snapshot to a new region](#) to learn how to copy an incremental snapshot across regions.

If you have more questions on snapshots, see the [snapshots](#) section of the FAQ.

If you'd like to see sample code demonstrating the differential capability of incremental snapshots, using .NET, see [Copy Azure Managed Disks backups to another region with differential capability of incremental snapshots](#).

Solution architectures using Azure NetApp Files

Article • 09/18/2023

Azure NetApp Files is an enterprise storage service that offers an ideal landing zone component in Azure to accelerate and simplify the migration of various workload categories. Azure NetApp Files provides a high-performance, scalable, and secure storage service for running mission-critical applications and workloads in Azure.

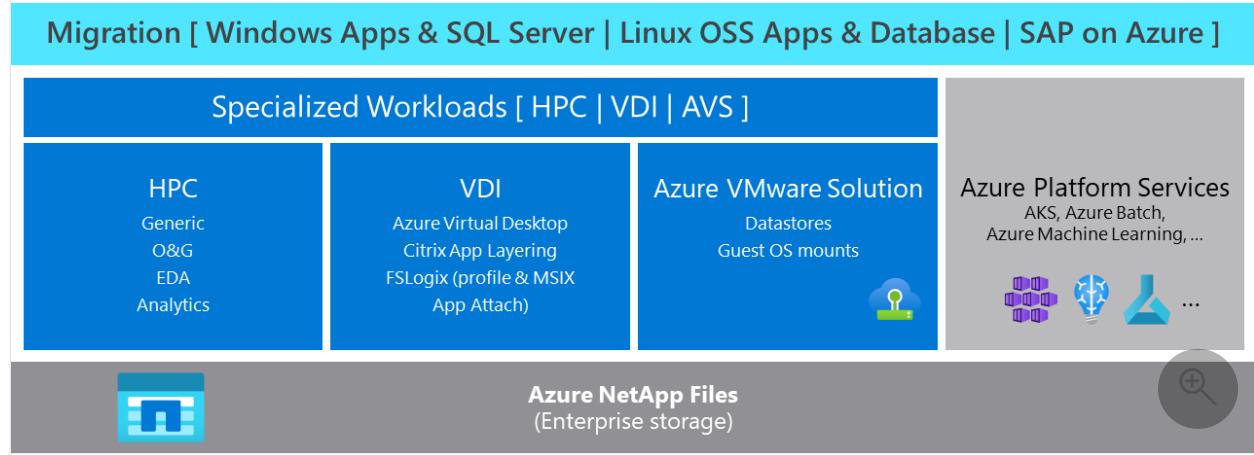
For businesses looking to migrate their applications and workloads to Azure, Azure NetApp Files provides a seamless experience for migrating Windows Apps and SQL server, Linux OSS Apps and Databases, and SAP on Azure. Azure NetApp Files' integration with Azure services makes the migration process easy, enabling users to move their workloads from on-premises to the cloud with minimal effort.

In addition to migration, Azure NetApp Files provides a platform for running specialized workloads in High-Performance Computing (HPC) like Analytics, Oil and Gas, and Electronic Design Automation (EDA). These specialized workloads require high-performance computing resources, and Azure NetApp Files' scalable and high-performance file storage solution provides the ideal platform for running these workloads in Azure. Azure NetApp Files also supports running Virtual Desktop Infrastructure (VDI) with Azure Virtual Desktop and Citrix, as well as Azure VMware Solution with guest OS mounts and datastores.

Azure NetApp Files' integration with Azure native services like Azure Kubernetes Service, Azure Batch, and Azure Machine Learning provides users with a seamless experience and enables them to leverage the full power of Azure's cloud-native services. This integration allows businesses to run their workloads in a scalable, secure, and highly performant environment, providing them with the confidence they need to run mission-critical workloads in the cloud.

The following diagram depicts the categorization of reference architectures, blueprints and solutions on this page as laid out in the above introduction:

Azure NetApp Files key use cases



In summary, Azure NetApp Files is a versatile and scalable storage service that provides an ideal platform for migrating various workload categories, running specialized workloads, and integrating with Azure native services. Azure NetApp Files' high-performance, security, and scalability features make it a reliable choice for businesses looking to run their applications and workloads in Azure.

Linux OSS Apps and Database solutions

This section provides references for solutions for Linux OSS applications and databases.

Linux OSS Apps

- [AIX UNIX on-premises to Azure Linux migration - Azure Example Scenarios](#)
- [Leverage Azure NetApp Files for R Studio workloads ↗](#)

Oracle

- [Oracle Database with Azure NetApp Files - Azure Example Scenarios](#)
- [Oracle VM images and their deployment on Microsoft Azure: Shared storage configuration options](#)
- [Oracle On Azure IaaS Recommended Practices For Success ↗](#)
- [Run Your Most Demanding Oracle Workloads in Azure without Sacrificing Performance or Scalability ↗](#)
- [Oracle database performance on Azure NetApp Files multiple volumes](#)
- [Oracle database performance on Azure NetApp Files single volumes](#)
- [Benefits of using Azure NetApp Files with Oracle Database](#)
- [Oracle Databases on Microsoft Azure Using Azure NetApp Files ↗](#)

Financial analytics and trading

- Host a Murex MX.3 workload on Azure

Product Lifecycle Management

- Use Teamcenter PLM with Azure NetApp Files
- Siemens Teamcenter baseline architecture
- Migrate Product Lifecycle Management (PLM) to Azure

Machine Learning

- Cloudera Machine Learning ↗
- Distributed ML Training for Lane Detection, powered by NVIDIA and Azure NetApp Files ↗
- Distributed ML Training for Click-Through Rate Prediction with NVIDIA, Dask and Azure NetApp Files ↗

Education

- Moodle deployment with Azure NetApp Files - Azure Example Scenarios
- Moodle on Azure NetApp Files NFS storage ↗

Mainframe refactor

- General mainframe refactor to Azure - Azure Example Scenarios
- Refactor mainframe applications with Advanced - Azure Example Scenarios
- Refactor mainframe applications with Astadia – Azure Example Scenarios
- Refactor mainframe computer systems that run Adabas & Natural - Azure Example Scenarios
- Refactor IBM z/OS mainframe coupling facility (CF) to Azure - Azure Example Scenarios
- Refactor mainframe applications to Azure with Raincode compilers - Azure Example Scenarios

Windows Apps and SQL Server solutions

This section provides references for Windows applications and SQL Server solutions.

File sharing and Global File Caching

- Enterprise file shares with disaster recovery - Azure Example Scenarios

- Disaster Recovery for Enterprise File Shares with Azure NetApp Files and DFS Namespaces ↗
- Build Your Own Azure NFS? Wrestling Linux File Shares into Cloud ↗
- Globally Distributed Enterprise File Sharing with Azure NetApp Files and NetApp Global File Cache ↗
- Cloud Compliance for Azure NetApp Files ↗

SQL Server

- SQL Server on Azure Virtual Machines with Azure NetApp Files - Azure Example Scenarios
- SQL Server on Azure Deployment Guide Using Azure NetApp Files ↗
- Benefits of using Azure NetApp Files for SQL Server deployment
- Managing SQL Server 2022 T-SQL snapshot backup with Azure NetApp Files snapshots ↗
- Deploy SQL Server Over SMB with Azure NetApp Files ↗
- Deploy SQL Server Always-On Failover Cluster over SMB with Azure NetApp Files ↗
- Deploy Always-On Availability Groups with Azure NetApp Files ↗

SAP on Azure solutions

This section provides references to SAP on Azure solutions.

Generic SAP and SAP Netweaver

- Run SAP NetWeaver in Windows on Azure - Azure Architecture Center
- High availability for SAP NetWeaver on Azure VMs on SUSE Linux Enterprise Server with Azure NetApp Files for SAP applications
- High availability for SAP NetWeaver on Azure VMs on Red Hat Enterprise Linux with Azure NetApp Files for SAP applications
- High availability for SAP NetWeaver on Azure VMs on Windows with Azure NetApp Files (SMB) for SAP applications
- Using Windows DFS-N to support flexible SAPMNT share creation for SMB-based file share
- High availability for SAP NetWeaver on Azure VMs on Red Hat Enterprise Linux for SAP applications multi-SID guide

SAP HANA

- SAP HANA for Linux VMs in scale-up systems - Azure Architecture Center

- SAP S/4HANA in Linux on Azure - Azure Architecture Center
- Run SAP BW/4HANA with Linux VMs - Azure Architecture Center
- SAP HANA Azure virtual machine storage configurations
- SAP on Azure NetApp Files Sizing Best Practices ↗
- Optimize HANA deployments with Azure NetApp Files application volume group for SAP HANA ↗
- Using Azure NetApp Files AVG for SAP HANA to deploy HANA with multiple partitions (MP) ↗
- NFS v4.1 volumes on Azure NetApp Files for SAP HANA
- High availability of SAP HANA Scale-up with Azure NetApp Files on Red Hat Enterprise Linux
- SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on SUSE Linux Enterprise Server
- SAP HANA scale-out with standby node on Azure VMs with Azure NetApp Files on Red Hat Enterprise Linux
- SAP HANA scale-out with HSR and Pacemaker on RHEL - Azure Virtual Machines
- Implementing Azure NetApp Files with Kerberos for SAP HANA ↗
- Azure Application Consistent Snapshot tool (AzAcSnap)
- Protecting HANA databases configured with HSR on Azure NetApp Files with AzAcSnap ↗
- Manual Recovery Guide for SAP HANA on Azure VMs from Azure NetApp Files snapshot with AzAcSnap ↗
- SAP HANA on Azure NetApp Files - Data protection with BlueXP backup and recovery ↗
- SAP HANA on Azure NetApp Files – System refresh & cloning operations with BlueXP backup and recovery ↗
- Azure NetApp Files Backup for SAP Solutions ↗
- SAP HANA Disaster Recovery with Azure NetApp Files ↗

SAP AnyDB

- SAP System on Oracle Database on Azure - Azure Architecture Center
- Oracle Azure Virtual Machines DBMS deployment for SAP workload - Azure Virtual Machines
- Deploy SAP AnyDB (Oracle 19c) with Azure NetApp Files ↗
- Manual Recovery Guide for SAP Oracle 19c on Azure VMs from Azure NetApp Files snapshot with AzAcSnap ↗
- SAP Oracle 19c System Refresh Guide on Azure VMs using Azure NetApp Files Snapshots with AzAcSnap ↗

- [IBM Db2 Azure Virtual Machines DBMS deployment for SAP workload using Azure NetApp Files](#)
- [DB2 Installation Guide on Azure NetApp Files ↗](#)
- [Manual Recovery Guide for SAP DB2 on Azure VMs from Azure NetApp Files snapshot with AzAcSnap ↗](#)
- [SAP ASE 16.0 on Azure NetApp Files for SAP Workloads on SLES15 ↗](#)
- [SAP Netweaver 7.5 with MaxDB 7.9 on Azure using Azure NetApp Files ↗](#)

SAP IQ-NLS

- [Deploy SAP IQ-NLS HA Solution using Azure NetApp Files on SUSE Linux Enterprise Server ↗](#)
- [How to manage SAP IQ License in HA Scenario ↗](#)

SAP tech community and blog posts

- [Architectural Decisions to maximize ANF investment in HANA N+M Scale-Out Architecture - Part 1 ↗](#)
- [Architectural Decisions to maximize ANF investment in HANA N+M Scale-Out Architecture - Part 2 ↗](#)
- [Architectural Decisions to maximize ANF investment in HANA N+M Scale-Out Architecture - Part 3 ↗](#)
- [SAP Landscape sizing and volume consolidation with Azure NetApp Files ↗](#)
- [Gain first hands-on experience with the new automated S/4HANA deployment in Microsoft Azure ↗](#)

Azure VMware Solution solutions

- [Attach Azure NetApp Files datastores to Azure VMware Solution hosts](#)
- [Attach Azure NetApp Files to Azure VMware Solution VMs - Guest OS Mounts](#)
- [Deploy disaster recovery using JetStream DR software](#)
- [Disaster Recovery with Azure NetApp Files, JetStream DR and AVS \(Azure VMware Solution\) ↗ - Jetstream](#)
- [Enable App Volume Replication for Horizon VDI on Azure VMware Solution using Azure NetApp Files ↗](#)
- [Disaster Recovery using cross-region replication with Azure NetApp Files datastores for AVS ↗](#)
- [Protecting Azure VMware Solution VMs and datastores on Azure NetApp Files with Cloud Backup for VMs ↗](#)

Virtual Desktop Infrastructure solutions

This section provides references for Virtual Desktop infrastructure solutions.

Azure Virtual Desktop

- Benefits of using Azure NetApp Files with Azure Virtual Desktop
- Storage options for FSLogix profile containers in Azure Virtual Desktop
- Create an FSLogix profile container for a host pool using Azure NetApp Files
- Azure Virtual Desktop at enterprise scale
- Microsoft FSLogix for the enterprise - Azure NetApp Files best practices
- Enhanced Performance and Scalability: Azure AD-joined Session Hosts with Azure NetApp Files ↗
- Setting up Azure NetApp Files for MSIX App Attach ↗
- Multiple forests with AD DS and Azure AD – Azure Example Scenarios
- Multiregion Business Continuity and Disaster Recovery (BCDR) for Azure Virtual Desktop – Azure Example Scenarios
- Deploy Esri ArcGIS Pro in Azure Virtual Desktop – Azure Example Scenarios

Citrix

- Citrix Profile Management with Azure NetApp Files Best Practices Guide ↗

HPC solutions

This section provides references for High Performance Computing (HPC) solutions.

Generic HPC

- Azure HPC OnDemand Platform ↗
- Azure NetApp Files: Getting the most out of your cloud storage ↗
- Run MPI workloads with Azure Batch and Azure NetApp Files ↗
- Azure Cycle Cloud: CycleCloud HPC environments on Azure NetApp Files

Oil and gas

- High performance computing (HPC): Oil and gas in Azure ↗
- Run reservoir simulation software on Azure

Electronic design automation (EDA)

- [EDA workloads on Azure NetApp Files - Performance Best Practice ↗](#)
- [Benefits of using Azure NetApp Files for electronic design automation](#)
- [Azure CycleCloud: EDA HPC Lab with Azure NetApp Files ↗](#)
- [Azure for the semiconductor industry ↗](#)

Analytics

- [SAS on Azure architecture guide - Azure Architecture Center | Azure NetApp Files](#)
- [Deploy SAS Grid 9.4 on Azure NetApp Files](#)
- [Best Practices for Using Microsoft Azure with SAS® ↗](#)
- [Azure NetApp Files: A shared file system to use with SAS Grid on Microsoft Azure ↗](#)
- [Azure NetApp Files: A shared file system to use with SAS Grid on MS Azure – RHEL8.3/nconnect UPDATE ↗](#)
- [Best Practices for Using Microsoft Azure with SAS® ↗](#)

Azure platform services solutions

This section provides solutions for Azure platform services.

Azure Kubernetes Services and Kubernetes

- [Astra: protect, recover, and manage your AKS workloads on Azure NetApp Files ↗](#)
- [Integrate Azure NetApp Files with Azure Kubernetes Service](#)
- [Azure NetApp Files SMB volumes for Azure Kubernetes Services with Astra Trident on Windows ↗](#)
- [Application data protection for AKS workloads on Azure NetApp Files - Azure Example Scenarios](#)
- [Disaster Recovery of AKS workloads with Astra Control Service and Azure NetApp Files ↗](#)
- [Protecting MongoDB on AKS/ANF with Astra Control Service using custom execution hooks ↗](#)
- [Comparing and Contrasting the AKS/ANF NFS subdir external provisioner with Astra Trident ↗](#)
- [Out-of-This-World Kubernetes performance on Azure with Azure NetApp Files ↗](#)
- [Azure NetApp Files + Trident = Dynamic and Persistent Storage for Kubernetes ↗](#)
- [Trident - Storage Orchestrator for Containers ↗](#)
- [Magento e-commerce platform in Azure Kubernetes Service \(AKS\)](#)

- Protecting Magento e-commerce platform in AKS against disasters with Astra Control Service ↗
- Protecting applications on private Azure Kubernetes Service clusters with Astra Control Service ↗
- Providing Disaster Recovery to CloudBees-Jenkins in AKS with Astra Control Service ↗
- Disaster protection for JFrog Artifactory in AKS with Astra Control Service and Azure NetApp Files ↗
- Develop and test easily on AKS with NetApp® Astra Control Service® and Azure NetApp Files ↗

Azure Machine Learning

- High-performance storage for AI Model Training tasks using Azure Machine Learning studio with Azure NetApp Files ↗
- How to use Azure Machine Learning with Azure NetApp Files ↗

Azure Red Hat Openshift

- Using Trident to Automate Azure NetApp Files from OpenShift ↗
- Deploy IBM Maximo Application Suite on Azure – Azure Example Scenarios

Azure Batch

- Run MPI workloads with Azure Batch and Azure NetApp Files ↗

Benefits of using Azure NetApp Files with Oracle Database

Article • 08/04/2022

Oracle Direct NFS (dNFS) makes it possible to drive higher performance than the operating system's own NFS driver. This article explains the technology and provides a performance comparison between dNFS and the traditional NFS client (Kernel NFS). It also shows the advantages and the ease of using dNFS with Azure NetApp Files.

ⓘ Important

For correct and optimal deployment of Oracle dNFS, follow the patching guidelines outlined [here](#).

How Oracle Direct NFS works

The following summary explains how Oracle Direct NFS works at a high level:

- Oracle Direct NFS bypasses the operating system buffer cache. Data is cached only once in the user space, eliminating the overhead of memory copies.
- The traditional NFS client uses a single network flow as shown below:

Proto	Local Address	Foreign Address	State
tcp	10.11.1.205:17372	172.16.50.52:2049	ESTABLISHED

Oracle Direct NFS further improves performance by load-balancing network traffic across multiple network flows. As tested and shown below, 650 distinct network connections were established dynamically by the Oracle Database:

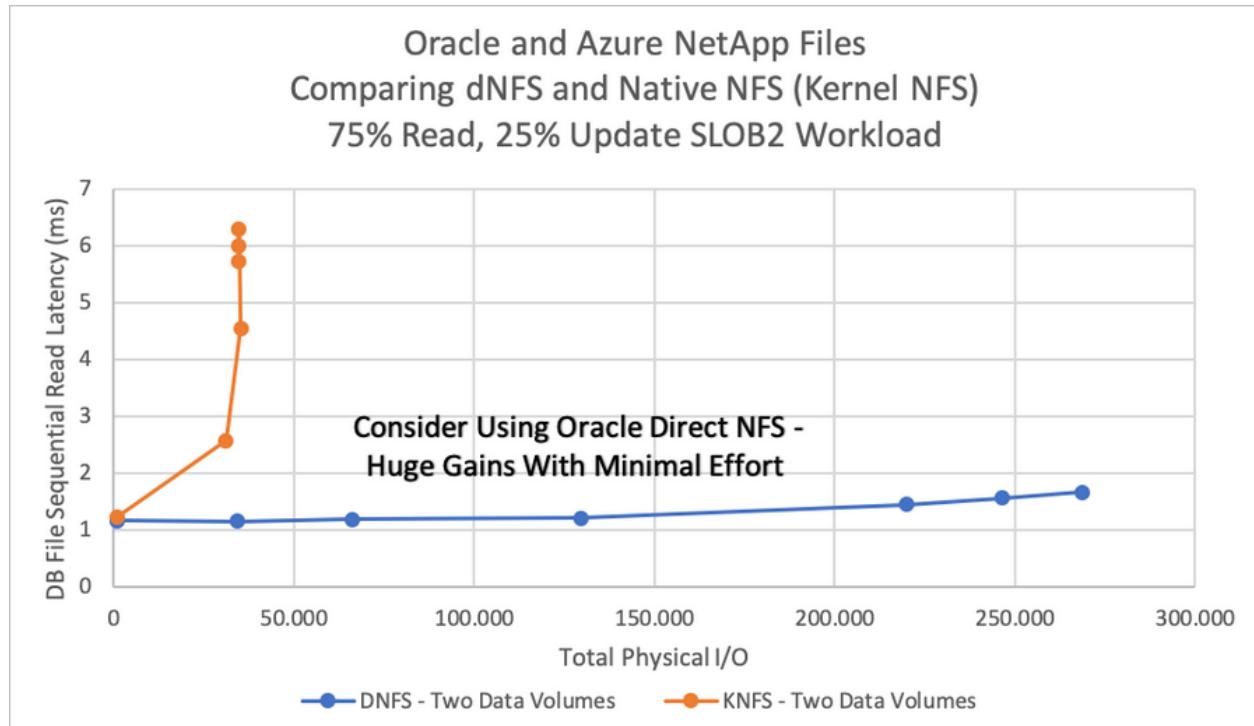
Proto	Local Address	Foreign Address	State
tcp	10.11.1.205:17372	172.16.50.52:2049	ESTABLISHED
tcp	10.11.1.205:61530	172.16.50.52:2049	ESTABLISHED
tcp	10.11.1.205:24526	172.16.50.52:2049	ESTABLISHED
tcp	10.11.1.205:29439	172.16.50.52:2049	ESTABLISHED
tcp	10.11.1.205:32993	172.16.50.52:2049	ESTABLISHED
tcp	10.11.1.205:50326	172.16.50.52:2049	ESTABLISHED

The [Oracle FAQ for Direct NFS](#) shows that Oracle dNFS is an optimized NFS client. It provides fast and scalable access to NFS storage that is located on NAS storage devices (accessible over TCP/IP). dNFS is built into the database kernel just like ASM, which is

used primarily with DAS or SAN storage. As such, *the guideline is to use dNFS when implementing NAS storage and use ASM when implementing SAN storage.*

dNFS is the default option in Oracle 18c.

dNFS is available starting with Oracle Database 11g. The diagram below compares dNFS with native NFS. When you use dNFS, an Oracle database that runs on an Azure virtual machine can drive more I/O than the native NFS client.



You can enable or disable dNFS by running two commands and restarting the database.

To enable:

```
cd $ORACLE_HOME/rdbms/lib ; make -f ins_rdbms.mk dnfs_on
```

To disable:

```
cd $ORACLE_HOME/rdbms/lib ; make -f ins_rdbms.mk dnfs_off
```

Azure NetApp Files combined with Oracle Direct NFS

You can enhance the performance of Oracle dNFS with the Azure NetApp Files service. The service gives you total control over your application performance. It can meet extremely demanding applications. The combination of Oracle dNFS with Azure NetApp Files provides great advantage to your workloads.

Next steps

- Solution architectures using Azure NetApp Files
- Overview of Oracle Applications and solutions on Azure

Benefits of using Azure NetApp Files for electronic design automation

Article • 01/03/2022

Time-to-market (TTM) is a critical consideration for the semiconductor and chip design industry. The industry has high bandwidth and low latency needs for storage. This article explains the solution Azure NetApp Files provides for meeting the industry's needs. It presents test scenarios running a standard industry benchmark for electronic design automation (EDA) using Azure NetApp Files.

Test scenario configurations

The tests involve three scenarios with the following configurations.

Scenario	Volumes	Clients SLES15 D16s_v3
One	1	1
Two	6	24
Three	12	24

The first scenario addresses how far a single volume can be driven.

The second and the third scenarios evaluate the limits of a single Azure NetApp Files endpoint. They investigate the potential benefits of I/O upper limits and latency.

Test scenario results

The following table summarizes the test scenarios results.

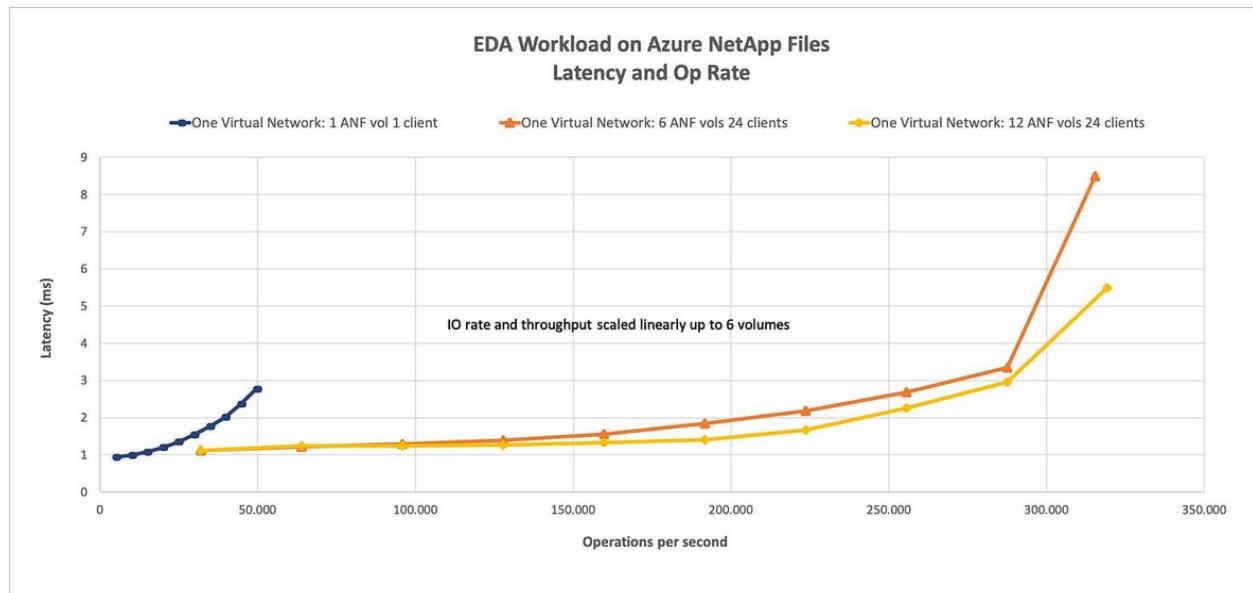
Scenario	I/O rate at 2 ms	I/O rate at the edge	Throughput at 2 ms	Throughput at the edge
1 volume	39,601	49,502	692 MiB/s	866 MiB/s
6 volumes	255,613	317,000	4,577 MiB/s	5,568 MiB/s
12 volumes	256,612	319,196	4,577 MiB/s	5,709 MiB/s

The single-volume scenario represents the basic application configuration. It's the baseline scenario for follow-on test scenarios.

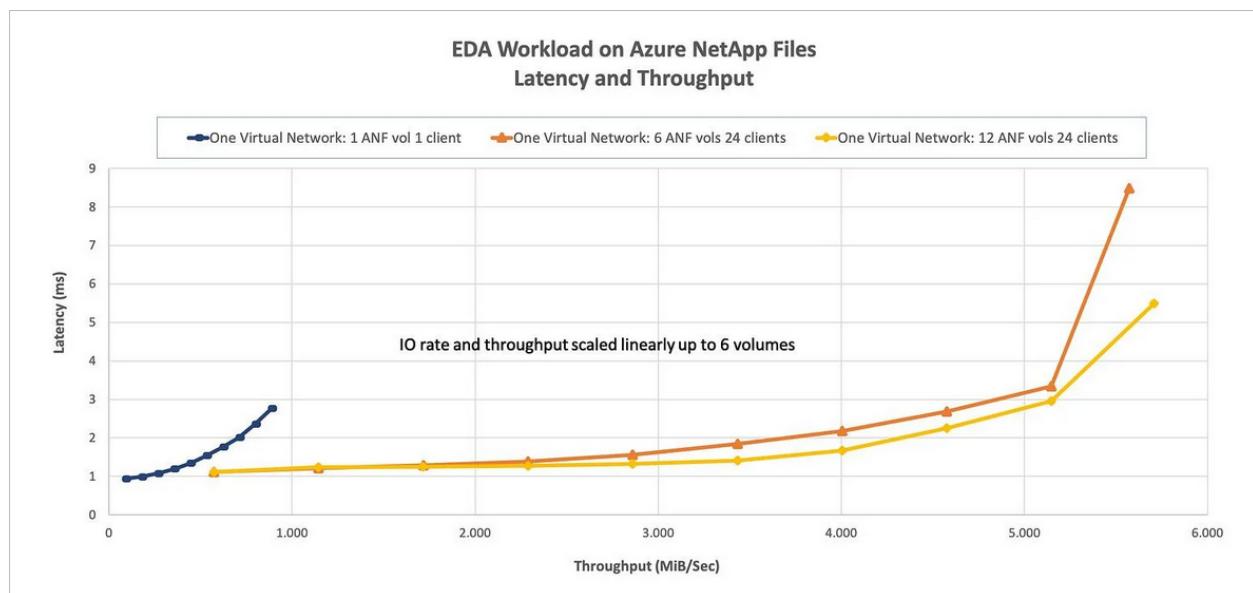
The six-volume scenario demonstrates a linear increase (600%) relative to the single-volume workload. All volumes within a single virtual network are accessed over a single IP address.

The 12-volume scenario demonstrates a general decrease in latency over the six-volume scenario. But it doesn't have a corresponding increase in achievable throughput.

The following graph illustrates the latency and operations rate for the EDA workload on Azure NetApp Files.



The following graph illustrates the latency and throughput for the EDA workload on Azure NetApp Files.



Layout of test scenarios

The table below summarizes the layout of the test scenarios.

Test scenario	Total number of directories	Total number of files
1 volume	88,000	880,000
6 volumes	568,000	5,680,000
12 volumes	568,000	5,680,000

The complete workload is a mixture of concurrently running functional and physical phases. It represents a typical flow from one set of EDA tools to another.

The functional phase consists of initial specifications and a logical design. The physical phase takes place when the logical design is converted to a physical chip. During the sign-off and tape-out phases, final checks are completed, and the design is delivered to a foundry for manufacturing.

The functional phase includes a mixture of sequential and random read and write I/O. The functional phase is metadata intensive, like file stat and access calls. Although metadata operations are effectively without size, the read and write operations range between less than 1 K and 16 K. Most reads are between 4 K and 16 K. Most writes are 4 K or less. The physical phase is composed of sequential read and write operations entirely, with a mixture of 32 K and 64 K OP sizes.

In the graphs above, most of the throughput comes from the sequential physical phase of workload. The I/O comes from the small random and metadata-intensive functional phase. Both phases happen in parallel.

In conclusion, you can pair Azure compute with Azure NetApp Files for EDA design to get scalable bandwidth.

Next steps

- [Solution architectures using Azure NetApp Files](#)

Benefits of using Azure NetApp Files with Azure Virtual Desktop

Article • 01/03/2022

This article provides best practice guidance on deploying Azure Virtual Desktop with Azure NetApp Files.

Azure NetApp Files is a highly performant file storage service from Azure. It can provide up to 450,000 IOPS and sub-millisecond latency, capable of supporting extremely large scale of Azure Virtual Desktop deployments. You can adjust the bandwidth and change the service level of your Azure NetApp Files volumes on demand almost instantaneously without pausing IO while retaining data plane access. This capability allows you to easily optimize your Azure Virtual Desktop deployment scale for cost. You can also create space-efficient, point-in-time volume snapshots without impacting volume performance. This capability makes it possible for you to roll back individual [FSLogix user profile containers](#) via a copy from the `~snapshot` directory, or to instantaneously roll back the entire volume at once via the volume revert capability. With up to 255 (rotational) snapshots in place to protect a volume from data loss or corruption, administrators have many chances to undo what has been done.

Sample blueprints

The following sample blueprints show the integration of Azure Virtual Desktop with Azure NetApp Files. In a pooled desktop scenario, users are directed to the best available session (the [breadth-first mode](#)) host in the pool, using [multi-session virtual machines](#). On the other hand, personal desktops are reserved for scenarios in which each user has their own virtual machine.

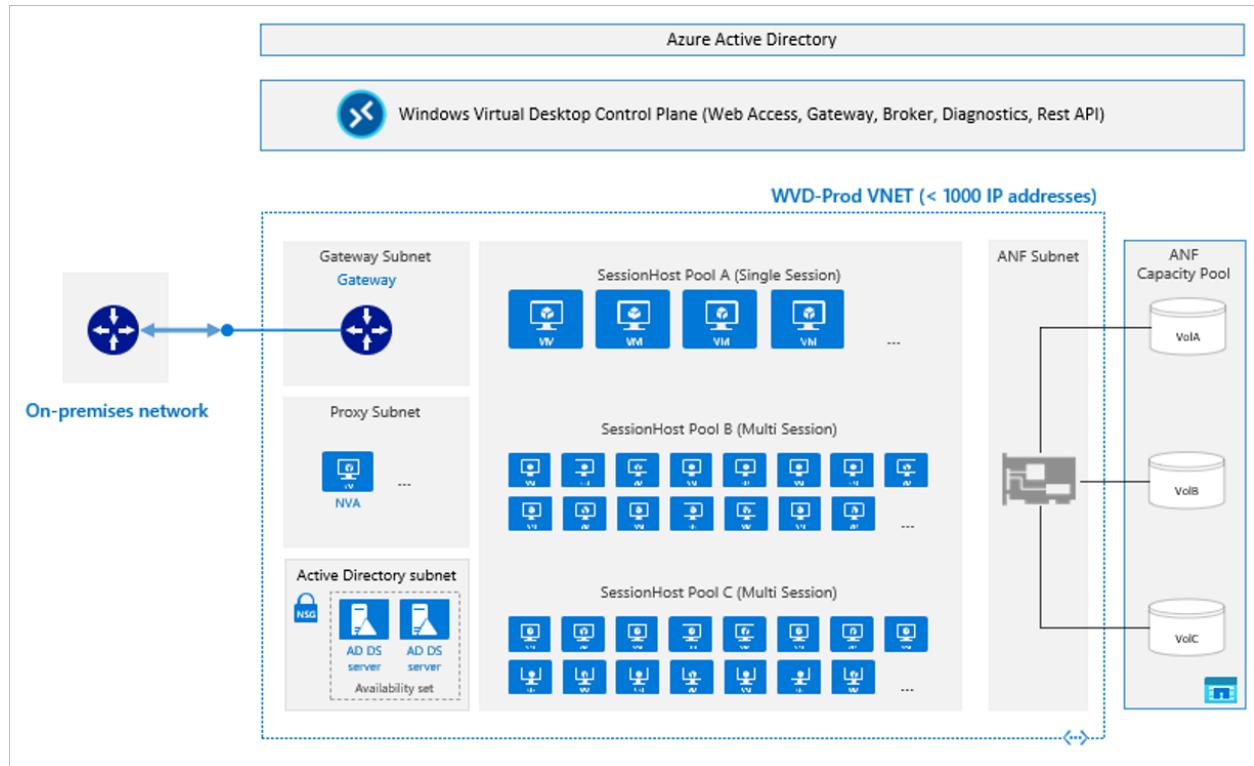
Pooled desktop scenario

For the pooled scenario, the Azure Virtual Desktop team [recommends](#) the following guidance by user count to vCPU. Note that no virtual machine size is specified in this recommendation.

Workload type	Light	Medium	Heavy
Users per vCPU	6	4	2

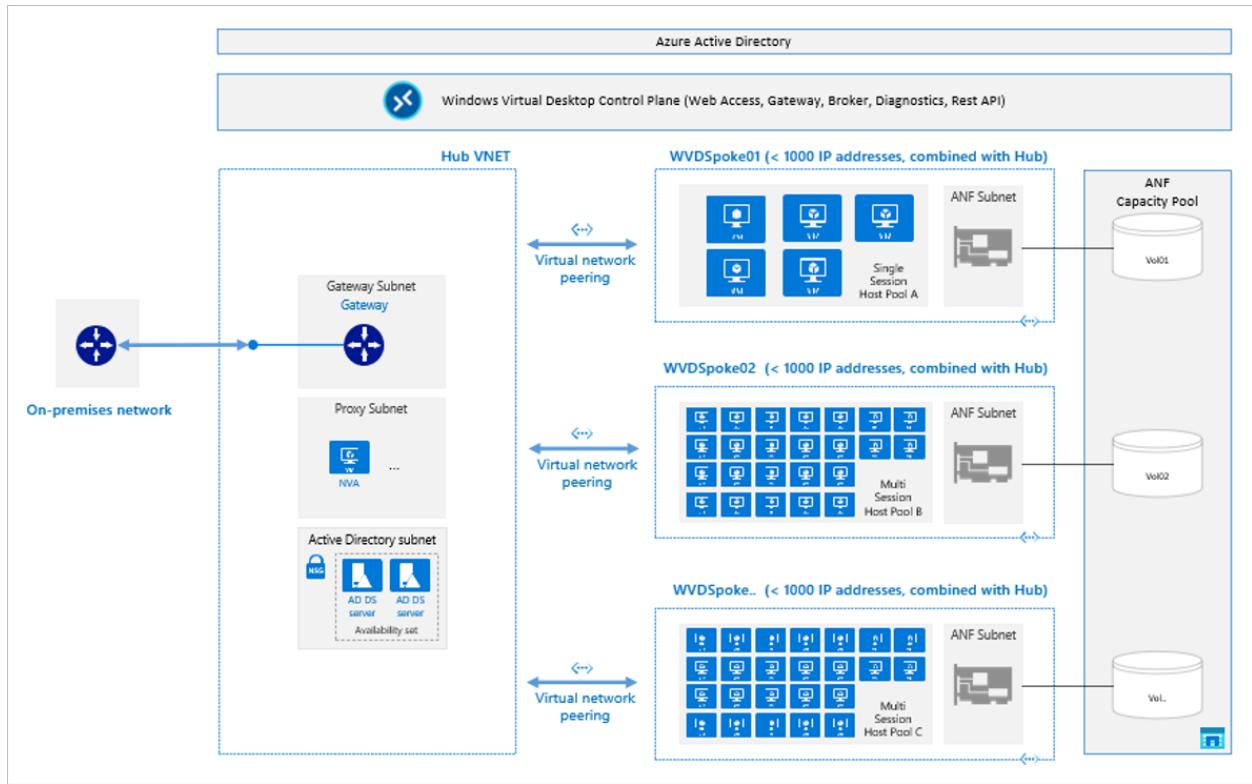
This recommendation is confirmed by a 500-user LoginVSI test, logging approximately 62 “knowledge / medium users” onto each D16as_V4 virtual machine.

As an example, at 62 users per D16as_V4 virtual machine, Azure NetApp Files can easily support 60,000 users per environment. Testing to evaluate the upper limit of the D32as_v4 virtual machine is ongoing. If the Azure Virtual Desktop user per vCPU recommendation holds true for the D32as_v4, more than 120,000 users would fit within 1,000 virtual machines before broaching [the 1,000 IP VNet limit](#), as shown in the following figure.



Personal desktop scenario

In a personal desktop scenario, the following figure shows the general-purpose architectural recommendation. Users are mapped to specific desktop pods and each pod has just under 1,000 virtual machines, leaving room for IP addresses propagating from the management VNet. Azure NetApp Files can easily handle 900+ personal desktops per single-session host pool VNet, with the actual number of virtual machines being equal to 1,000 minus the number of management hosts found in the Hub VNet. If more personal desktops are needed, it's easy to add more pods (host pools and virtual networks), as shown in the following figure.



When building a POD based architecture like this, assigning users to the correct pod upon login is of importance to assure users will always find their user profiles.

Next steps

- Solution architectures using Azure NetApp Files

Benefits of using Azure NetApp Files for SQL Server deployment

Article • 01/03/2022

Azure NetApp Files reduces SQL Server total cost of ownership (TCO) as compared to block storage solutions. With block storage, virtual machines have imposed limits on I/O and bandwidth for disk operations. Only network bandwidth limits are applied against Azure NetApp Files, and on egress only at that. In other words, no VM level I/O limits are applied to Azure NetApp Files. Without these I/O limits, SQL Server running on smaller virtual machines connected to Azure NetApp Files can perform as well as SQL Server running on much larger virtual machines. Sizing instances down as such reduces the compute cost to 25% of the former price tag. *You can reduce compute costs with Azure NetApp Files.*

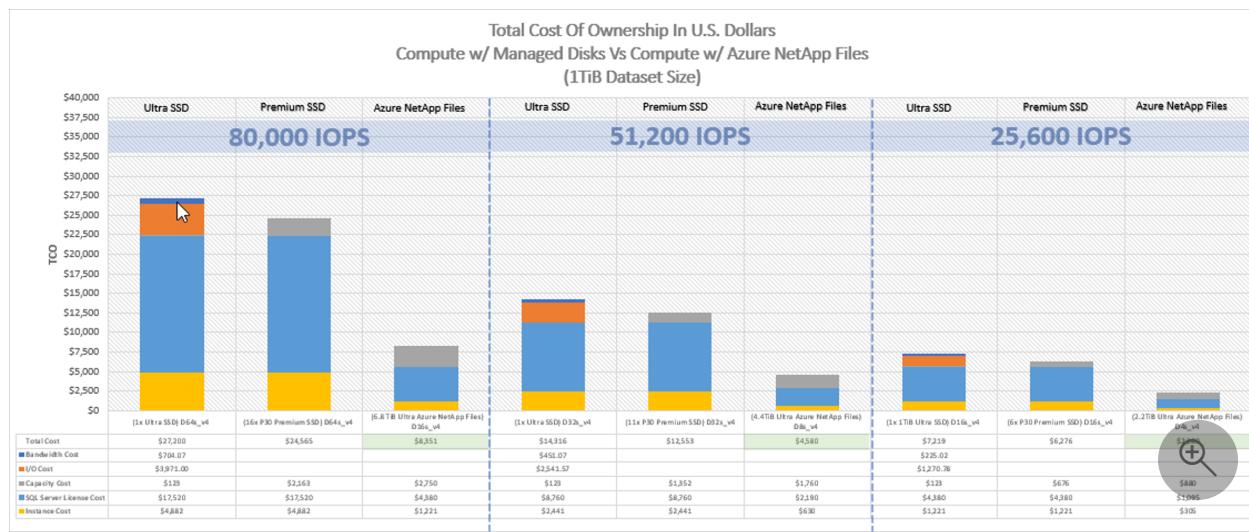
Compute costs, however, are small compared to SQL Server license costs. Microsoft SQL Server [licensing](#) is tied to physical core count. As such, decreasing instance size introduces an even larger cost saving for software licensing. *You can reduce software license costs with Azure NetApp Files.*

This article shows a detailed cost analysis and performance benefits about using Azure NetApp Files for SQL Server deployment. Not only do smaller instances have sufficient CPU to do the database work only possible with block on larger instances, *in many cases, the smaller instances are even more performant than their larger, disk-based counterparts because of Azure NetApp Files.*

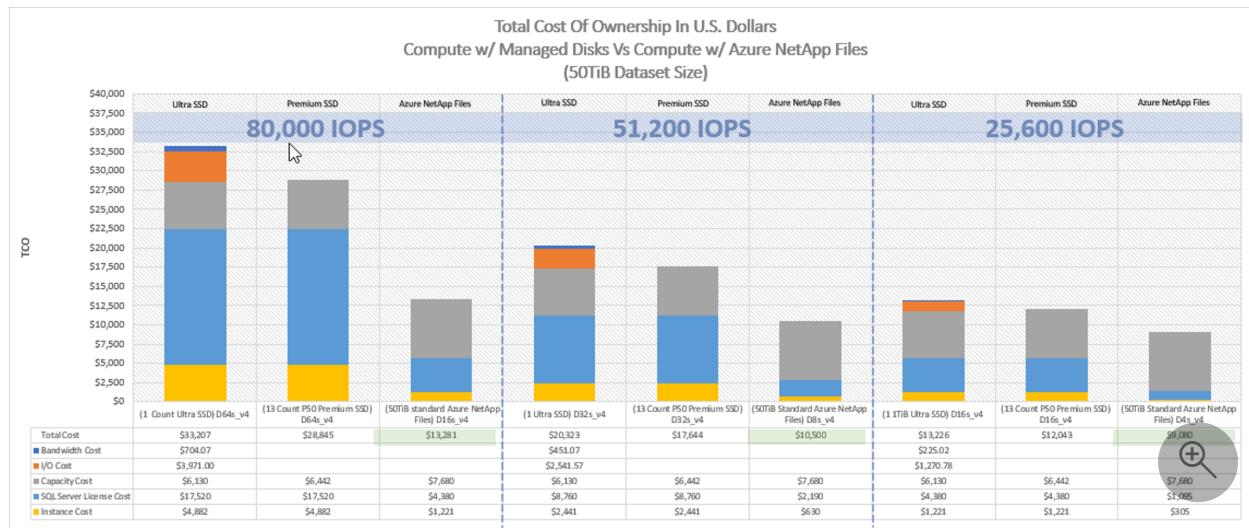
Detailed cost analysis

The two sets of graphics in this section show the TCO example. The number and type of managed disks, the Azure NetApp Files service level, and the capacity for each scenario have been selected to achieve the best price-capacity-performance. Each graphic is made up of grouped machines (D16 with Azure NetApp Files, compared to D64 with managed disk by example), and prices are broken down for each machine type.

The first set of graphic shows the overall cost of the solution using a 1-TiB database size, comparing the D16s_v4 to the D64, the D8 to the D32, and the D4 to the D16. The projected IOPs for each configuration are indicated by a green or yellow line and corresponds to the right-hand side Y axis.



The second set of graphic shows the overall cost using a 50-TiB database. The comparisons are otherwise the same – D16 compared with Azure NetApp Files versus D64 with block by example.



Performance, and lots of it

To deliver on the significant cost reduction assertion requires lots of performance - the largest instances in the general Azure inventory support 80,000 disk IOPS by example. A single Azure NetApp Files volume can achieve 80,000 database IOPS, and instances such as the D16 are able to consume the same. The D16, normally capable of 25,600 disk IOPS, is 25% the size of the D64. The D64s_v4 is capable of 80,000 disk IOPS, and as such, presents an excellent upper level comparison point.

The D16s_v4 can drive an Azure NetApp Files volume to 80,000 database IOPS. As proven by the SQL Storage Benchmark (SSB) benchmarking tool, the D16 instance achieved a workload 125% greater than that achievable to disk from the D64 instance. See the [SSB testing tool](#) section for details about the tool.

Using a 1-TiB working set size and an 80% read, 20% update SQL Server workload, performance capabilities of most the instances in the D instance class were measured; most, not all, as the D2 and D64 instances themselves were excluded from testing. The former was left out as it doesn't support accelerated networking, and the latter because it's the comparison point. See the following graph to understand the limits of D4s_v4, D8s_v4, D16s_v4, and D32s_v4, respectively. Managed disk storage tests are not shown in the graph. Comparison values are drawn directly from the [Azure Virtual Machine limits table](#) for the D class instance type.

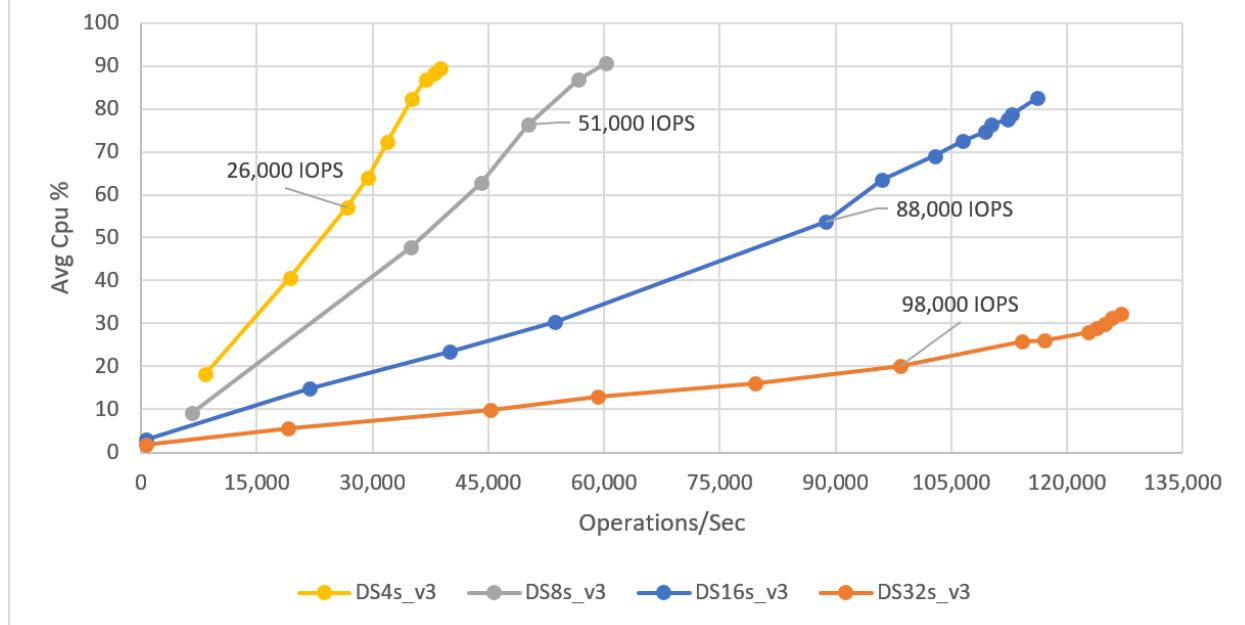
With Azure NetApp Files, each of the instances in the D class can meet or exceed the disk performance capabilities of instances two times larger. *You can reduce software license costs significantly with Azure NetApp Files.*

- The D4 at 75% CPU utilization matched the disk capabilities of the D16.
 - The D16 is rate limited at 25,600 disk IOPS.
- The D8 at 75% CPU utilization matched the disk capabilities of the D32.
 - The D32 is rate limited at 51,200 disk IOPS.
- The D16 at 55% CPU utilization matched the disk capabilities of the D64.
 - The D64 is rate limited at 80,000 disk IOPS.
- The D32 at 15% CPU utilization matched the disk capabilities of the D64 as well.
 - The D64 as stated above is rate limited at 80,000 disk IOPS.

S3B CPU limits test – Performance versus processing power

The following diagram summarizes the S3B CPU limits test:

Single Instance Sql Server over Azure Netapp Files 80% Reads CPU% Comparison



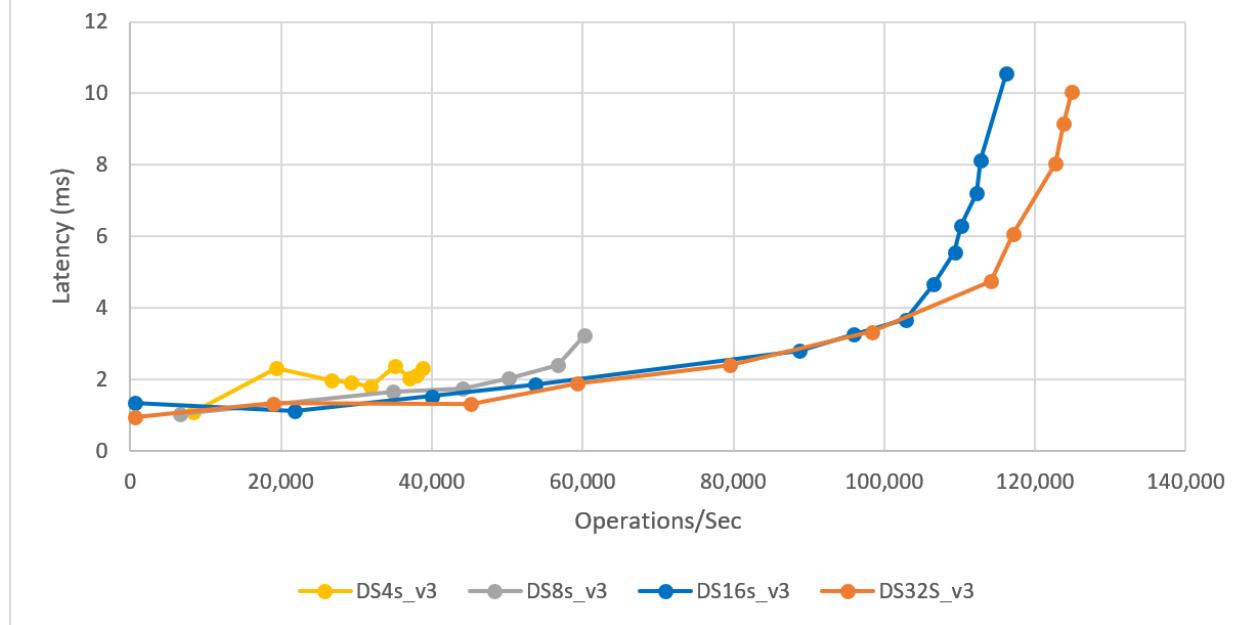
Scalability is only part of the story. The other part is latency. It's one thing for smaller virtual machines to have the ability to drive much higher I/O rates, it's another thing to do so with low single-digit latencies as shown below.

- The D4 drove 26,000 IOPS against Azure NetApp Files at 2.3-ms latency.
- The D8 drove 51,000 IOPS against Azure NetApp Files at 2.0-ms latency.
- The D16 drove 88,000 IOPS against Azure NetApp Files at 2.8-ms latency.
- The D32 drove 80,000 IOPS against Azure NetApp Files at 2.4-ms latency.

S3B per instance type latency results

The following diagram shows the latency for single-instance SQL Server over Azure NetApp Files:

Single Instance Sql Server over Azure Netapp Files 80% Reads Ops Comparison



SSB testing tool

The [TPC-E](#) benchmarking tool, by design, stresses *compute* rather than *storage*. The test results shown in this section are based on a stress testing tool named SQL Storage Benchmark (SSB). The SQL Server Storage Benchmark can drive massive-scale SQL execution against a SQL Server database to simulate an OLTP workload, similar to the [SLOB2 Oracle benchmarking tool](#).

The SSB tool generates a SELECT and UPDATE driven workload issuing the said statements directly to the SQL Server database running within the Azure virtual machine. For this project, the SSB workloads ramped from 1 to 100 SQL Server users, with 10 or 12 intermediate points at 15 minutes per user count. All performance metrics from these runs were from the point of view of perfmon, for repeatability SSB ran three times per scenario.

The tests themselves were configured as 80% SELECT and 20% UPDATE statement, thus 90% random read. The database itself, which SSB created, was 1000 GB in size. It's comprised of 15 user tables and 9,000,000 rows per user table and 8192 bytes per row.

The SSB benchmark is an open-source tool. It's freely available at the [SQL Storage Benchmark GitHub page](#).

In summary

With Azure NetApp Files, you can increase SQL server performance while reducing your total cost of ownership significantly.

Next Steps

- [Create an SMB volume for Azure NetApp Files](#)
- [Solution architectures using Azure NetApp Files – SQL Server](#)

Understand Azure NetApp Files application volume group for SAP HANA

Article • 02/24/2023

This article helps you understand the use cases and key features of Azure NetApp Files application volume group for SAP HANA.

Application volume group for SAP HANA enables you to deploy all volumes required to install and operate an SAP HANA database according to best practices. Instead of individually creating the required SAP HANA volumes (including data, log, shared, log-backup, and data-backup volumes), application volume group for SAP HANA creates these volumes in a single "atomic" call. The atomic call ensures that either all volumes or no volumes at all are created.

Application volume group for SAP HANA provides technical improvements to simplify and standardize the process to help you streamline volume deployments for SAP HANA. As a result, you can focus on your application demands instead of managing technical settings such as individual QoS or sizes for volumes.

Key features

Application volume group for SAP HANA is supported for all regions. It provides the following key features:

- Supporting SAP HANA configurations for both single and multiple host setups, including:
 - Volumes for a single or primary SAP HANA database
 - Volumes for an SAP HANA System Replication (HSR) secondary system
 - Volumes for a disaster recovery (DR) scenario using [cross-region replication](#)
- Creating the following volumes:
 - SAP HANA data volumes (one for each database host)
 - SAP HANA log volumes (one for each database host)
 - SAP HANA shared volumes (for the first SAP HANA host only)
 - Log-backup volumes (optional)
 - File-based data-backup volumes (optional)
- Creating volumes in a [manual QoS capacity pool](#). The volume size and the required performance (in MiB/s) are proposed based on user input for the memory size of

the database.

- The application volume group GUI and Azure Resource Manager (ARM) template provide best practices to simplify sizing management and volume creation. For example:
 - Proposing volume naming convention based on SAP System ID (SID) and volume type
 - Calculating the size and performance based on memory size

Application volume group for SAP HANA helps you simplify the deployment process and increase the storage performance for SAP HANA workloads. Some of the new features are as follows:

- Use of proximity placement group (PPG) instead of manual pinning.
 - You will anchor the SAP HANA VMs using a PPG to guarantee lowest possible latency. This PPG will be used to enforce that the data, log, and shared volumes are created in the close proximity to the SAP HANA VMs. See [Best practices about Proximity Placement Groups](#) for detail.
- Creation of separate storage endpoints (with different IP addresses) for data and log volumes.
 - This deployment method provides better performance and throughput for the SAP HANA database.

Next steps

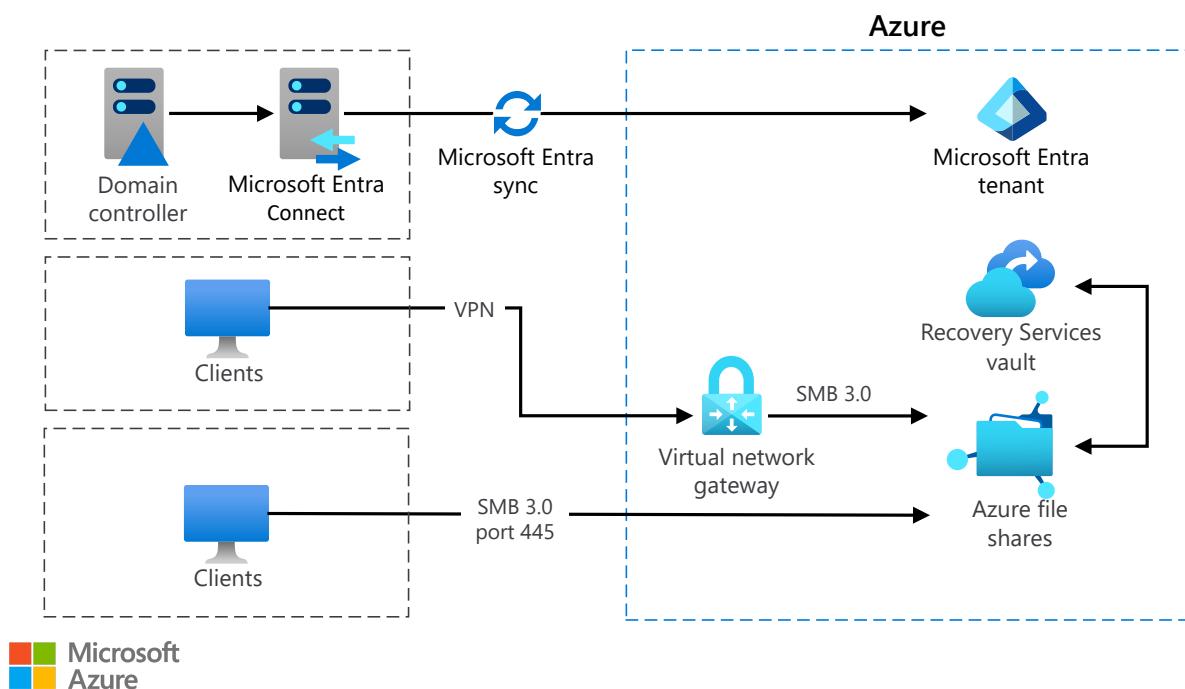
- Requirements and considerations for application volume group for SAP HANA
- Deploy the first SAP HANA host using application volume group for SAP HANA
- Add hosts to a multiple-host SAP HANA system using application volume group for SAP HANA
- Add volumes for an SAP HANA system as a secondary database in HSR
- Add volumes for an SAP HANA system as a DR system using cross-region replication
- Manage volumes in an application volume group
- Delete an application volume group
- Application volume group FAQs
- Troubleshoot application volume group errors

Use Azure file shares in a hybrid environment

Microsoft Entra ID Azure Files

This architecture shows how to include Azure file shares in your hybrid environment. Azure file shares are used as serverless file shares. By integrating them with Active Directory Directory Services (AD DS), you can control and limit access to AD DS users. Azure file shares then can replace traditional file servers.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

The architecture consists of the following components:

- **Microsoft Entra tenant.** This component is an instance of Microsoft Entra that's created by your organization. It acts as a directory service for cloud applications, by storing objects that are copied from the on-premises Active Directory. It also provides identity services when accessing Azure file shares.

- **AD DS server.** This component is an on-premises directory and identity service. The AD DS directory is synchronized with Microsoft Entra ID to enable it to authenticate on-premises users.
- **Microsoft Entra Connect Sync server.** This component is an on-premises server that runs the Microsoft Entra Connect Sync service. This service synchronizes information held in the on-premises Active Directory to Microsoft Entra ID.
- **Virtual network gateway.** This optional component is used to send encrypted traffic between an Azure Virtual Network and an on-premises location over the internet.
- **Azure file shares.** Azure file shares provide storage for files and folders that you can access over Server Message Block (SMB), Network File System (NFS), and Hypertext Transfer Protocol (HTTP) protocols. File shares are deployed into Azure storage accounts.
- **Recovery Services Vault.** This optional component provides Azure file shares backup.
- **Clients.** These components are AD DS member computers, from which users can access Azure file shares.

Components

Key technologies used to implement this architecture:

- [Microsoft Entra ID](#) is an enterprise identity service that provides single sign-on, multifactor authentication, and conditional access.
- [Azure Files](#) offers fully managed file shares in the cloud that are accessible by using the industry standard protocols.
- [VPN Gateway](#) VPN Gateway sends encrypted traffic between an Azure virtual network and an on-premises location over the public Internet.

Scenario details

Potential use cases

Typical uses for this architecture include:

- **Replace or supplement on-premises file servers.** Azure Files can completely replace or supplement traditional on-premises file servers or network-attached storage devices. With Azure file shares and AD DS authentication, you can migrate data to Azure Files. This migration can take advantage of high availability and scalability while minimizing client changes.

- **Lift and shift.** Azure Files makes it easy to "lift and shift" applications that expect a file share to store application or user data to the cloud.
- **Backup and disaster recovery.** You can use Azure Files as storage for backups or for disaster recovery to improve business continuity. You can use Azure Files to back up your data from existing file servers while preserving configured Windows discretionary access control lists. Data that's stored on Azure file shares isn't affected by disasters that might affect on-premises locations.
- **Azure File Sync.** With Azure File Sync, Azure file shares can replicate to Windows Server, either on-premises or in the cloud. This replication improves performance and distributes caching of data to where it's being used.

Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a specific requirement that overrides them.

Use general-purpose v2 (GPv2) or FileStorage storage accounts for Azure file shares

You can create an Azure file share in various storage accounts. Although general-purpose v1 (GPv1) and classic storage accounts can contain Azure file shares, most new features of Azure Files are available only in GPv2 and FileStorage storage accounts. While an Azure file share stores GPv2 storage accounts data on hard disk drive-based (HDD-based) hardware, it stores FileStorage storage accounts data on solid-state drive-based (SSD-based) hardware. For more information, see [Create an Azure file share](#).

Create Azure file shares in storage accounts that contain only Azure file shares

Storage accounts allow you to use different storage services in the same storage account. These storage services include Azure file shares, blob containers, and tables. All storage services in a single storage account share the same storage account limits. Mixing storage services in the same storage account make it more difficult to troubleshoot performance issues.

 **Note**

Deploy each Azure file share in its own separate storage account, if possible. If multiple Azure file shares are deployed into the same storage account, they all share the storage account limits.

Use premium file shares for workloads that require high throughput

Premium file shares are deployed to FileStorage storage accounts and are stored on solid-state drive-based (SSD-based) hardware. This setup makes them suitable for storing and accessing data that requires consistent performance, high throughput, and low latency. (For example, these premium file shares work well with databases.) You can store other workloads that are less sensitive to performance variability on standard file shares. These workload types include general-purpose file shares and dev/test environments. For more information, see [How to create an Azure file share](#).

Always require encryption when accessing Azure file shares

Always use encryption in transit when accessing data in Azure file shares. (Encryption in transit is enabled by default.) Azure Files will only allow the connection if it's made with a protocol that uses encryption, such as SMB 3.0. Clients that don't support SMB 3.0 will be unable to mount the Azure file share if encryption in transit is required.

Use VPN if port that SMB uses (port 445) is blocked

Many internet service providers block Transmission Control Protocol (TCP) port 445, which is used to access Azure file shares. If unblocking **TCP port 445** isn't an option, you can access Azure file shares over an ExpressRoute or virtual private network (VPN) connection (site-to-site or point-to-site) to avoid traffic blocking. For more information, see [Configure a Point-to-Site \(P2S\) VPN on Windows for use with Azure Files](#) and [Configure a Site-to-Site VPN for use with Azure Files](#).

Consider using Azure File Sync with Azure file shares

The Azure File Sync service allows you to cache Azure file shares on an on-premises Windows Server file server. When you enable cloud tiering, File Sync helps ensure a file server always has free available space, even as it makes more files available than a file server could store locally. If you have on-premises Windows Server file servers, consider

integrating file servers with Azure file shares by using Azure File Sync. For more information, see [Planning for an Azure File Sync deployment](#).

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Scalability

- Azure file share size is limited to 100 tebibytes (TiB). There's no minimum file share size and no limit on the number of Azure file shares.
- Maximum size of a file in a file share is 1 TiB, and there's no limit on the number of files in a file share.
- Maximum I/O operations per second (IOPS) per standard file share is 10,000 IOPS and 100,000 IOPS per premium file share.
- Maximum throughput for a single standard file share is up to 300 mebibytes/sec (MiB/sec) and up to 6,204 MiB/s for premium file shares.
- IOPS and throughput limits are per Azure storage account and are shared between Azure file shares in the same storage account.
- For more information, see [Azure Files scalability and performance targets](#).

Availability

Note

Azure storage account is the parent resource for Azure file shares. Azure file share has the level of redundancy that's provided by the storage account that contains the share.

- Azure file shares currently support the following data redundancy options:
 - **Locally redundant storage (LRS).** Data is copied synchronously three times within a single physical location in the primary region. This practice protects against loss of data because of hardware faults, such as a bad disk drive.
 - **Zone-redundant storage (ZRS).** Data is copied synchronously across three Azure availability zones in the primary region. Availability zones are unique physical locations within an Azure region. Each zone consists of one or more datacenters equipped with independent power, cooling, and networking.

- **Geo-redundant storage (GRS).** Data is copied synchronously three times within a single physical location in the primary region using LRS. Your data is then copied asynchronously to a single physical location in the secondary region. Geo-redundant storage provides six copies of your data spread between two Azure regions.
- **Geo-zone-redundant storage (GZRS).** Data is copied synchronously across three Azure availability zones in the primary region using ZRS. Your data is then copied asynchronously to a single physical location in the secondary region.
- Premium file shares can be stored in locally redundant storage (LRS) and zone redundant storage (ZRS) only. Standard file shares can be stored in LRS, ZRS, geo-redundant storage (GRS), and geo-zone-redundant storage (GZRS). For more information, see [Planning for an Azure Files deployment](#) and [Azure Storage redundancy](#).
- Azure Files is a cloud service, and as with all cloud services, you must have internet connectivity to access Azure file shares. A redundant internet connection solution is highly recommended to avoid disruptions.

Manageability

- You can manage Azure file shares by using the same tools as any other Azure service. These tools include Azure portal, Azure Command-Line Interface, and Azure PowerShell.
- Azure file shares enforce standard Windows file permissions. You can configure directory or file-level permissions by mounting an Azure file share and configuring permissions using File Explorer, Windows `icacls.exe` command, or the `Set-Acl` Windows PowerShell cmdlet.
- You can use Azure file share snapshot for creating a point-in-time, read-only copy of the Azure file share data. You create a share snapshot at the file share level. You can then restore individual files in the Azure portal or in File Explorer, where you can also restore a whole share. You can have up to 200 snapshots per share, which enables you to restore files to different point-in time versions. If you delete a share, its snapshots are also deleted. Share snapshots are incremental. Only the data that has changed after your most recent share snapshot is saved. This practice minimizes the time required to create the share snapshot and saves on storage costs. Azure file share snapshots are also used when you protect Azure file shares with Azure Backup. For more information, see [Overview of share snapshots for Azure Files](#).
- You can prevent accidental deletion of Azure file shares by enabling soft delete for file shares. If you delete a file share when a soft delete is enabled, file share transitions to a soft deleted state instead of being permanently erased. You can

configure the amount of time soft deleted data is recoverable before it's permanently deleted and restore the share anytime during this retention period. For more information, see [Enable soft delete on Azure file shares](#).

 **Note**

Azure Backup enables soft delete for all file shares in the storage account when you configure backup for the first Azure file share in the respective storage account.

 **Note**

Both standard and premium file shares are billed on used capacity when soft deleted, rather than provisioned capacity.

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

- Use AD DS authentication over SMB for accessing Azure file shares. This setup provides the same seamless single sign-on (SSO) experience when accessing Azure file shares as accessing on-premises file shares. For more information, see [How it works](#) and feature [enablement steps](#). Your client needs to be domain joined to AD DS, because the authentication is still done by the AD DS domain controller. Also, you need to assign both share level and file/directory level permissions to get access to the data. [Share level permission assignment](#) goes through Azure RBAC model. [File/directory level permission](#) is managed as Windows ACLs.

 **Note**

Access to Azure file shares is always authenticated. Azure file shares don't support anonymous access. Besides identity-based authentication over SMB, users can authenticate to Azure file share also by using storage access key and Shared Access Signature.

- All data that's stored on Azure file share is encrypted at rest using Azure storage service encryption (SSE). SSE works similarly to BitLocker Drive Encryption on Windows, where data is encrypted beneath the file system level. By default, data stored in Azure Files is encrypted with Microsoft-managed keys. With Microsoft-

managed keys, Microsoft maintains the keys to encrypt/decrypt the data and manages rotating them regularly. You can also choose to manage your own keys, which gives you control over the rotation process.

- All Azure storage accounts have encryption in transit enabled by default. This setup means that all communication with Azure file shares is encrypted. Clients that don't support encryption can't connect to Azure file shares. If you disable encryption in transit, clients that run older operating systems, such as Windows Server 2008 R2 or older Linux, can also connect. In such instances, data isn't encrypted in transit from Azure file shares.
- By default, clients can connect to Azure file share from anywhere. To limit the networks from which clients can connect to Azure file shares, configure the Firewall, virtual networks, and private endpoint connections. For more information, see [Configure Azure Storage firewalls and virtual networks](#) and [Configuring Azure Files network endpoints](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- Azure Files has two storage tiers and two pricing models:
 - **Standard storage:** Uses HDD-based storage. There's no minimum file share size, and you pay only for used storage space. Also, you need to pay for file operations, such as enumerating a directory or reading a file.
 - **Premium storage:** Uses SSD-based storage. The minimum size for a premium file share is 100 gibibytes and you pay per provisioned storage space. When using premium storage, all file operations are free.
- Extra costs are associated with file share snapshots and outbound data transfers. (When you transfer data from Azure file shares, inbound data transfer is free.) Data transfer costs depend on the amount of transferred data and the stock keeping unit (SKU) of your virtual network gateway, if you use one. For more information about the actual costs, see [Azure Files Pricing](#) and [Azure Pricing calculator](#). The actual cost varies by Azure region and your individual contract. Contact a Microsoft sales representative for additional information on pricing.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Andrew Coughlin](#) | Senior Cloud Solutions Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

Learn more about the component technologies:

- [How to create an Azure file share](#) for instructions on getting started with an SMB share.
- [How to create an NFS share](#) for instructions on getting started with an NFS mount share.
- [Enable and create large file shares](#) documentation on creating large file shares upto 100 TiB.

Related resources

Explore related architectures:

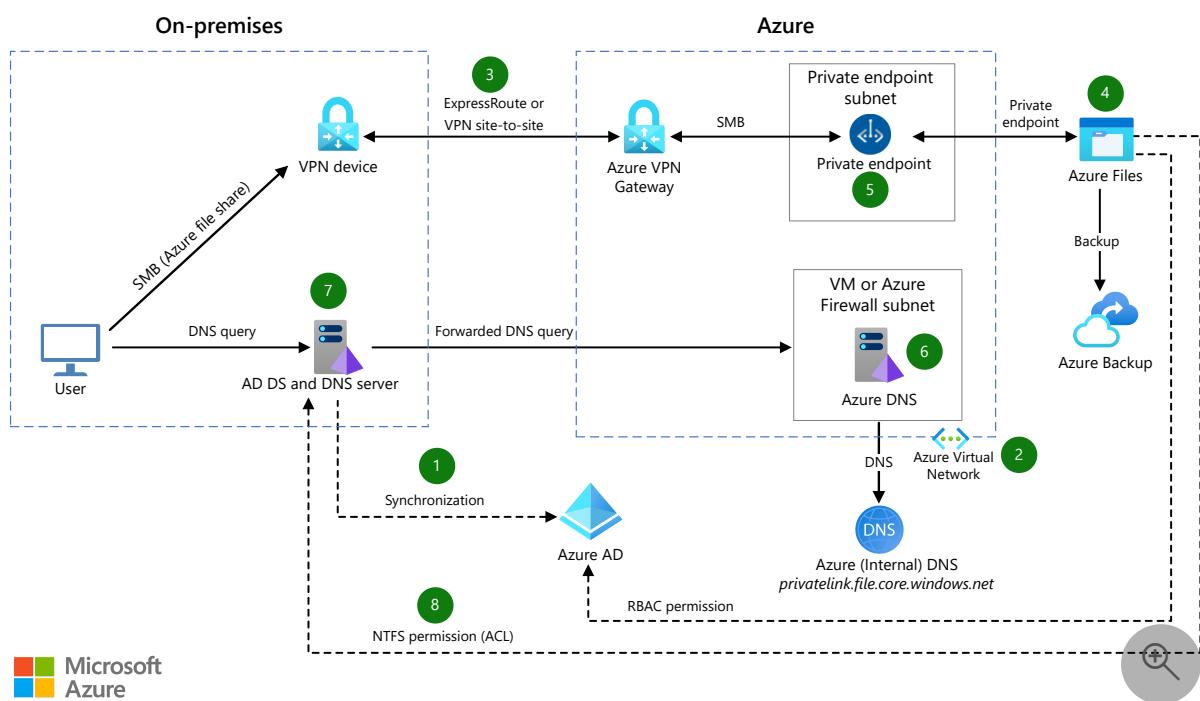
- [Azure enterprise cloud file share](#)
- [Hybrid file services](#)
- [Back up files and applications on Azure Stack Hub](#)
- [Multiple forests with AD DS and Microsoft Entra ID](#)
- [Multiple forests with AD DS, Microsoft Entra ID, and Microsoft Entra Domain Services](#)
- [Azure Virtual Desktop for the enterprise](#)

Azure Files accessed on-premises and secured by AD DS

Azure Virtual Network Azure ExpressRoute Azure Storage Accounts Azure Files Azure DNS

This architecture demonstrates a way to provide file shares in the cloud to on-premises users and applications that also access files on Windows Server.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

1. This solution synchronizes the on-premises AD DS and the cloud-based Microsoft Entra ID. Synchronizing makes users more productive by providing a common identity for accessing both cloud and on-premises resources.

Microsoft Entra Connect is the on-premises Microsoft application that does the synchronizing. For more information about Microsoft Entra Connect, see [What is Microsoft Entra Connect?](#) and [Microsoft Entra Connect Sync: Understand and customize synchronization](#).

2. Azure Virtual Network provides a virtual network in the cloud. For this solution, it has at least two subnets, one for Azure DNS, and one for a private endpoint to access the file share.
3. Either VPN or Azure ExpressRoute provides secure connections between the on-premises network and the virtual network in the cloud. If you use VPN, create a gateway by using Azure VPN Gateway. If you use ExpressRoute, create an ExpressRoute virtual network gateway. For more information, see [What is VPN Gateway?](#) and [About ExpressRoute virtual network gateways](#).
4. Azure Files provides a file share in the cloud. This requires an Azure Storage account. For more information about file shares, see [What is Azure Files?](#).
5. A private endpoint provides access to the file share. A private endpoint is like a network interface card (NIC) inside a subnet that attaches to an Azure service. In this case, the service is the file share. For more information about private endpoints, see [Use private endpoints for Azure Storage](#).
6. The on-premises DNS server resolves IP addresses. However, Azure DNS resolves the Azure file share Fully Qualified Domain Name (FQDN). All DNS queries to Azure DNS originate from the virtual network. There's a DNS proxy inside the virtual network to route these queries to Azure DNS. For more information, see [On-premises workloads using a DNS forwarder](#).

You can provide the DNS proxy on a Windows or Linux server, or you can use Azure Firewall. For information on the Azure Firewall option, which has the advantage that you don't have to manage a virtual machine, see [Azure Firewall DNS settings](#).

7. The on-premises custom DNS is configured to forward DNS traffic to Azure DNS via a conditional forwarder. Information on conditional forwarding is also found in [On-premises workloads using a DNS forwarder](#).
8. The on-premises AD DS authenticates access to the file share. This is a four-step process, as described in [Part one: enable AD DS authentication for your Azure file shares](#)

Components

- [Azure Storage](#) is a set of massively scalable and secure cloud services for data, apps, and workloads. It includes [Azure Files](#), [Azure Table Storage](#), and [Azure Queue Storage](#).

- [Azure Files](#) offers fully managed file shares in an Azure Storage account. The files are accessible from the cloud or on-premises. Windows, Linux, and macOS deployments can mount Azure file shares concurrently. File access uses the industry standard Server Message Block (SMB) protocol.
- [Azure Virtual Network](#) is the fundamental building block for private networks in Azure. It provides the environment for Azure resources, such as virtual machines, to securely communicate with each other, with the internet, and with on-premises networks.
- [Azure ExpressRoute](#) extends on-premises networks into the Microsoft cloud over a private connection.
- [Azure VPN Gateway](#) connects on-premises networks to Azure through site-to-site VPNs, in much the same way as you connect to a remote branch office. The connectivity is secure and uses the industry-standard protocols Internet Protocol Security (IPsec) and Internet Key Exchange (IKE).
- [Azure Private Link](#) provides private connectivity from a virtual network to Azure platform as a service (PaaS), customer-owned, or Microsoft partner services. It simplifies the network architecture and secures the connection between endpoints in Azure by eliminating data exposure to the public internet.
- A private endpoint is a network interface that uses a private IP address from your virtual network. You can use private endpoints for your Azure Storage accounts to allow clients on a virtual network to access data over a private link.
- [Azure Firewall](#) is a managed, cloud-based network security service that protects your Azure Virtual Network resources. It's a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability. You can configure Azure Firewall to act as a DNS proxy. A DNS proxy is an intermediary for DNS requests from client virtual machines to a DNS server.

Scenario details

Consider the following common situation. An on-premises Windows Server provides files to users and applications. Windows Server Active Directory Domain Services (AD DS) secures the files, and there's an on-premises DNS server. Everything is on the same private network.

Now suppose that the need arises to have file shares in the cloud.

The architecture that's described here shows how to use Azure to satisfy this need, and how to do it at low cost, and by continuing to use the on-premises network, AD DS, and DNS.

In this architecture, Azure Files provides the file share. Site-to-site VPN or Azure ExpressRoute provides secure connections between the on-premises network and Azure virtual network. Users and applications use the connections to access the files. Microsoft Entra ID and Azure DNS cooperate with on-premises AD DS and DNS to secure the access.

In short, if you're in the described situation, you can provide cloud files to your on-premises users at low cost, and continue to provide secure file access with your on-premises AD DS and DNS.

Potential use cases

- The file server moves to the cloud, but the users must remain on-premises.
- Applications that are migrated to the cloud need to access on-premises files, and also files that are migrated to the cloud.
- You need to reduce costs by moving file storage to the cloud.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Reliability

Reliability ensures that your application can meet the commitments that you make to your customers. For more information, see [Overview of the reliability pillar](#).

- Azure Storage always stores multiple copies of your data in the same zone, so that it's protected from planned and unplanned outages. There are options for creating additional copies in other zones or regions. For more information, see [Azure Storage redundancy](#).
- Azure Firewall has built-in high availability. For more information, see [Azure Firewall Standard features](#).

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

These articles have security information for Azure components:

- [Azure security baseline for Azure Storage](#)
- [Azure security baseline for Azure Private Link](#)
- [Azure security baseline for Virtual Network](#)
- [Azure security baseline for Azure Firewall](#)

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To estimate the cost of Azure products and configurations, use the Azure [Pricing calculator](#).

These articles have pricing information for Azure components:

- [Azure Files pricing](#)
- [Azure Private Link pricing](#)
- [Virtual Network pricing](#)
- [Azure Firewall pricing](#)

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

- Your Azure Storage accounts contain all of your Azure Storage data objects, including file shares. A storage account provides a unique namespace for its data, a namespace that's accessible from anywhere in the world over HTTP or HTTPS. For this architecture, your storage account contains file shares that are provided by Azure Files. For best performance, we recommend the following:
 - Don't put databases, blobs, and so on, in storage accounts that contain file shares.
 - Have no more than one highly active file share per storage account. You can group file shares that are less active into the same storage account.
 - Use SSD-based storage rather than HDD. For more information about the scalability and performance of file shares, see [Azure Files scalability and performance targets](#).
 - Don't select a general-purpose v1 storage account, because it lacks important features. The storage account types are described in [Storage account overview](#).

- Pay attention to size, speed, and other limitations. For this information, refer to [Azure subscription and service limits, quotas, and constraints](#).
- There's little you can do to improve the performance of non-storage components, except to be sure that your deployment honors the limits, quotas, and constraints that are described in [Azure subscription and service limits, quotas, and constraints](#).
- For scalability information for Azure components, see [Azure subscription and service limits, quotas, and constraints](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- Rudnei Oliveira  | Senior Customer Engineer

Next steps

- Quickstart: Create a virtual network using the Azure portal
- What is VPN Gateway?
- Tutorial: Create and manage a VPN gateway using Azure portal
- Azure enterprise cloud file share
- Azure Virtual Network concepts and best practices
- Planning for an Azure Files deployment
- Use private endpoints for Azure Storage
- Azure Private Endpoint DNS configuration
- Azure Firewall DNS settings
- Compare self-managed Active Directory Domain Services, Microsoft Entra ID, and managed Microsoft Entra Domain Services

Related resources

- Hybrid file share with disaster recovery for remote and local branch workers
- Azure enterprise cloud file share
- Using Azure file shares in a hybrid environment
- Hybrid file services

Application data protection for AKS workloads on Azure NetApp Files

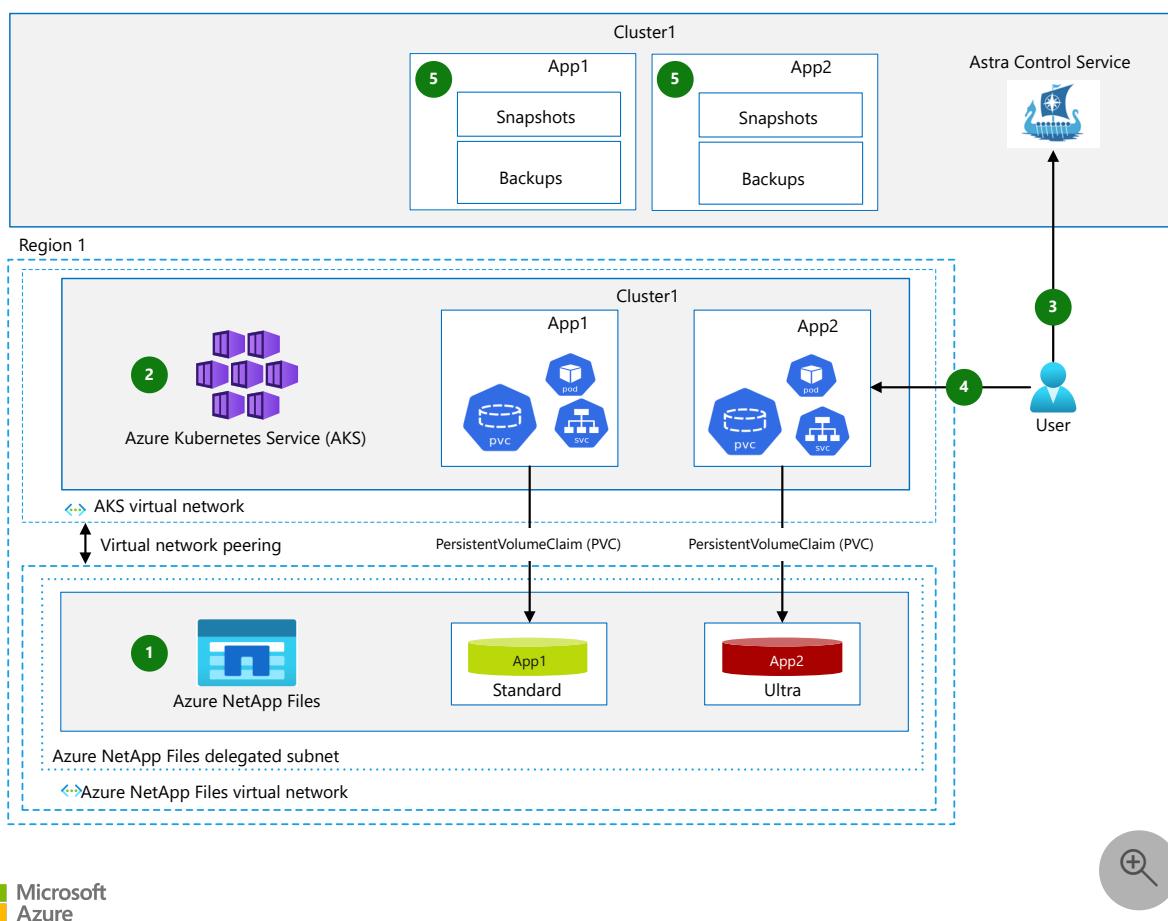
Azure NetApp Files

Azure Kubernetes Service (AKS)

Azure Virtual Network

This article outlines a solution for managing and performing application data management of stateful containerized applications, their resources, and their data.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. An Azure NetApp Files account is created on an Azure subscription, and capacity pools are defined. These pools map to service levels that the implementation needs, such as Standard, Premium, and Ultra.

2. One or more AKS clusters are deployed. The clusters need to be:

- In a region where AKS and Azure NetApp Files are available. For regions where these products are available, see [Products available by region](#).
- In a virtual network that has direct access to a subnet that's delegated for Azure NetApp Files. For more information, see [Guidelines for Azure NetApp Files network planning](#).

3. A user signs up for an [Astra Control Service account](#). Astra Control Service uses an Azure service principal credential that has contributor access to locate the AKS clusters to be managed. Astra Control Service installs Astra Trident and creates StorageClasses mapped to each tier of service when a cluster is added to Astra Control Service. Astra Trident creates Kubernetes PersistentVolumes (PVs) from application PersistentVolumeClaims (PVCs) using the automatically deployed StorageClass (SC) objects that map to the Azure NetApp Files capacity pools. The mapping takes into account the service level of the capacity pools.

4. The user installs applications on the AKS clusters. Possible deployment methods include Helm charts, operators, and YAML manifests. The applications can be grouped by labels or namespaces. Astra Trident provisions persistent volumes based on the PersistentVolumeClaims using the `StorageClass` objects.

5. Astra Control Service manages applications and their associated resources, such as pods, services, deployments, and `PersistentVolumeClaim` (PVC) objects. It also manages the PersistentVolume (PV) bound to the PVC. Users define applications by using one of these methods:

- Confining them to a namespace
- Using a custom Kubernetes label to group resources

Users can also group cluster-scoped objects, such as storageclasses, with (a) specific application(s) to manage them together.

Astra Control Service orchestrates [point-in-time snapshots](#) and backups, [backup policies](#), and [instant active clones](#) to help protect application workloads. Astra Control Service achieves this protection by:

- Creating Astra Control Service protection policies. These can be made for snapshots and/or backups and specify a schedule and backup target. These policies make it possible to automatically protect applications on a pre-determined schedule.
- Taking snapshots on demand for individual or a group of applications.

- Making instantaneous backups or clones for individual or a group of applications.

When disasters or app failures occur, backups and snapshots restore applications' state. Users can clone and migrate apps across namespaces and AKS clusters. The clusters can be in the same or separate regions.

Components

- [AKS](#) is a fully managed Kubernetes service that makes it easy to deploy and manage containerized applications. AKS offers serverless Kubernetes technology, an integrated continuous integration and continuous delivery (CI/CD) experience, and enterprise-grade security and governance.
- [Azure NetApp Files](#) is an Azure storage service. This service provides enterprise-grade network file system (NFS) and server message block (SMB) file shares. Azure NetApp Files makes it easy to migrate and run complex, file-based applications with no code changes. This service is well suited for users with persistent volumes in Kubernetes environments.
- [Azure Virtual Network](#) is the fundamental building block for private networks in Azure. Through Virtual Network, Azure resources like virtual machines can securely communicate with each other, the internet, and on-premises networks.
- [Astra Control Service](#) is a fully managed application-aware data management service. Astra Control Service helps you manage, protect, and move data-rich Kubernetes workloads in public clouds and on-premises environments. This service provides data protection, disaster recovery, and migration for Kubernetes workloads. Astra Control Service uses the industry-leading [data management technology of Azure NetApp Files for snapshots, backups, cross-region replication, and cloning](#).

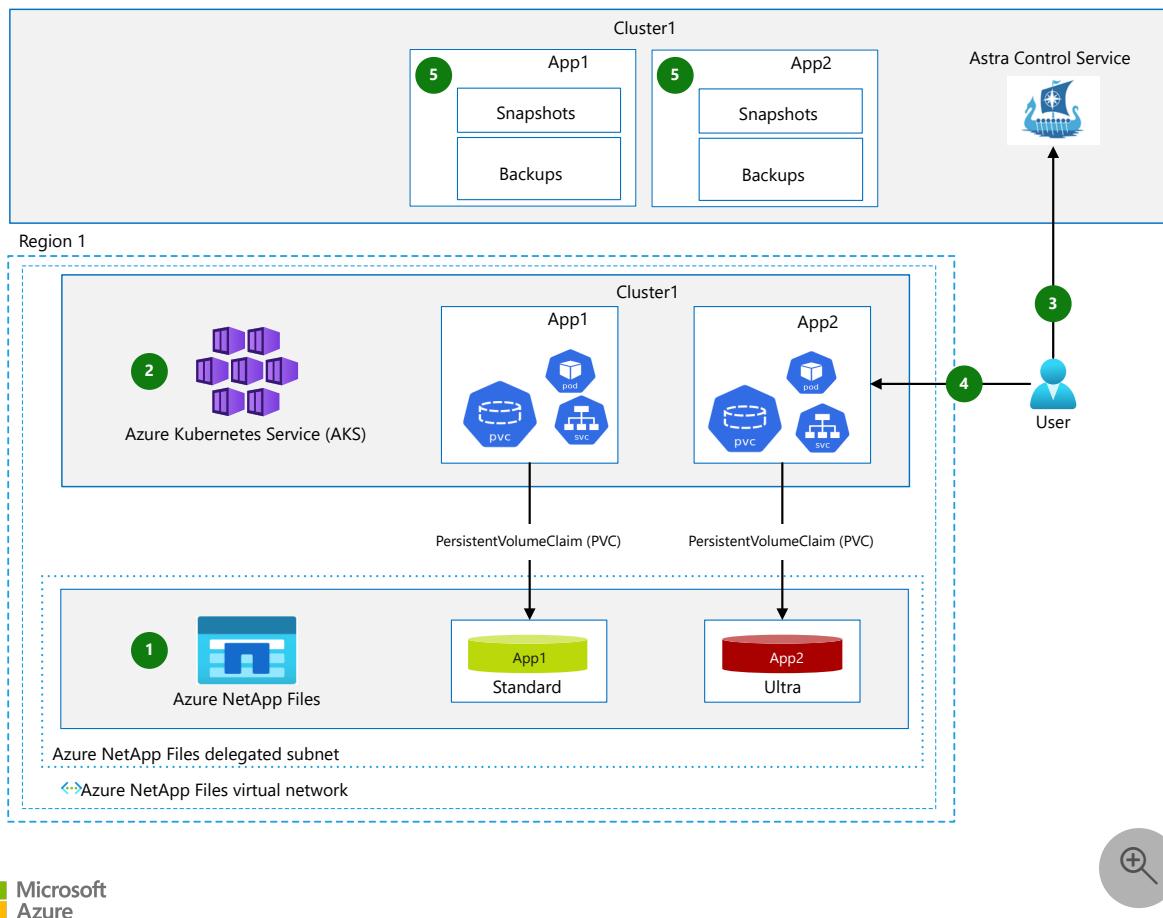
Alternatives

You can use a custom multi-pronged approach to separately back up or replicate persistent volumes, Kubernetes resources, and other configuration state resources that you need when you restore an application. But this approach can be:

- Cumbersome.
- Difficult to make compatible with all apps.
- Difficult to scale across the multiple apps and environments that a typical enterprise has.

In certain environments, you can reduce costs by avoiding cross-peered virtual network traffic. To eliminate this traffic, simplify the solution. Specifically, bring the AKS clusters

and the subnet that you delegate for Azure NetApp Files into the same virtual network, as this diagram illustrates:



Download a [Visio file](#) of this architecture.

Scenario details

With containerized applications, it can be challenging to perform application-data protection. The application consists of multiple microservices, which must be managed as one entity. When you deploy business-critical workloads on Kubernetes, application data management should be:

- Simple. Establishing data protection policies and on-demand snapshots and backups should be intuitive. These policies shouldn't be dependent on the details of the underlying infrastructure.
- Portable. To make cross-region mobility possible for applications, multiple Kubernetes clusters should be able to consume the backups.
- Application-aware. Your solution should protect the entire application, including standard Kubernetes resources like secrets, **ConfigMap** objects, and persistent volumes. You also need to protect custom Kubernetes resources. When possible,

procedures should quiesce the application prior to the snapshot and backup. This practice prevents the loss of in-flight data during backups.

[NetApp Astra Control Service](#) is a solution for performing stateful application data management that helps you meet these goals. Astra Control Service offers data protection, disaster recovery, and application mobility capabilities. It provides stateful AKS workloads with a rich set of storage and application-aware data management services. The data protection technology of Azure NetApp Files underlies these services.

Potential use cases

This solution applies to systems that run stateful applications:

- Continuous integration (CI) systems such as Jenkins
- Database workloads like MySQL, MongoDB, and PostgreSQL
- AI and machine-learning components such as TensorFlow and PyTorch
- Elasticsearch deployments
- Kafka applications
- Source code management platforms like GitLab

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Reliability

Reliability ensures your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

When you deploy an AKS cluster, you deploy it in a single region. To protect application workloads, it's best to deploy the workloads across [multiple AKS clusters that span multiple regions](#). Factors that affect deployment include [AKS region availability](#) and Azure [paired regions](#). When you deploy clusters across multiple availability zones, you distribute nodes across multiple zones within a single region. This distribution of AKS cluster resources improves cluster availability because the clusters are resilient to the failure of a specific zone.

Azure NetApp Files is highly available by design. It's built on a highly available bare-metal fleet of all flash storage systems. For this service's availability guarantee, see [SLA](#)

for Azure NetApp Files [♂](#).

Azure NetApp Files supports cross-region replication for disaster recovery. You can replicate volumes between Azure region pairs continuously. For more information about cross-region replication, see these resources:

- For general information, see [Cross-region replication of Azure NetApp Files volumes](#).
- For requirements for cross-region replication, see [Manage disaster recovery using cross-region replication](#).
- For information about configuring cross-region replication, see [Create volume replication for Azure NetApp Files](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Use the [Azure Pricing calculator](#) [♂](#) to estimate the cost of the following components:

- AKS
- Azure NetApp Files
- Virtual Network

For Astra Control Service pricing plans, see [Pricing](#) [♂](#). By adopting Astra Control Service, you can focus on your application instead of spending time and resources building custom solutions that don't scale. Astra Control Service is available on [Azure Marketplace](#) [♂](#).

To run detailed bandwidth and pricing calculations, use the [Azure NetApp Files Performance Calculator](#) [♂](#). Basic and advanced calculators are available.

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

When you work with the Kubernetes control plane, it's important to monitor your infrastructure and platform layer. Astra Control Service provides a unified control plane that you can use to define and manage application protection policies across multiple AKS clusters. A [dashboard](#) [♂](#) provides a way for you to continuously handle workloads

across regions. Astra Trident also provides a rich set of [Prometheus metrics](#) that you can use to monitor provisioned storage.

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

AKS clusters can add extra worker nodes to increase scalability. To scale your solution, you can add node pools or scale existing node pools. These steps increase the number of nodes in your cluster, the total number of cores, and the memory that's available for your containerized applications.

In each virtual network, you can only delegate one subnet for Azure NetApp Files.

When you use a basic configuration for Azure NetApp Files network features, there's a limit of 1,000 IP addresses per virtual network. The standard network features configuration doesn't limit the number of IP addresses. For more information, see [Configurable network features](#). For a complete list of resource limits for Azure NetApp Files, see [Resource limits for Azure NetApp Files](#).

Azure NetApp Files offers multiple performance tiers. When you use Astra Control Service to discover AKS clusters, the onboarding process creates curated `StorageClass` objects that map to the Standard, Premium, and Ultra service tiers. When users deploy applications, they choose a storage tier that suits their requirements. Multiple capacity pools can coexist. Provisioned volumes have a performance guarantee that corresponds to the service tier. For a list of service levels that Azure NetApp Files supports, see [Service levels for Azure NetApp Files](#).

Deploy this scenario

To implement this solution, you need an Azure account. [Create an account for free](#).

To deploy this scenario, follow these steps:

1. [Register the resource provider](#) that makes it possible to use Azure NetApp Files.
2. Review the [Requirements for using Astra Control Service with AKS](#).
3. Use the Azure portal to [create a NetApp account](#).
4. [Set up capacity pools](#) on the Azure NetApp Files account.
5. [Delegate a subnet](#) for Azure NetApp Files.

6. Create a [service principal](#) for Astra Control Service to use to discover AKS clusters and perform backup, restore, and data management operations.
7. [Register for Astra Control Service](#) by creating a NetApp Cloud Central account.
8. [Add AKS clusters to Astra Control Service](#) to start managing applications.
9. [Detect applications](#) in Astra Control Service. The way you discover and manage applications depends on the way you deploy and identify them. Typical identification strategies include grouping application objects in a dedicated namespace, assigning labels to objects that make up an application, and using Helm charts. Astra Control Service supports all three strategies.
10. [Establish protection policies](#) to back up and restore applications. Before you define protection policies, clearly identify your workloads. A prerequisite is that Astra Control Service can uniquely detect each application. For more information, see [Start managing apps](#).

For steps that you can take to help protect applications, see [Disaster Recovery of AKS Workloads with Astra Control Service and Azure NetApp Files](#).

For detailed information about Astra Control Service, see [Astra Control Service documentation](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributor.

Principal author:

- [Arnt de Gier](#) | Technical Marketing Engineer

Next steps

- For information about using AKS to deploy a cluster, see [Tutorial: Deploy an Azure Kubernetes Service \(AKS\) cluster](#).
- To get started with Azure NetApp Files, see [Quickstart: Set up Azure NetApp Files and create an NFS volume](#).
- To learn more about Astra Control Service, see [Astra Control Service documentation](#).
- For an in-depth explanation of using Astra Control Service for disaster recovery, see [Disaster Recovery of AKS workloads with Astra Control Service and Azure](#)

[NetApp Files ↗](#).

- For information about running multiple instances of an AKS cluster across multiple regions, see [AKS baseline for multiregion clusters](#).
- For general information about solution components, see these resources:
 - [What is Azure Kubernetes Service?](#)
 - [What is Azure NetApp Files](#)
 - [NetApp Astra Control Service ↗](#)

Related resources

- [Enterprise file shares with disaster recovery](#)
- [Magento e-commerce platform in Azure Kubernetes Service](#)
- [Baseline architecture for an Azure Kubernetes Service \(AKS\) cluster](#)

Enterprise file shares with disaster recovery

Azure NetApp Files Microsoft Entra Windows Server

This architecture provides file shares that fail over automatically to a backup region in case of failure. The failover is transparent to the clients and applications that access the shares. The shares can be used for applications and virtual desktops that must be resilient to disruption, whether planned or unplanned.

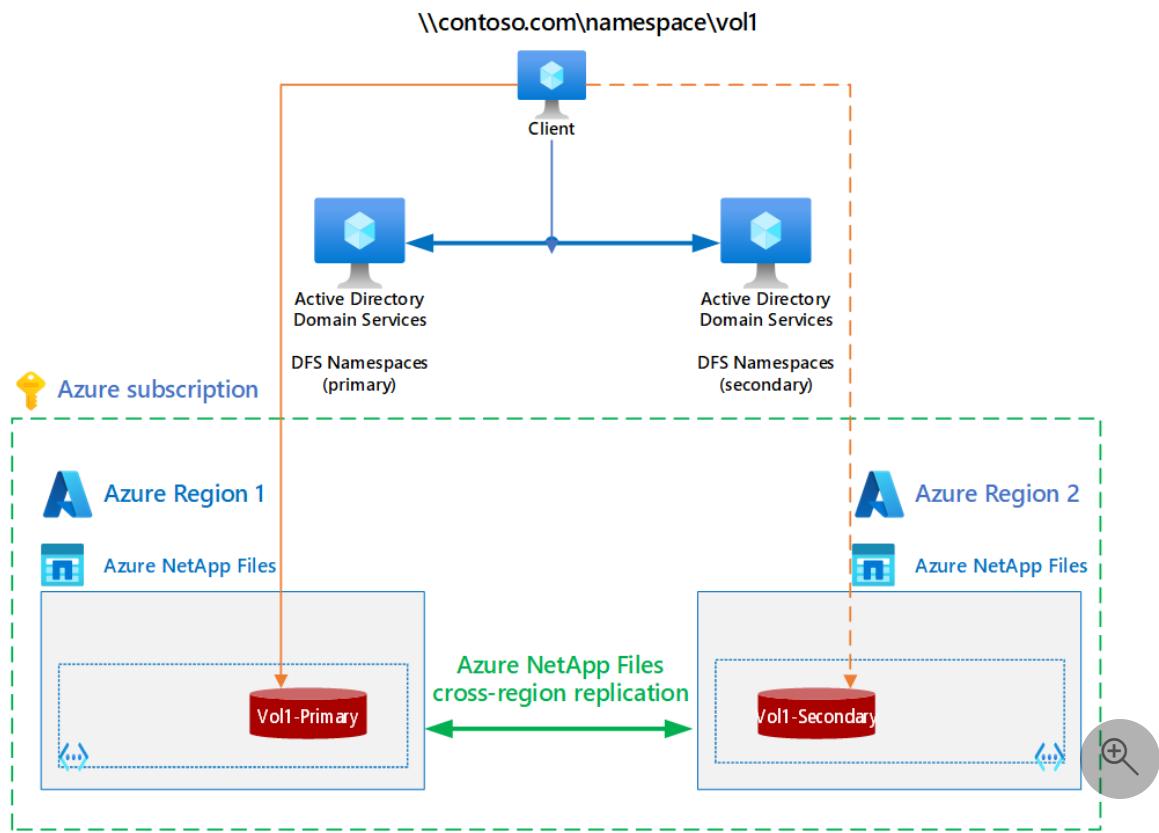
Azure NetApp Files provides the file shares. Its cross-region replication capability replicates the shares from the primary region to the secondary. Distributed File System (DFS) Namespaces in Windows Server can group shared folders on different servers into one or more logically structured namespaces.

Potential use cases

This architecture applies to businesses that want to provide file shares for clients or applications that must be resilient to unplanned outages or service maintenance events. Some examples are:

- Service Message Block (SMB) protocol file shares for desktop environments.
- SMB file shares for applications.

Architecture



Download a [Visio file](#) of this architecture.

- There are two Azure regions, a primary and a secondary.
- The Azure subscription includes a virtual network and an Azure NetApp Files account for each region.
- The cross-region replication feature of Azure NetApp Files replicates the files and folders from the primary region to the secondary region. This technique doesn't need virtual machines.
- Access to the file shares is managed by DFS Namespaces, a feature of Windows Server. You can think of it as Domain Name Server (DNS) for file shares.
- The Windows servers and Active Directory Domain servers can be hosted on Azure or on-premises.

Components

- [Azure NetApp Files](#) provides enterprise-grade Azure file shares that are powered by NetApp. Azure NetApp Files makes it easy for enterprises to migrate and run complex file-based applications with no code changes. It also provides a way to replicate data asynchronously from an Azure NetApp Files volume in one region to an Azure NetApp Files volume in another region. This capability provides data protection during region-wide outages or disasters. For more information, see [Cross-region replication of Azure NetApp Files volumes](#).
- DFS Namespaces is a role service in Windows Server that can group shared folders that are located on different servers into one or more logically structured

namespaces. For more information, see [DFS Namespaces overview](#).

Alternatives

- Instead of Azure NetApp Files, you can use a Windows Server Scale-Out File Server cluster with custom replication of the file shares across regions. For more information, see [Scale-Out File Server for application data overview](#).
- Instead of Azure NetApp Files cross-region replication, you can use Azure File Sync to transform Windows Server into a quick cache of your Azure file shares. This might be appropriate for smaller file shares. For more information, see [Deploy Azure File Sync](#).

Considerations

The [Azure Well-Architected Framework](#) provides reference guidance and best practices to apply to your architecture.

Availability

Replicating to a second region increases availability by protecting against regional service interruptions.

Performance efficiency

- Azure NetApp Files comes with three performance tiers: Standard, Premium, and Ultra. Cross-region replication can replicate between different tiers. When the primary region uses the Premium or Ultra tier, you can replicate to a lower tier, for example Standard. In case of a failover, you can then upgrade the tier of the secondary as required.
- The replication of the data is performed at the incremental block level—only changed data blocks are transferred—which minimizes data transfer.

Scalability

This solution can be used for file shares ranging from 4 tebibytes (TiB) to a total volume of 12.5 pebibytes (PiB) on a single Azure NetApp Files account.

Resiliency

- This solution has greater resiliency than a single-region deployment, and has failover capabilities.
- The secondary volume is read-only. It can be verified at any given time, increasing resiliency.
- You can run a disaster recovery test in isolation without interfering with the production deployment. The test uses the space-efficient volume clone feature to get a read/write copy of a volume in seconds.

Cost optimization

The cost of the solution depends on the size of the volume that's replicated, the rate of change, and the destination tier of the Azure NetApp Files capacity pool. For details, see [Azure NetApp Files pricing](#) or use the Azure [Pricing calculator](#).

See [Cost model for cross-region replication](#) for more examples.

Deploy this scenario

To deploy on Azure, perform the following configuration tasks in the Windows Server DFS namespace:

1. Deploy the primary Azure NetApp Files account.
2. Create an SMB volume on the primary.
3. Deploy the secondary Azure NetApp Files account.
4. Replicate the volume to the secondary Azure NetApp Files account.
5. Configure DFS Namespaces to point to the primary volume.

In case of a failover:

1. Fail over the volumes of Azure NetApp Files.
2. Change the targets in DFS Namespaces.

These tasks can be and should be automated.

See [Disaster Recovery for Enterprise File Shares](#) for a step-by-step deployment guide.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- Max Melcher  | Cloud Solution Architect

Next steps

- [Register for NetApp Resource Provider](#)
- [Create a NetApp account](#)
- [Quickstart: Set up Azure NetApp Files and create an NFS volume](#)
- [Disaster Recovery for Enterprise File Shares](#) 

Related resources

- [Hybrid file share with disaster recovery for remote and local branch workers](#)

Moodle deployment with Azure NetApp Files

Azure Application Gateway Azure Cache for Redis Azure Database for MySQL Azure NetApp Files

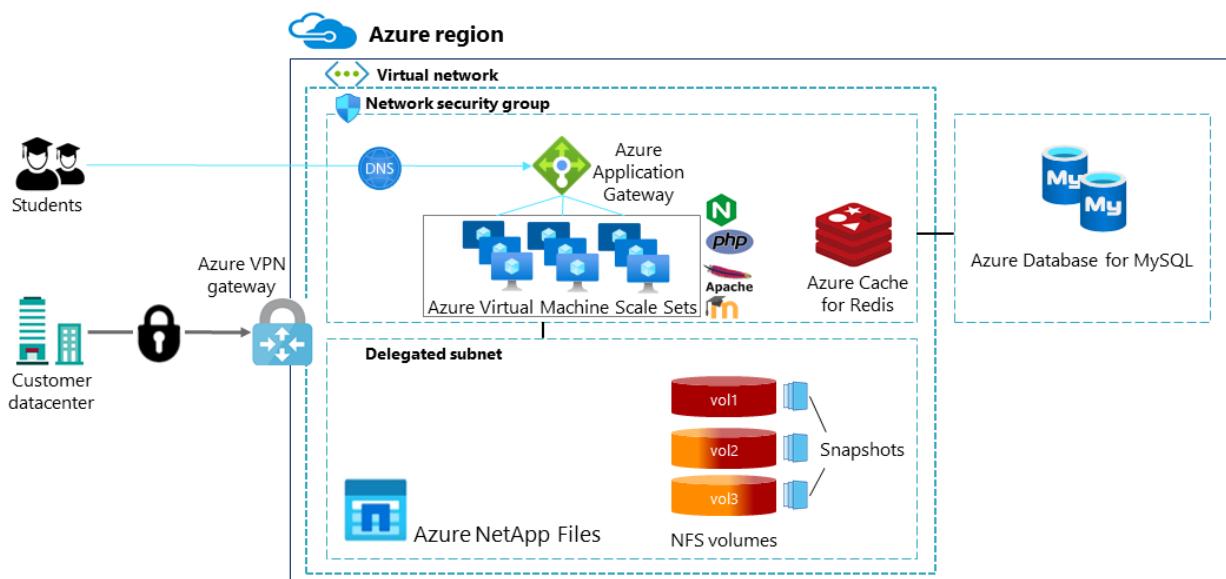
Azure Virtual Machine Scale Sets

In a single region, this solution provides highly available access to the Moodle app and other components. For detailed information on availability, see [Availability](#), later in this article. You can also use two regions to implement this solution. With two regions, the solution provides disaster recovery. To protect against an unlikely Azure region failure, you replicate the data volumes to the second region. Only the Azure NetApp Files volumes need to be present in that region.

Apache® is either a registered trademark or trademark of the Apache Software Foundation in the United States and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of this mark.

Architecture

Single-region highly available setup

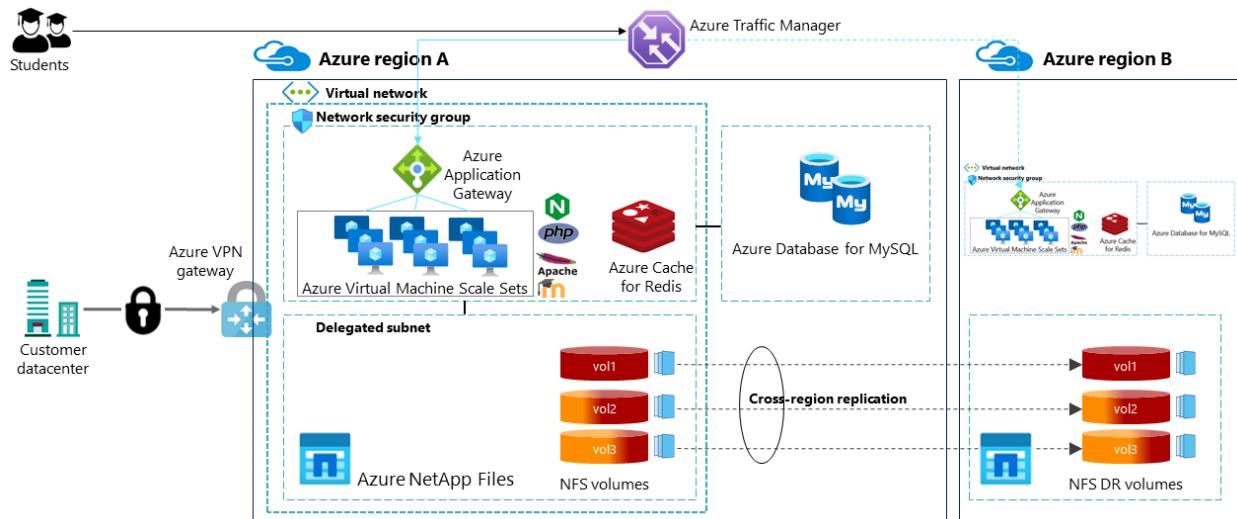


Download a [PowerPoint file](#) of this architecture.

1. Students access Moodle application data through [Azure Application Gateway](#).

2. Moodle is written in PHP. Moodle runs in a [virtual machine scale set](#) on a web server such as Apache HTTP Server or NGINX.
3. [Azure NetApp Files](#) makes the content data available to Moodle.
4. The solution uses [Azure Cache for Redis](#) for user session caching, locking, and key awareness.
5. An [Azure Database for MySQL](#) database stores the learning content, student progress data, and internal data.
6. Learning content enters the system through a secure virtual private network (VPN) gateway directly from the customer datacenter.

Dual-region disaster recovery setup



Download a [PowerPoint file](#) of this architecture.

1. [Cross-region replication](#) provides replication for the Azure NetApp Files volumes. This storage-based replication engine is built into Azure NetApp Files.
2. When you use cross-region replication, you don't have to turn on some components during normal operation. So those components don't incur any cost. When a failover occurs, you can start those components and use them with the replicated data volumes.
3. After you recover the primary region, the replication direction reverses. The primary region is updated with any changes that were applied during the failover. You can then fail the service back.
4. [Azure Traffic Manager](#) directs users to the region that's currently active.

Components

- [Moodle](#) is a free, open-source learning management system.

- [Azure Database for MySQL](#) is a fully managed relational database service that's based on the community edition of the open-source MySQL database engine.
- [Azure Cache for Redis](#) is a fully managed, in-memory data store that's based on the open-source software Redis.
- [Azure Virtual Machine Scale Sets](#) provides a way to manage a group of load-balanced virtual machines (VMs). The number of VMs in a set automatically increases or decreases in response to demand or a defined schedule.
- [Azure NetApp Files](#) makes it easy to migrate and run file-based applications with no code changes. This shared file-storage service is a joint development from Microsoft and NetApp, a Microsoft partner.
- [Cross-region replication](#) provides a way to replicate data asynchronously from an Azure NetApp Files volume in one region to another Azure NetApp Files volume in another region. This capability provides data protection during region-wide outages or disasters.
- [Azure Application Gateway](#) is a load balancer that manages traffic to web applications.
- [Traffic Manager](#) is a load balancer that distributes traffic to applications across global Azure regions. Traffic Manager also provides public endpoints with high availability and quick responsiveness.

Alternatives

To deploy Moodle, you can use any NFS-based shared file service that meets requirements for very low latency, high IOPS, and high throughput. These conditions are especially important for high numbers of concurrent users. You can use an NFS service that's built on top of a set of Linux VMs. But this approach presents manageability, scalability, and performance challenges. In contrast, Azure NetApp Files offers a competitive, low-latency solution that delivers excellent performance and secure access to NFS shared storage.

Scenario details

Moodle is one of the most popular and widely adopted free, open-source learning management systems. With more than 30 percent of the global market share, Moodle has more than 180,000 customers worldwide. By providing a high-bandwidth, low-latency solution for workloads, Azure NetApp Files meets Moodle's performance

requirements. This solution is also flexible. Deployments can grow or shrink on demand to make your configuration cost effective.

Since the emergence of COVID-19, Moodle has seen a surge in growth. The company is now the market leader in learning management systems. This growth has forced Moodle to explore options for quickly expanding its business and enabling customers to quickly and efficiently deploy Moodle instances in the cloud. Moodle architecture relies on the Network File System (NFS) 3.0 protocol (NFSv3) for content storage.

Moodle strives to meet the demands of home workers and to provide the best possible user experience. As a result, Moodle requires:

- Consistent high-throughput, low-latency access to shared storage.
- A way to scale up the solution to accommodate an increasing number of concurrent users. Customers prefer autoscaling configurations.

This article outlines a solution that meets Moodle's needs. At the core of the solution is Azure NetApp Files, a first-party storage service. You can use this service to migrate and run the most demanding enterprise-scale file workloads in the cloud:

- Native Server Message Block (SMB) version 3, NFSv3, and NFSv4.1 file shares
- Database workloads
- Data warehouse workloads
- High-performance computing applications

Potential use cases

This solution applies to Moodle deployments. Organizations that use Moodle span many industries, including education, business, IT, and finance.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Keep the following points in mind when you implement this solution.

Scalability

This solution scales up or down as needed:

- Virtual Machine Scale Sets provides automatic scaling of resources. For more information, see [Overview of autoscale with Azure Virtual Machine Scale Sets](#).
- You can easily and non-intrusively scale the Azure NetApp Files capacity pools and volumes up and down to meet demand. For more information, see [Resize a capacity pool or a volume](#).
- You can adjust the Azure NetApp Files volume service level, which can be either Standard, Premium, or Ultra. The level that you select affects the throughput limit of volumes with automatic quality of service (QoS). For more information, see [Performance considerations for Azure NetApp Files](#).

Availability

For the Azure NetApp Files availability guarantee, see [SLA for Azure NetApp Files](#) ↗.

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

For all deployment options, you need to provide a valid Secure Shell (SSH) protocol 2 (SSH-2) RSA public–private key pair. The length should be at least 2048 bits. Azure doesn't support other key formats such as ED25519 and ECDSA. For information about Azure NetApp Files security, see [Security FAQs for Azure NetApp Files](#).

Resiliency

Azure NetApp Files is built on a bare-metal fleet of redundant, solid-state hardware. The service operates without interruption, even during maintenance operations. For more information about resiliency, see [Fault Tolerance, High Availability, and Resiliency in Azure NetApp Files](#) ↗.

Disaster recovery

As [Architecture](#) explains earlier in this article, you can make the solution more resilient. You can provide disaster recovery by adding a secondary region and using Azure NetApp Files cross-region replication. This functionality efficiently replicates the NFS volumes to a secondary passive region. During the unlikely event of a complete region failure, the application runs in that secondary region.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Consider a medium-sized to large-sized Moodle deployment of approximately 5,000 users with a 10 percent concurrency ratio. The recommended throughput for this case is about 500 MBps. You can build this type of system on a Linux-based Standard_D32s_v4 VM that uses 8 TB of P60 managed disk.

Azure NetApp Files offers a more cost-effective solution. It achieves the recommended throughput of 500 MBps but uses only 4 TB of Ultra service-level capacity. The Premium and Standard service levels are also often sufficient, further improving the cost effectiveness. Even when the scale of the application is larger and the application requires more Azure NetApp Files capacity, these service levels can likely deliver the recommended throughput.

Use the [Azure pricing calculator](#) to estimate the cost of the Azure resources that your implementation requires. For more information on Azure NetApp Files cost modeling, see [Cost model for Azure NetApp Files](#).

For a calculator that computes the Azure NetApp Files performance and total cost of ownership (TCO), see [Azure NetApp Files Performance Calculator](#). Use this calculator to find the optimal balance between capacity, performance, and cost.

Deploy this scenario

For a deployment guide for Moodle on Azure NetApp Files, see [Azure NetApp Files for NFS storage with Moodle](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Arnt de Gier](#) | Technical Marketing Engineer

Next steps

- [The MoodleCloud model, a typical starting model](#)
- [Directions for scaling up or deploying Moodle quickly and efficiently in Azure](#)

- Solution architectures using Azure NetApp Files
- Redis cache store ↗
- Azure NetApp Files for NFS storage with Moodle ↗
- Public preview: Automatic scaling with Azure Virtual Machine Scale Sets flexible orchestration mode ↗

Product documentation:

- [What are Azure Virtual Machine Scale Sets?](#)
- [What is Azure Database for MySQL?](#)
- [What is Azure Cache for Redis?](#)
- [What are Azure Virtual Machine Scale Sets?](#)
- [What is Azure NetApp Files?](#)
- [What is Azure Application Gateway?](#)
- [What is Azure Traffic Manager?](#)

Related resources

- [SQL Server on Azure Virtual Machines with Azure NetApp Files](#)
- [Oracle Database with Azure NetApp Files](#)
- [Run SAP NetWeaver in Windows on Azure](#)
- [SAS on Azure architecture](#)

Oracle Database with Azure NetApp Files

Azure NetApp Files

Azure Virtual Machines

Azure Virtual Network

The most demanding Oracle Database workloads require very high I/O capacity. They also need low-latency access to storage. This document describes a high-bandwidth, low-latency solution for Oracle Database workloads.

The solution provides shared file access with the network file system (NFS) protocol. The architecture uses Azure NetApp Files, a shared file-storage service. Azure NetApp Files offers benefits:

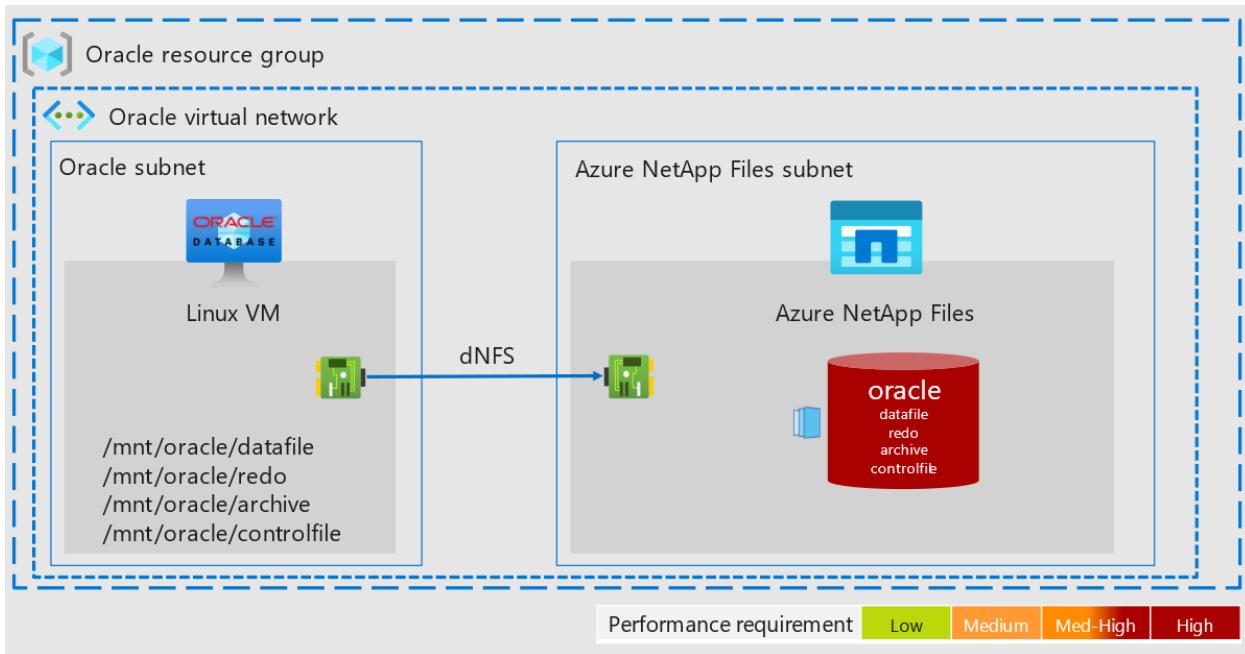
- Disk I/O limits on access rates that apply at the virtual machine (VM) level don't affect Azure NetApp Files. As a result, you can use smaller VMs than you would with disk storage without degrading performance. This approach significantly reduces costs.
- Azure NetApp Files offers flexibility. You can enlarge or reduce deployments on demand to make your configuration cost effective.

Potential use cases

This solution has many uses:

- Running new Oracle Database instances that require high availability (HA) and have high standards for performance.
- Migrating highly performant, highly available Oracle Database instances from on-premises infrastructure to Azure Virtual Machines.
- Cloning enterprise-scale Oracle Database systems for use in test and development environments. The solution is particularly suited for cases that require advanced data management capabilities. It can help these cases meet aggressive data protection service level agreements (SLAs).
- Migrating Oracle Exadata systems to Azure.
- Implementing Oracle Pacemaker clusters that use NFS shared storage.
- Deploying SAP AnyDB, or Oracle 19c.

Architecture



[Download an **SVG** of this architecture.](#)

The components interact in these ways:

- Oracle Database runs on Azure VMs within the Oracle subnet.
- In the Azure NetApp Files subnet, Azure NetApp Files provides NFS access to the data and log files.
- The connection protocol [Oracle Direct NFS \(dNFS\)](#) improves performance and throughput.

Components

The solution uses the following components:

- [Azure NetApp Files](#) makes it easy to migrate and run file-based applications with no code changes. This shared file-storage service is a joint development from Microsoft and NetApp, a Microsoft partner.
- [Virtual Machines](#) is an infrastructure-as-a-service (IaaS) offer. You can use Virtual Machines to deploy on-demand, scalable computing resources. Virtual Machines provides the flexibility of virtualization but eliminates the maintenance demands of physical hardware. This solution uses [Linux VMs with Oracle Database software](#).
- [Azure Virtual Network](#) is a networking service that manages virtual private networks in Azure. Through Virtual Network, Azure resources like VMs can securely communicate with each other, the internet, and on-premises networks. An Azure virtual network is like a traditional network operating in a datacenter. But an Azure virtual network also provides scalability, availability, isolation, and other benefits of the Azure infrastructure.

- [Oracle Database](#) is a multi-model database management system. It supports various data types and workloads.
- The [dNFS](#) client optimizes I/O paths between Oracle and NFS servers. As a result, it provides better performance than traditional NFS clients.

Alternatives

This solution uses Oracle Data Guard (ODG) for disaster recovery (DR), and snapshots for local replication. A few options exist, as the following sections explain.

Cross-region replication

[Cross-region replication](#) provides efficient DR across regions in Azure. Cross-region replication uses storage-based replication. It doesn't use VM resources. For more information, see [Create volume replication for Azure NetApp Files](#).

Availability sets and availability zones

ODG on Azure Virtual Machines functions like ODG in on-premises systems. But this product relies on its underlying architecture. If you run ODG on Azure VMs, consider also using one of these options to increase redundancy and availability:

- Place the Oracle VMs in the same availability set. This approach provides protection during these events:
 - Outages that equipment failures cause within a datacenter. VMs within an availability set don't share resources.
 - Updates. VMs within an availability set undergo updates at different times.
- Place the Oracle VMs in different availability zones. This approach provides protection against the failure of an entire datacenter. Each zone represents a set of datacenters within a region. If you place resources in different availability zones, datacenter-level outages can't take all your VMs offline.

You can only choose one of these options. An Azure VM can't participate in availability sets and zones at the same time. Each option has advantages:

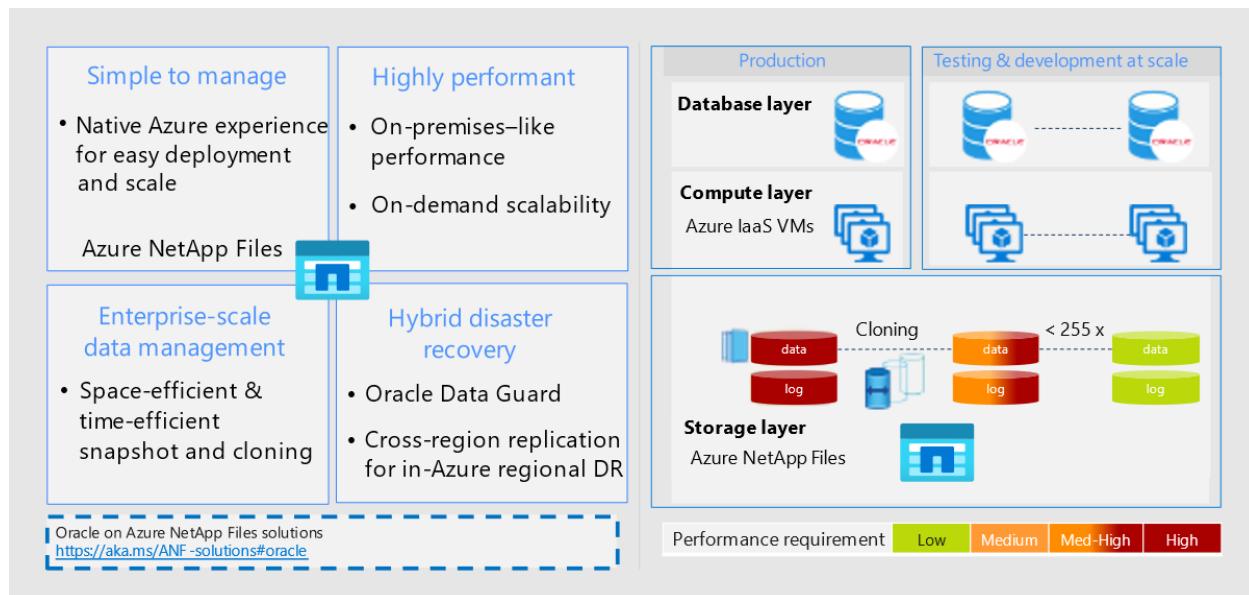
- Availability zones provide better availability than availability sets. See [SLA for Virtual Machines](#) for a comparison.
- You can place VMs that are in the same availability set in a [proximity placement group](#). This configuration minimizes the network latency between the VMs by guaranteeing that they're close to each other. In contrast, VMs that you place in different availability zones have greater network latency between them. It then

takes longer to synchronize data between the primary and secondary replicas. As a result, the primary replica may experience delays. There's also an increased chance of data loss during unplanned failovers.

After you choose a solution, test it under load. Ensure that it meets SLAs for performance and availability.

Key benefits

This image shows the benefits of using Azure NetApp Files with Oracle Database.



Download an [SVG](#) of this architecture.

Simple and reliable service

As a simple-to-consume Azure native service, Azure NetApp Files runs within the Azure datacenter environment. You can provision, consume, and scale Azure NetApp Files just like other Azure storage options. Azure NetApp Files uses reliability features that the NetApp data management software ONTAP provides. With this software, you can quickly and reliably provision enterprise-grade NFS volumes for Oracle Database and other enterprise application workloads.

Highly performant systems

Azure NetApp Files uses a bare-metal fleet of all-flash storage. Besides using shared and highly scalable storage, Azure NetApp Files provides latencies of less than 1 millisecond. These factors make this service well-suited for using the NFS protocol to run Oracle Database workloads over networks.

The Azure DCsv2-series VMs can use high-performance, all-flash NetApp storage systems. These systems are also integrated into the Azure software-defined networking (SDN) and Azure Resource Manager frameworks. As a result, you get high-bandwidth, low-latency shared storage that's comparable to an on-premises solution. The performance of this architecture meets the requirements of the most demanding, business-critical enterprise workloads. For more information on the performance benefits of Azure NetApp Files, see [Benefits of using Azure NetApp Files with Oracle Database](#).

Azure NetApp Files offers on-demand scalability. You can enlarge or reduce deployments to optimize each workload's configuration.

Enterprise-scale data management

This solution can handle workloads that require advanced data management features. ONTAP provides functionality in this area that's unmatched in the industry:

- Space-efficient, instantaneous cloning enhances development and test environments.
- On-demand capacity and performance scaling makes efficient use of resources.
- Snapshots provide database consistency points and offer these benefits:
 - They're storage efficient. You only need limited capacity to create snapshots.
 - You can quickly create, replicate, restore, or clone them. As a result, they provide backup and recovery solutions that achieve aggressive recovery time objective (RTO) and recovery point objective (RPO) SLAs.
 - They don't affect volume performance.
 - They provide scalability. You can create them frequently and store many simultaneously.

Hybrid DR

The combination of ODG and Azure NetApp Files provides DR for this architecture. Those DR solutions are appropriate for cloud and hybrid systems. Their plans work across multiple regions and with on-premises datacenters.

Considerations

The following considerations apply to this solution:

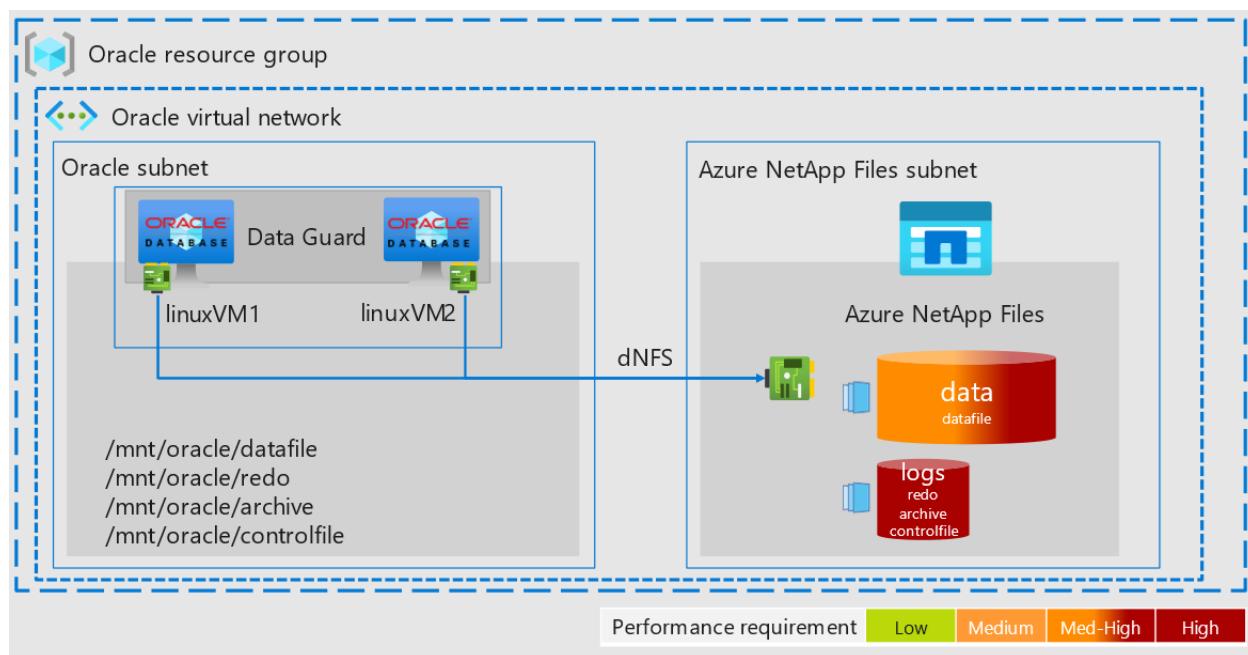
Availability

For Azure NetApp Files:

- See [SLA for Azure NetApp Files](#) for this service's availability guarantee.
- As [Enterprise-scale data management](#) discusses, you can use snapshots in backup and recovery solutions. Use Oracle hot backup mode and Azure NetApp Files APIs to orchestrate database-consistent snapshots.

When you use Oracle Database in Azure, implement a solution for HA and DR to avoid downtime:

- Use [ODG](#).
- Run the database on one virtual machine.
- Deploy a secondary VM, but only install the binaries on it.
- Put both VMs in the same virtual network. Then they can access each other over the private persistent IP address.



[Download an SVG of this architecture.](#)

Scalability

As [Highly performant systems](#) discusses, Azure NetApp Files provides built-in scalability.

Security

Azure NetApp Files secures data in many ways. For information about inherent protection, encryption, policy rules, role-based access control features, and activity logs,

see [Security FAQs](#).

Cost optimization

Using Azure NetApp Files instead of block storage can reduce costs:

- You can make the configuration cost-efficient. Traditional on-premises configurations are sized for maximum workload requirements. Consequently, these configurations are most cost-effective at maximum usage. In contrast, an Azure NetApp Files deployment is scalable. You can optimize the configuration for the current workload requirement to reduce expenses.
- You can use smaller VMs:
 - Azure NetApp Files provides low-latency storage access. With smaller VMs, you get the same performance that larger VMs deliver with ultra disk storage.
 - Cloud resources usually place limits on I/O operations. This practice prevents sudden slowdowns that resource exhaustion or unexpected outages can cause. As a result, VMs have disk throughput limitations and network bandwidth limitations. The network limitations are typically higher than disk throughput limitations. With network-attached storage, only network bandwidth limits are relevant, and they only apply to data egress. In other words, VM-level disk I/O limits don't affect Azure NetApp Files. Because of these factors, network-attached storage can achieve better performance than disk I/O. This fact is true even when Azure NetApp Files runs on smaller VMs.

Smaller VMs offer these pricing advantages over larger ones:

- They cost less.
- They carry a lower Oracle Database license cost, especially when you use smaller, constrained-code SKUs.
- The network-attached storage doesn't have an I/O cost component.

These factors make Azure NetApp Files less costly than disk storage solutions.

Deploy this scenario

- For resources on deploying Oracle Database on Azure VMs with Azure NetApp Files, see [Solution architectures using Azure NetApp Files](#).
- For information on how to deploy and access Azure NetApp Files volumes, see [Azure NetApp Files documentation](#).
- Consider the database size:

- For small databases, you can deploy all components, such as data files, the redo log, the archive log, and control files, into a single volume. Such simplified configurations are easy to manage.
- For large databases, it's more efficient to configure multiple volumes. You can use [automatic or manual Quality of Service \(QoS\) volumes](#). These volume types provide more granular control over performance requirements.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Deanna Garcia](#) | Principal Program Manager

Next steps

- [Oracle database performance on Azure NetApp Files single volumes](#)
- [Linux NFS mount options best practices for Azure NetApp Files](#)
- [Azure NetApp Files performance benchmarks for Linux](#)
- [Capacity management FAQs](#)

Related resources

Fully deployable architectures that use Azure NetApp Files:

- [Run SAP BW/4HANA with Linux virtual machines on Azure](#)
- [Run SAP NetWeaver in Windows on Azure](#)

SQL Server on Azure Virtual Machines with Azure NetApp Files

Azure NetApp Files

Azure SQL Server on Virtual Machines

Azure Virtual Machines

Azure Virtual Network

The most demanding SQL Server database workloads require very high I/O capacity. They also need low-latency access to storage. This document describes a high-bandwidth, low-latency solution for SQL Server workloads.

The solution provides shared file access with the Server Message Block (SMB) protocol. The architecture uses SQL Server on Azure Virtual Machines. It also uses Azure NetApp Files, a shared file-storage service. Azure NetApp Files provides benefits:

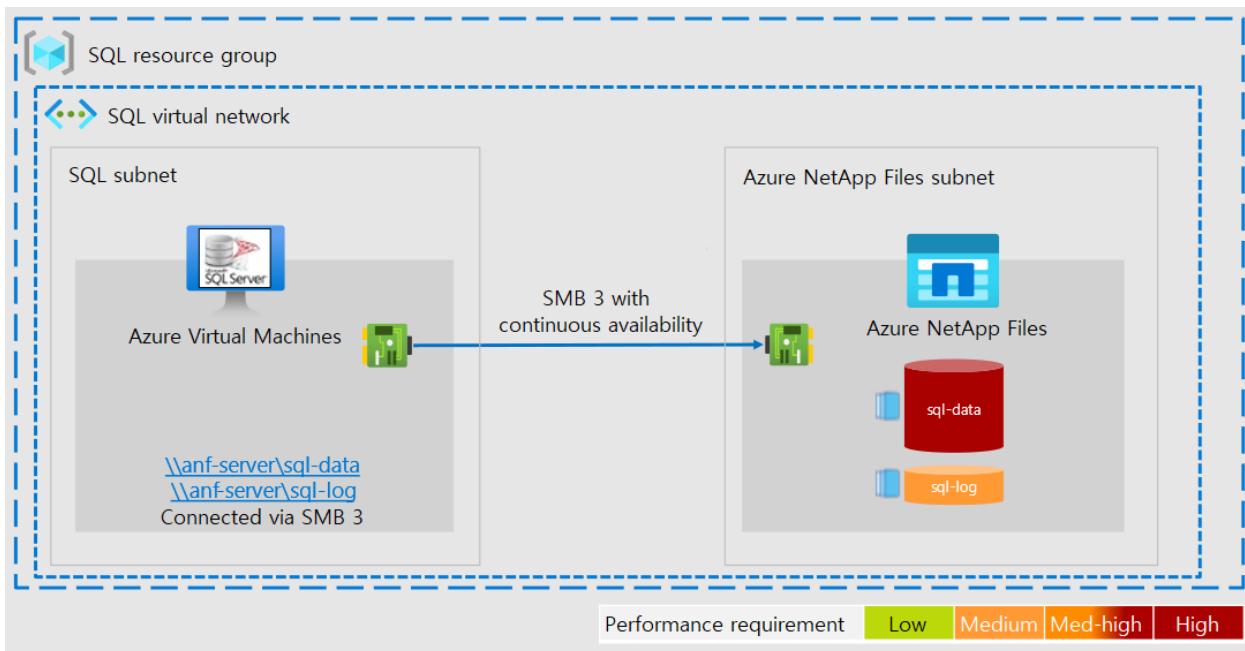
- Disk I/O limits on access rates that apply at the virtual machine (VM) level don't affect Azure NetApp Files. As a result, you can use smaller VMs than you would with disk storage without degrading performance. This approach significantly reduces costs.
- Azure NetApp Files offers flexibility. You can enlarge or reduce deployments on demand to make your configuration cost effective.

Potential use cases

This solution has many uses:

- Running new SQL Server instances that require high availability (HA) and have high standards for performance.
- Migrating highly performant, highly available SQL Server instances from on-premises infrastructure to Azure Virtual Machines.
- Using availability sets and SMB shared storage to deploy cost-effective, enterprise-scale, highly available SQL Server Always On Failover Cluster Instances.
- Deploying enterprise-scale disaster recovery (DR) architectures for hybrid or Azure systems by using SQL Server Always On availability groups.
- Cloning enterprise-scale SQL Server systems for use in test and development environments. The solution is particularly suited for cases that require advanced data management capabilities. It can help these cases meet aggressive data protection service level agreements (SLAs).

Architecture



[Download an **SVG** of this architecture.](#)

Workflow

The components interact in these ways:

- This architecture uses SQL Server on Azure Virtual Machines. With this Azure service, SQL Server runs on Azure VMs within the SQL subnet.
- In the Azure NetApp Files subnet, Azure NetApp Files provides SMB 3 access to the database and log files.
- Azure NetApp Files has the [SMB continuous availability shares option](#) turned on. This feature makes SMB Transparent Failover possible, so you can observe service maintenance events on Azure NetApp Files non-disruptively for your SQL server deployment.

Components

The solution uses the following components:

- [Azure NetApp Files](#) makes it easy to migrate and run file-based applications with no code changes. This shared file-storage service is a joint development from Microsoft and NetApp, a Microsoft partner.
- [Virtual Machines](#) is an infrastructure-as-a-service (IaaS) offer. You can use Virtual Machines to deploy on-demand, scalable computing resources. Virtual Machines provides the flexibility of virtualization but eliminates the maintenance demands of physical hardware. This solution uses Windows VMs.

- **SQL Server on Azure Virtual Machines** provides a way to migrate SQL Server workloads to the cloud with 100 percent code compatibility. As part of the Azure SQL family, this database solution runs SQL Server on VMs. SQL Server on Azure Virtual Machines offers the flexibility and hybrid connectivity of Azure. But this solution also provides the performance, security, and analytics of SQL Server. You can continue to use your current SQL Server version. You can also access the latest SQL Server updates and releases.
- **Azure Virtual Network** is a networking service that manages virtual private networks in Azure. Through Virtual Network, Azure resources like VMs can securely communicate with each other, the internet, and on-premises networks. An Azure virtual network is like a traditional network operating in a datacenter. But an Azure virtual network also provides scalability, availability, isolation, and other benefits of the Azure infrastructure.

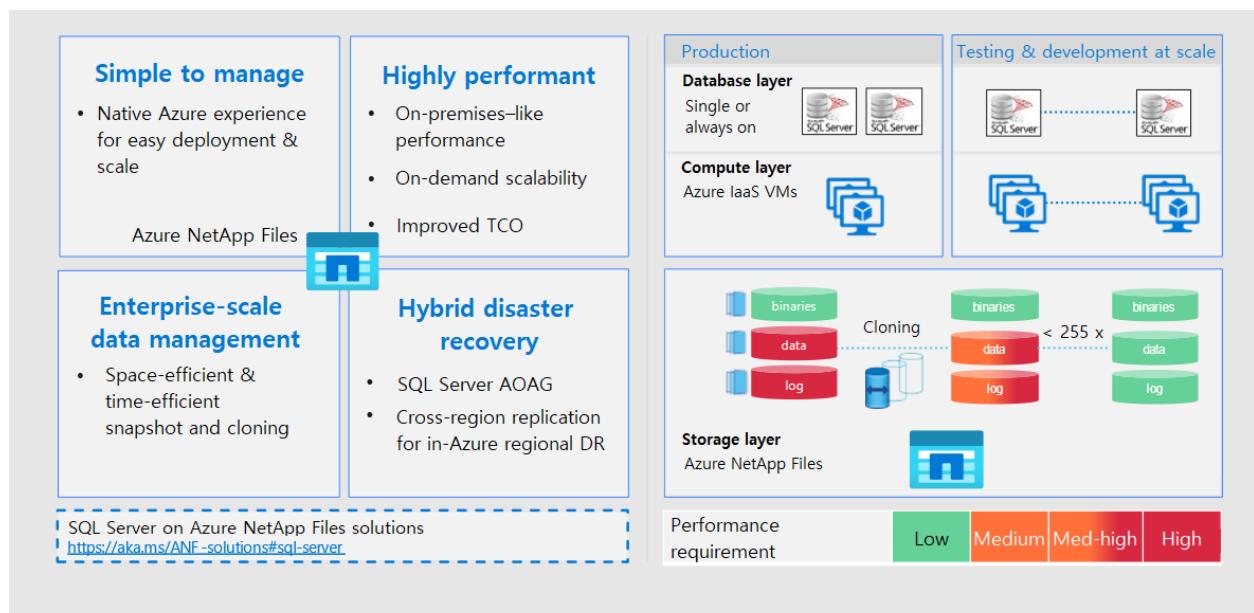
Alternatives

This solution uses Always On availability groups for DR. As an alternative, [cross-region replication](#) provides efficient DR across regions in Azure. Cross-region replication uses storage-based replication. It doesn't use VM resources. For more information, see [Create volume replication for Azure NetApp Files](#).

Scenario details

Key benefits

This image shows the benefits of using SQL Server with Azure NetApp Files.



Download an [SVG](#) of this architecture.

Simple and reliable service

As a simple-to-consume Azure native service, Azure NetApp Files runs within the Azure datacenter environment. You can provision, consume, and scale Azure NetApp Files just like other Azure storage options. Azure NetApp Files uses reliability features that the NetApp data management software ONTAP provides. With this software, you can quickly and reliably provision enterprise-grade SMB volumes for SQL Server and other workloads.

Highly performant systems

Azure NetApp Files uses a bare-metal fleet of all-flash storage. Besides using shared and highly scalable storage, Azure NetApp Files provides latencies of less than 1 millisecond. These factors make this service well suited for using the SMB protocol to run SQL Server workloads over networks.

Azure DCsv2-series VMs have built-in high-performance, all-flash ONTAP enterprise systems. These systems are also integrated in the Azure software-defined networking (SDN) and Azure Resource Manager frameworks. As a result, you get high-bandwidth, low-latency shared storage that's comparable to an on-premises solution. The performance of this architecture meets the requirements of the most demanding, business-critical enterprise workloads.

Azure NetApp Files offers on-demand scalability. You can enlarge or reduce deployments to optimize each workload's configuration.

As [Pricing](#) explains, using Azure NetApp Files instead of block storage reduces the SQL Server total cost of ownership (TCO).

Enterprise-scale data management

This solution can handle workloads that require advanced data management features. ONTAP provides functionality in this area that's unmatched in the industry:

- Space-efficient, instantaneous cloning enhances development and test environments.
- On-demand capacity and performance scaling makes efficient use of resources.
- Snapshots provide database consistency points. You can use the [NetApp SQL Server Database Quiesce Tool](#) to create application-consistent snapshots. They

provide these benefits:

- They're storage efficient. You only need limited capacity to create snapshots.
- You can quickly create, replicate, restore, or clone them. As a result, they provide backup and recovery solutions that achieve aggressive recovery time objective (RTO) and recovery point objective (RPO) SLAs.
- They don't affect volume performance.
- They provide scalability. You can create them frequently and store many simultaneously.

Hybrid DR

The combination of Always On availability groups and Azure NetApp Files provides DR for this architecture. Those DR solutions are appropriate for cloud and hybrid systems. Their plans work across multiple regions and with on-premises datacenters.

Considerations

The following considerations apply to this solution:

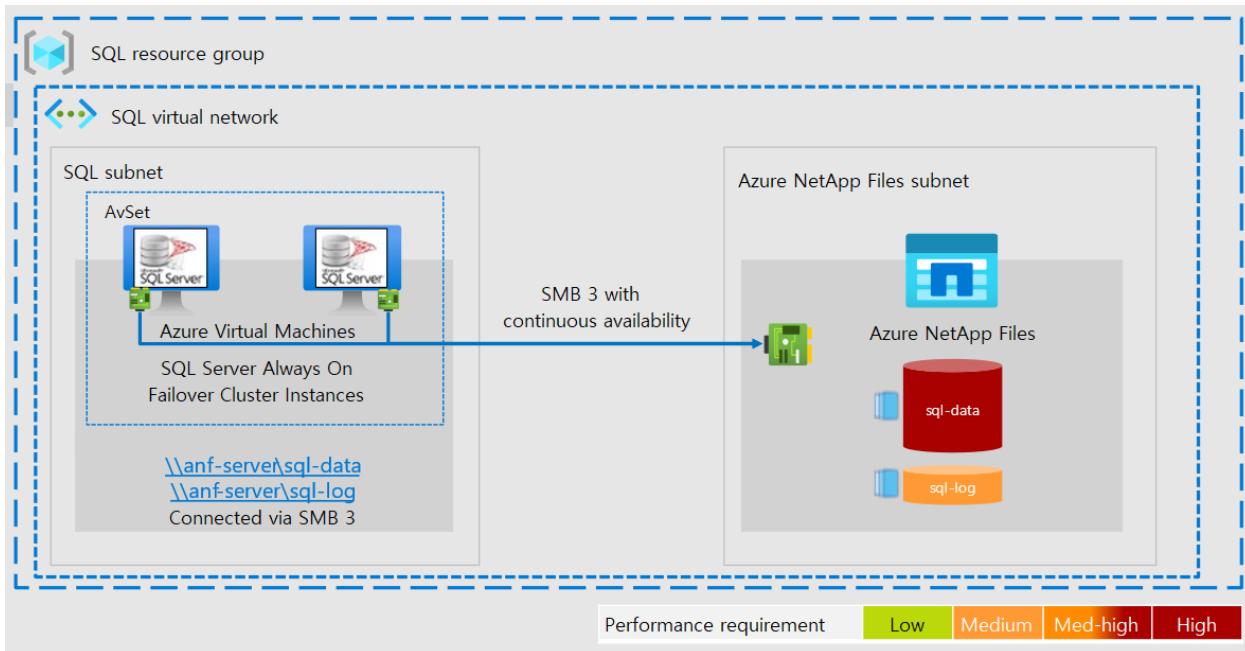
Availability

For Azure NetApp Files:

- See [SLA for Azure NetApp Files](#) for this service's availability guarantee.
- You can [convert existing SMB volumes to use Continuous Availability](#).

For SQL Server on Azure Virtual Machines, implement a solution for HA and DR to avoid downtime:

- Use an instance of [Always On Failover Cluster Instances](#) with two databases on two separate VMs.
- Put both VMs in the same virtual network. Then they can access each other over the private persistent IP address.
- Place the VMs in the same [availability set](#). Then Azure can place them in separate fault domains and upgrade domains.
- For geo-redundancy:
 - Set up the two databases to replicate between two different regions.
 - Configure [Always On availability groups](#).



[Download an **SVG** of this architecture.](#)

Scalability

- As [Highly performant systems](#) discusses, Azure NetApp Files provides built-in scalability.
- With SQL Server on Azure Virtual Machines, you can add or remove VMs when data and compute requirements change. You can also switch to a higher or lower memory-to-vCore ratio. For more information, see [VM size: Performance best practices for SQL Server on Azure VMs](#).

Security

- Azure NetApp Files secures data in many ways. For information about inherent protection, encryption, policy rules, role-based access control features, and activity logs, see [Security FAQs](#).
- SQL Server on Azure Virtual Machines also protects data. For information about encryption, access control, vulnerability assessments, security alerts, and other features, see [Security considerations for SQL Server on Azure Virtual Machines](#).

Cost optimization

Using Azure NetApp Files instead of block storage can reduce costs:

- You can make the configuration cost-efficient. Traditional on-premises configurations are sized for maximum workload requirements. Consequently, these configurations are most cost-effective at maximum usage. In contrast, an Azure

NetApp Files deployment is scalable. You can optimize the configuration for the current workload requirement to reduce expenses.

- You can use smaller VMs:
 - Azure NetApp Files provides low-latency storage access. With smaller VMs, you get the same performance that larger VMs deliver with ultra disk storage.
 - Cloud resources usually place limits on I/O operations. This practice prevents sudden slowdowns that resource exhaustion or unexpected outages can cause. As a result, VMs have disk throughput limitations and network bandwidth limitations. The network limitations are typically higher than disk throughput limitations. With network-attached storage, only network bandwidth limits are relevant, and they only apply to data egress. In other words, VM-level disk I/O limits don't affect Azure NetApp Files. Because of these factors, network-attached storage can achieve better performance than disk I/O. This fact is true even when Azure NetApp Files runs on smaller VMs.

Smaller VMs offer these pricing advantages over larger ones:

- They cost less.
- They carry a lower SQL Server license cost.
- The network-attached storage doesn't have an I/O cost component.

These factors make Azure NetApp Files less costly than disk storage solutions. For a detailed TCO analysis, see [Benefits of using Azure NetApp Files for SQL Server deployment](#).

Deploy this scenario

- For resources on deploying SQL Server on Azure NetApp Files, see [Solution architectures using Azure NetApp Files](#).
- For information on how to deploy and access Azure NetApp Files volumes, see [Azure NetApp Files documentation](#).
- Consider the database size:
 - For small databases, you can deploy database and log files into a single volume. Such simplified configurations are easy to manage.
 - For large databases, it can be more efficient to configure multiple volumes. You can also use a [manual Quality of Service \(QoS\) capacity pool](#). This type provides more granular control over performance requirements.
- Install SQL Server with SMB fileshare storage. SQL Server 2012 (11.x) and later versions support SMB file server as a storage option. Database engine user

databases and system databases like Master, Model, MSDB, and TempDB provide that support. This point applies to SQL Server stand-alone and SQL Server failover cluster installations (FCI). For more information, see [Install SQL Server with SMB fileshare storage](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Deanna Garcia](#) | Principal Program Manager

Next steps

- For information about setting up a SQL Server VM, see [Quickstart: Create SQL Server 2017 on a Windows virtual machine in the Azure portal](#).
- To learn how to migrate SQL Server to Azure while retaining application and OS control, see [Migration overview: SQL Server to SQL Server on Azure VMs](#).
- For information about SQL Server on Azure NetApp Files, see the [solutions architectures landing page](#).

Related resources

Fully deployable architectures that use Azure NetApp Files:

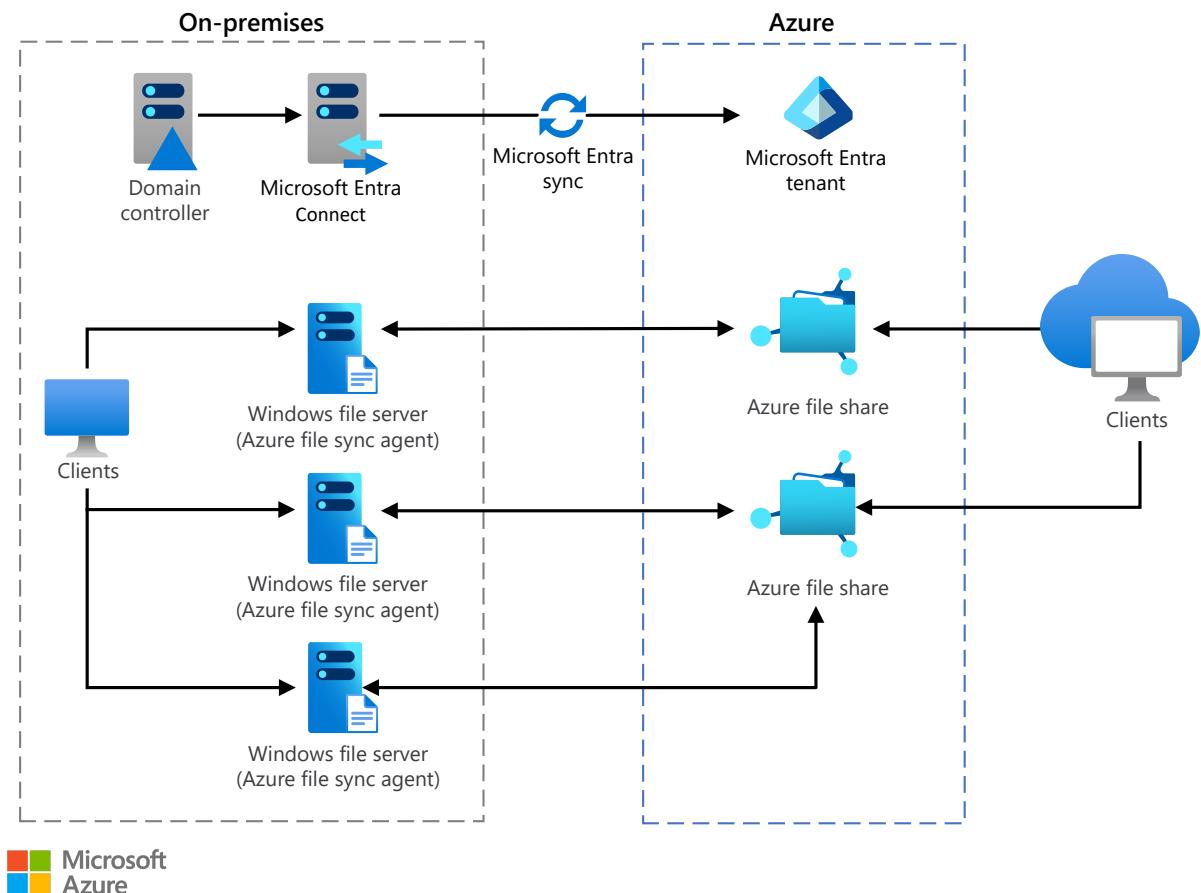
- [Run SAP BW/4HANA with Linux virtual machines on Azure](#)
- [Run SAP NetWeaver in Windows on Azure](#)

Hybrid file services

Microsoft Entra ID Azure ExpressRoute Azure Files Azure Storage Accounts

This reference architecture illustrates how to use Azure File Sync and Azure Files to extend file services hosting capabilities across cloud and on-premises file share resources.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

The architecture consists of the following components:

- **Azure storage account**. A storage account that's used to host file shares.
- **Azure Files**. A serverless cloud file share that provides the cloud endpoint of a sync relationship by using Azure File Sync. Files in an Azure file share can be accessed directly with Server Message Block (SMB) or FileREST protocol.

- **Sync groups.** Logical groupings of Azure file shares and servers that run Windows Server. Sync groups are deployed into Storage Sync Service, which registers servers for use with Azure File Sync and contains the sync group relationships.
- **Azure File Sync agent.** This is installed on Windows Server machines to enable and configure sync with cloud endpoints.
- **Windows Servers.** On-premises or cloud-based Windows Server machines that host a file share that syncs with an Azure file share.
- **Microsoft Entra ID.** The Microsoft Entra tenant that's used for identity synchronization across Azure and on-premises environments.

Components

- [Azure storage accounts](#)
- [Azure Files](#)
- [Microsoft Entra ID](#)

Scenario details

Typical uses for this architecture include:

- Hosting file shares that need to be accessible from cloud and on-premises environments.
- Synchronizing data between multiple on-premises data stores with a single cloud-based source.

Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a requirement that overrides them.

Azure Files usage and deployment

You store your files in the cloud in serverless Azure file shares. You can use them in two ways: by directly mounting them (SMB) or by caching them on-premises by using Azure File Sync. What you need to consider as you plan for your deployment depends on which of the two ways that you choose.

- Direct mount of an Azure file share. Because Azure Files provides SMB access, you can mount Azure file shares on-premises or in the cloud using the standard SMB client available in the Windows, macOS, and Linux operating systems. Azure file

shares are serverless, so deploying them for production scenarios doesn't require managing a file server or network-attached storage (NAS) device. This means you don't have to apply software patches or swap out physical disks.

- Cache Azure file share on-premises with Azure File Sync. Azure File Sync makes it possible for you to centralize your organization's file shares in Azure Files, while keeping the flexibility, performance, and compatibility of an on-premises file server. Azure File Sync transforms an on-premises (or cloud) Windows Server into a quick cache of your Azure file share.

Deploy the Storage Sync Service

Begin Azure File Sync deployment by deploying a Storage Sync Service resource into a resource group of your selected subscription. We recommend provisioning as few Storage Sync Service objects as possible. You'll create a trust relationship between your servers and this resource, and a server can only be registered to one Storage Sync Service. As a result, we recommend that you deploy as many Storage Sync Services as you need to separate groups of servers. Keep in mind that servers from different Storage Sync Services can't sync with each other.

Registering Windows Server machines with the Azure File Sync agent

To enable the sync capability on Windows Server, you must install the Azure File Sync downloadable agent. The Azure File Sync agent provides two main components:

- `FileSyncSvc.exe`. The background Windows service that's responsible for monitoring changes on the server endpoints and for initiating sync sessions.
- `StorageSync.sys`. A file system filter that enables cloud tiering and faster disaster recovery.

You can download the agent from the [Azure File Sync Agent Download](#) page at the Microsoft Download Center.

Operating system requirements

Azure File Sync is supported by the Windows Server versions that are listed in the following table.

[] [Expand table](#)

Version	Supported SKUs	Supported deployment options
Windows Server 2019	Datacenter, Standard, and IoT	Full and Core
Windows Server 2016	Datacenter, Standard, and Storage Server	Full and Core
Windows Server 2012 R2	Datacenter, Standard, and Storage Server	Full and Core

For more information, see [Windows file server considerations](#).

Configuring sync groups and cloud endpoints

A *sync group* defines the sync topology for a set of files. Endpoints within a sync group are kept in sync with each other. A sync group must contain one *cloud endpoint*, which represents an Azure file share, and one or more server endpoints. A *server endpoint* represents a path on a registered server. A server can have server endpoints in multiple sync groups. You can create as many sync groups as you need to appropriately describe your desired sync topology.

A *cloud endpoint* is a pointer to an Azure file share. All server endpoints will sync with a cloud endpoint, making the cloud endpoint the hub. The storage account for the Azure file share must be located in the same region as the Storage Sync Service. The entirety of the Azure file share is synced, with one exception: a special folder, comparable to the hidden **System Volume Information** folder on an NT file system (NTFS) volume, is provisioned. This directory is called **.SystemShareInformation**, and it contains important sync metadata that doesn't sync to other endpoints.

Configuring server endpoints

A server endpoint represents a specific location on a registered server, such as a folder on a server volume. A server endpoint must be a path on a registered server (rather than a mounted share), and must use cloud tiering. The server endpoint path must be on a non-system volume. NAS isn't supported.

Azure File Share to Windows file share relationships

You should deploy Azure file shares one-to-one with Windows file shares wherever possible. The server endpoint object gives you a great degree of flexibility on how you set up the sync topology on the server-side of the sync relationship. To simplify management, make the path of the server endpoint match the path of the Windows file share.

Use as few Storage Sync Services as possible. This simplifies management when you have sync groups that contain multiple server endpoints, because a Windows Server can only be registered to one Storage Sync Service at a time.

Pay attention to I/O operations per second (IOPS) limitations on a storage account when you deploy Azure file shares. The ideal is to map file shares one-to-one with storage accounts. It isn't always possible to do that because of various limits and restrictions from your organization and from Azure. When it's not possible to have only one file share deployed in a storage account, ensure that your most active file shares aren't in the same storage account.

Topology recommendations: firewalls, edge networks, and proxy connectivity

Consider the following recommendations for solution topology.

Firewall and traffic filtering

Based on the policies of your organization or on unique regulatory requirements, you might need to restrict communication with Azure. Therefore, Azure File Sync provides several mechanisms for configuring networking. Based on your requirements, you can:

- Tunnel the sync and file upload and download traffic over your Azure ExpressRoute or Azure virtual private network (VPN).
- Make use of Azure Files and Azure networking features such as service endpoints and private endpoints.
- Configure Azure File Sync to support your proxy in your environment.
- Throttle network activity from Azure File Sync.

To learn more about Azure File Sync and networking, see [Azure File Sync networking considerations](#).

Configuring proxy servers

Many organizations use a proxy server as an intermediary between resources inside their on-premises network and resources outside their network, such as in Azure. Proxy

servers are useful for many applications, such as network isolation and security, and monitoring and logging. Azure File Sync can interoperate fully with a proxy server; however, you must manually configure the proxy endpoint settings for your environment with Azure File Sync. You do this by using the Azure File Sync server cmdlets in Azure PowerShell.

For more information on how to configure Azure File Sync with a proxy server, see [Azure File Sync proxy and firewall settings](#).

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework,

which is a set of guiding tenets that can be used to improve the quality of a workload.

For more information, see [Microsoft Azure Well-Architected Framework](#).

Reliability

Reliability ensures that your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

- You should consider the type and performance of the storage account that you use to host Azure file shares. All storage resources that are deployed into a storage account share the limits that apply to that storage account. To find out more about determining the current limits for a storage account, see [Azure Files scalability and performance targets](#).
- There are two main types of storage accounts for Azure Files deployments:
 - General purpose version 2 (GPv2) storage accounts. GPv2 storage accounts allow you to deploy Azure file shares on standard, hard disk-based (HDD-based) hardware. In addition to storing Azure file shares, GPv2 storage accounts can store other storage resources such as blob containers, queues, and tables.
 - FileStorage storage accounts: FileStorage storage accounts make it possible for you to deploy Azure file shares on premium, solid-state disk-based (SSD-based) hardware. FileStorage accounts can only be used to store Azure file shares. You can't deploy other storage resources such as blob containers, queues, and tables in a FileStorage account.
- By default, standard file shares can span no more than 5 terabytes (TiB), although the share limit can be increased to 100 TiB. To do this, the large file share feature must be enabled at the storage-account level. Premium storage accounts (FileStorage storage accounts) don't have the large file share feature flag, because all premium file shares are already enabled for provisioning up to the full 100-TiB capacity. You can only enable large file shares on locally redundant or zone-

redundant standard storage accounts. After you enable the large file share feature flag, you can't change the redundancy level to geo-redundant storage or geo-zone-redundant storage. To enable large file shares on an existing storage account, enable the **Large file share** option in the **Configuration** view of the associated storage account.

- You should ensure that Azure File Sync is supported in the regions where you deploy your solution. For more information, see [Azure file sync region availability](#).
- You should ensure that the services that are referenced in the **Architecture** section are supported in the region where you deploy the hybrid file services architecture.
- To protect the data in your Azure file shares against data loss or corruption, all Azure file shares store multiple copies of each file as it's written. Depending on the requirements of your workload, you can select more degrees of redundancy.
- *Previous Versions* is a Windows feature that enables you to use server-side Volume Shadow Copy Service (VSS) snapshots of a volume to present restorable versions of a file to an SMB client. VSS snapshots and Previous Versions work independently of Azure File Sync. However, cloud tiering must be set to a compatible mode. Many Azure File Sync server endpoints can exist on the same volume. You have to make the following PowerShell call per volume that has even one server endpoint, where you plan to or are using cloud tiering. For more information about Previous Versions and VSS, see [Self-service restore through Previous Versions and VSS \(Volume Shadow Copy Service\)](#).

Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

- Azure File Sync works with your standard Active Directory Domain Services (AD DS) identity without any special setup beyond setting up Azure File Sync. When you use Azure File Sync, file access typically goes through the Azure File Sync caching servers rather than through the Azure file share. Because the server endpoints are located on Windows Server machines, the only requirement for identity integration is to use domain-joined Windows file servers to register with the Storage Sync Service. Azure File Sync stores access control lists (ACLs) for the files in the Azure file share, and replicates them to all server endpoints.
- Even though changes that are made directly to the Azure file share take longer to sync to the server endpoints in the sync group, you might want to ensure that you can enforce your AD DS permissions on your file share directly in the cloud also. To do this, you must domain join your storage account to your on-premises AD DS domain, just as your Windows file servers are domain joined. To learn more about

domain joining your storage account to a customer-owned AD DS instance, see [Overview of Azure Files identity-based authentication options for SMB access](#).

- When you use Azure File Sync, there are three different layers of encryption to consider:
 - Encryption at rest for data that's stored in Windows Server. There are two strategies for encrypting data on Windows Server that work generally with Azure File Sync: encryption beneath the file system such that the file system and all of the data written to it is encrypted, and encryption within the file format itself. These methods can be used together if desired, because their purposes differ.
 - Encryption in transit between the Azure File Sync agent and Azure. Azure File Sync agent communicates with your Storage Sync Service and Azure file share by using the Azure File Sync REST protocol and the FileREST protocol, both of which always use HTTPS over port 443. Azure File Sync doesn't send unencrypted requests over HTTP.
 - Encryption at rest for data that's stored in the Azure file share. All data that's stored in Azure Files is encrypted at rest using Azure storage service encryption (SSE). Storage service encryption works much like BitLocker on Windows: data is encrypted beneath the file system level. Because data is encrypted beneath the file system of the Azure file share as the data is encoded to disk, you don't need access to the underlying key on the client to read or write to the Azure file share.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- Consult the [Principles of cost optimization](#) page in the Azure Well-Architected Framework for cost optimization recommendations.
- The [Azure Storage Pricing](#) page provides detailed pricing information based on account type, storage capacity, replication, and transactions.
- The [Data Transfers Pricing Details](#) article provides detailed pricing information for data egress.
- You can use the [Azure Storage Pricing Calculator](#) to help estimate your costs.

Operational Excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational](#)

excellence pillar.

- The Azure File Sync agent is updated on a regular basis to add new functionality and to address issues. Microsoft recommends that you configure Microsoft Update to provide updates for the Azure File Sync agent as they become available. For more information, see [Azure File Sync agent update policy](#).
- Azure Storage offers soft delete for file shares so that you can recover your data when it's mistakenly deleted by an application or by another storage account user. To learn more about soft delete, see [Enable soft delete on Azure file shares](#).
- Cloud tiering is an optional feature of Azure File Sync that caches frequently accessed files locally on the server and tiers the others to Azure Files based on policy settings. When a file is tiered, the Azure File Sync file system filter (StorageSync.sys) replaces the file locally with a pointer to the file in Azure Files. A tiered file has both the **offline** attribute and the **FILE_ATTRIBUTE_RECALL_ON_DATA_ACCESS** attribute set in NTFS so that third-party applications can securely identify tiered files. For more information, see [Cloud Tiering Overview](#).

Next steps

- [What is Azure File Sync?](#)
- [How is Azure File Sync billed?](#)
- [How to plan for Azure File Sync Deployment?](#)
- [How to deploy Azure File Sync?](#)
- [Azure File Sync network consideration.](#)
- [What is Cloud Tiering?](#)
- [What disaster recovery option are available in Azure File Sync?](#)
- [How to backup Azure File Sync?](#)

Related resources

Related hybrid guidance:

- [Hybrid architecture design](#)
- [Azure hybrid options](#)
- [Hybrid app design considerations](#)
- [Deploy a hybrid app with on-premises data that scales cross-cloud](#)

Related architectures:

- [Azure enterprise cloud file share](#)

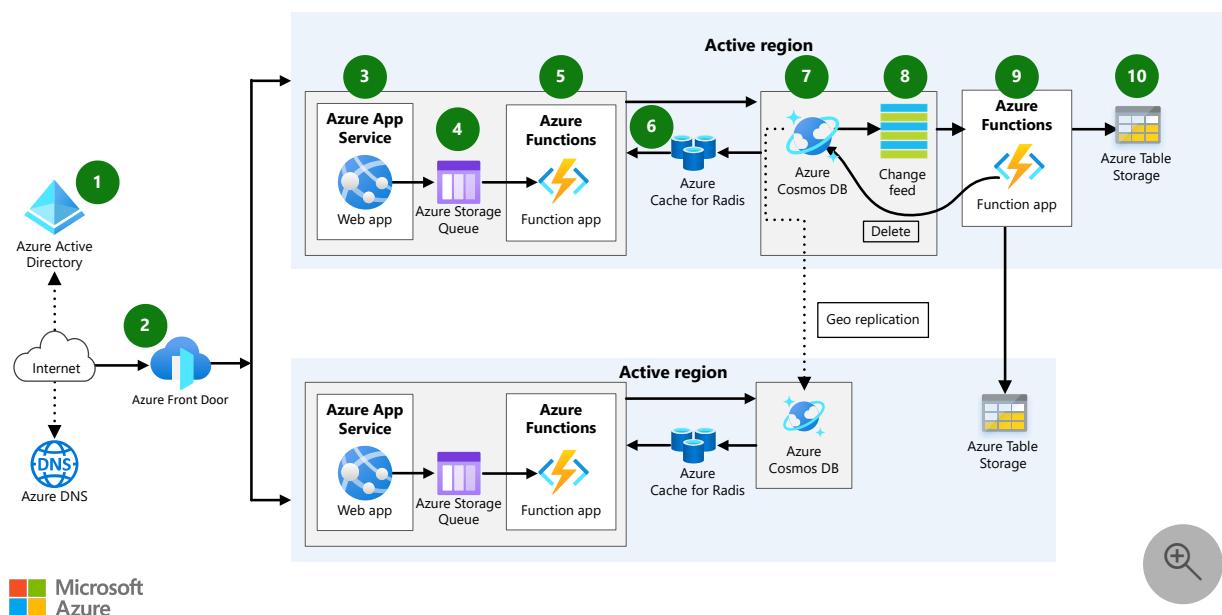
- Azure Files accessed on-premises and secured by AD DS
- Hybrid file share with disaster recovery for remote and local branch workers
- Run containers in a hybrid environment
- Use Azure file shares in a hybrid environment

Minimal storage – change feed to replicate data

Azure Front Door Azure App Service Azure Functions Azure Cosmos DB Azure Table Storage

This article presents a high-availability solution for a web application that uses massive amounts of data that must be available for a specific time period. It stores the data in Azure Cosmos DB, and uses the Azure Cosmos DB change feed to replicate the data to secondary storage. After the specified time period elapses, Azure Functions is used to delete the data from Azure Cosmos DB.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. The client authenticates with Microsoft Entra ID and is granted access to web applications hosted on Azure App Service.
2. Azure Front Door, a firewall and layer-7 load balancer, switches user traffic to the standby region if there's a regional outage.
3. App Service hosts websites and RESTful web APIs. Browser clients run AJAX applications that use the APIs.

4. Web APIs delegate responsibility to code hosted by Functions to handle background tasks. The tasks are queued in Azure Queue Storage queues.
5. The queued messages trigger the functions, which perform the background tasks.
6. Azure Cache for Redis caches database data for the functions. By using the cache, the solution offloads database activity and speeds up the function apps and web apps.
7. Azure Cosmos DB holds recently generated data.
8. Azure Cosmos DB issues a change feed that can be used to replicate changes.
9. A function app reads the change feed and replicates the changes to Azure Table Storage tables. Another function app periodically removes expired data from Azure Cosmos DB.
10. Table Storage provides low-cost storage.

Components

- [Azure Microsoft Entra ID](#) is a multi-tenant identity and access management service that can synchronize with an on-premises directory.
- [Azure DNS](#) is a high-availability hosting service for DNS domains that provides apps with fast DNS queries and quick updates to DNS records. Managing Azure DNS is like managing other Azure services, and uses the same credentials, APIs, tools, and billing.
- [Azure Front Door](#) is a secure content delivery network (CDN) and load balancer with instant failover. It operates at the edge close to users, accelerating content delivery while protecting apps, APIs, and websites from cyber threats.
- [App Service](#) is a fully managed service for building, deploying, and scaling web apps. You can build apps using .NET, .NET Core, Node.js, Java, Python, or PHP. Apps can run in containers or on Windows or Linux. In a mainframe migration, the front-end screens or web interface can be coded as HTTP-based REST APIs. They can be segregated and can be stateless to orchestrate a microservices-based system. For more information on web APIs, see [RESTful web API design](#).
- [Functions](#) provides an environment for running small pieces of code, called functions, without having to establish an application infrastructure. You can use it to process bulk data, integrate systems, work with Internet of Things (IoT) devices, and build simple APIs and microservices. With microservices, you can create servers that connect to Azure services and are always up to date.
- [Azure Storage](#) is a set of massively scalable and secure cloud services for data, apps, and workloads. It includes [Azure Files](#), [Table Storage](#), and [Queue Storage](#). Azure Files is often an effective tool for migrating mainframe workloads.
- [Queue Storage](#) provides simple, cost-effective, durable message queueing for large workloads.

- [Table Storage](#) is a NoSQL key-value store for rapid development that uses massive semi-structured datasets. The tables are schemaless and adapt readily as needs change. Access is fast and cost-effective for many types of applications, and typically costs less than other types of keyed storage.
- [Azure Cache for Redis](#) is a fully managed in-memory caching service and message broker for sharing data and state among compute resources. It includes both the open-source Redis and a commercial product from Redis Labs as managed services. You can improve the performance of high-throughput online transaction processing applications by designing them to scale and to make use of an in-memory data store such as Azure Cache for Redis.
- [Azure Cosmos DB](#) is a globally distributed, multi-model database from Microsoft that enables your solutions to elastically and independently scale throughput and storage across any number of geographic regions. It offers throughput, latency, availability, and consistency guarantees with comprehensive service level agreements (SLAs).

Alternatives

- [Azure Traffic Manager](#) directs incoming DNS requests across the global Azure regions based on your choice of traffic routing methods. It also provides automatic failover and performance routing.
- [Azure Content Delivery Network](#) caches static content in edge servers for quick response, and uses network optimizations to improve response for dynamic content. Content Delivery Network is especially useful when the user base is global.
- [Azure Kubernetes Service \(AKS\)](#) is a fully managed Kubernetes service for deploying and managing containerized applications. You can use it to implement a microservices architecture whose components scale independently on demand.
- [Azure Container Instances](#) provides a quick and simple way to run tasks without having to manage infrastructure. It's useful during development or for running unscheduled tasks.
- [Azure Service Fabric](#) is a platform for scaling and orchestrating containers and microservices.
- [Azure Service Bus](#) is a reliable cloud messaging service for simple hybrid integration. It can be used instead of Queue Storage in this architecture. For more information, see [Storage queues and Service Bus queues - compared and contrasted](#).

Scenario details

This solution uses Azure Cosmos DB to store the large volume of data that the web application uses. Web apps that handle massive amounts of data benefit from the ability of Azure Cosmos DB to elastically and independently scale throughput and storage.

Another key solution component is the Azure Cosmos DB change feed. When changes are made to the database, the change feed stream is sent to an event-driven Functions trigger. A function then runs and replicates the changes to Table Storage tables, which provide a low-cost storage solution.

The web app needs the data for only a limited amount of time. The solution takes advantage of that fact to further reduce costs. Specifically, another function periodically runs and deletes expired data from Azure Cosmos DB. Besides being triggered, functions can also be scheduled to run at set times.

Potential use cases

The architecture is appropriate for any application that:

- Uses a massive amount of data.
- Requires that data is always available when it's needed.
- Uses data that expires.

Examples include apps that:

- Track customer spending habits and shopping behavior.
- Forecast weather.
- Offer smart traffic systems or implement smart traffic systems or use smart technology to monitor traffic.
- Analyze manufacturing IoT data.
- Display smart meter data or use smart technology to monitor meter data.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

- When you implement and maintain this solution, you incur extra costs.
- Using the change feed for replication requires less code maintenance than doing the replication in the core application.
- You need to migrate existing data. The migration process requires ad-hoc scripts or routines to copy old data to storage accounts. When you migrate the data,

make sure that you use time stamps and copy flags to track migration progress.

- To avoid deleting entries from the Azure Table secondary storage, ignore delete feeds that are generated when your functions delete entries from Azure Cosmos DB.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Nabil Siddiqui](#) | Cloud Solution Architect - Digital and Application Innovation

Next steps

- Web-Queue-Worker architecture style
- Design a geographically distributed application
- Distribute your data globally with Azure Cosmos DB
- Choose the appropriate API for Azure Cosmos DB
- Store and access NoSQL data with Azure Cosmos DB for Table
- Work with NoSQL data in Azure Cosmos DB
- How to model and partition data on Azure Cosmos DB using a real-world example
- Options to migrate your on-premises or cloud data to Azure Cosmos DB
- Migrate hundreds of terabytes of data into Azure Cosmos DB
- Change feed design patterns in Azure Cosmos DB
- Serverless event-based architectures with Azure Cosmos DB and Azure Functions
- Introduction to Azure Data Factory
- Orchestrate data movement and transformation in Azure Data Factory or Azure Synapse Pipeline

Related resources

- [Build scalable database solutions with Azure services](#)
- [RESTful web API design](#)

Multi-region web application with custom Storage Table replication

Azure Front Door Azure App Service Azure Functions Azure Table Storage Azure Cache for Redis

This architecture provides a high availability solution for a web application that uses massive amounts of data. It's a flexible approach that can provide a global solution that distributes applications and data to keep it close to users.

The architecture requires custom replication software. depending on the applications and the configuration, this can be challenging to create.

Here are some possible configurations:

- **Active/Passive:** There's a primary region that normally provides service to all users. There's also a standby region that becomes active when the primary region can't function. When the primary system is active, a replication service replicates database changes to the standby region.
- **Active/Active:** There's a primary region that normally is active, providing read service to nearby users and write service to all users. One or more other regions are active and provide read-only service to nearby users. Writes are always directed to the primary region, and reads are always directed to the nearest active region.

As with the Active/Passive configuration, there's a standby region that becomes active when the primary region can't function. When the primary system is active, a replication service replicates database changes to the read-only regions and the standby region. When the standby region is active, the replication service replicates database changes to the read-only regions.

One drawback to this approach is the high latency of write operations.

- **Multi-active:** There are multiple active regions, each capable of providing full service to users. User activity is always directed to the nearest active region.

Implementation of multi-active is quite challenging, and can require expert design and implementation.

Since replication is a custom implementation, the consistency level can be whatever is needed.

The possible difficulty of implementing custom replication and the time required are important considerations with this architecture.

① Note

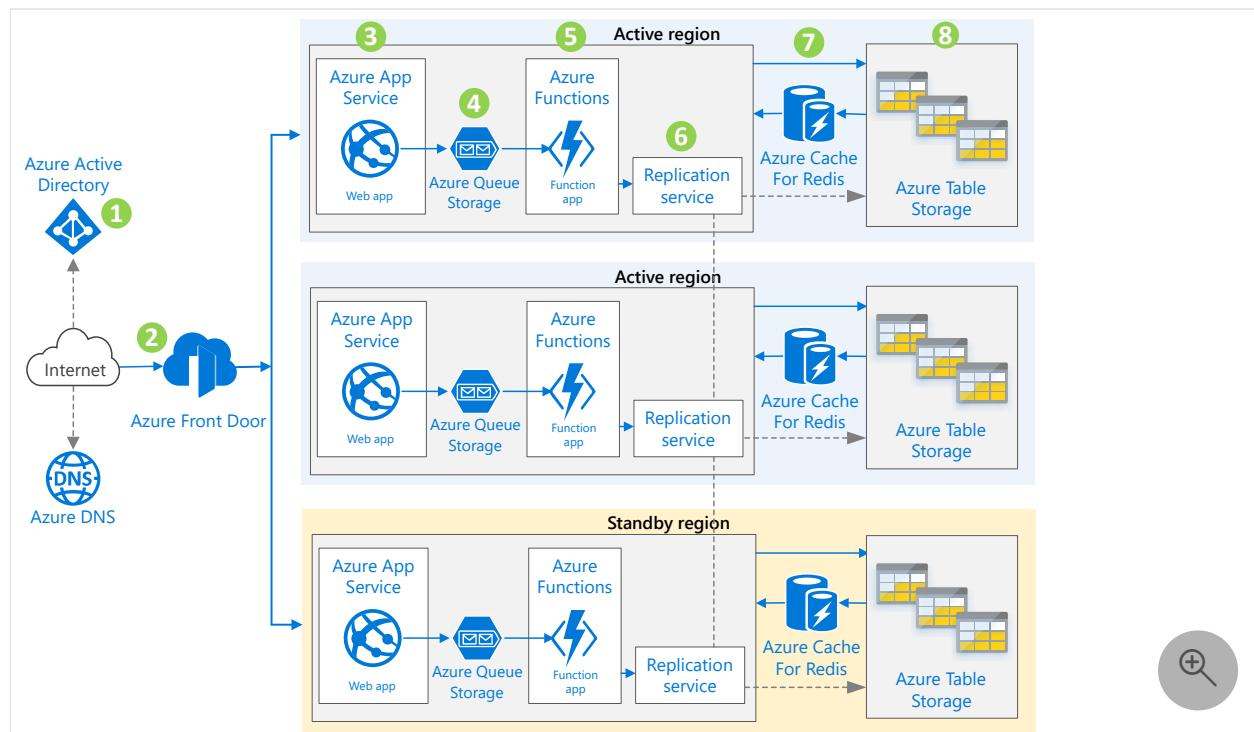
Your application may require multiple storage accounts under some circumstances. See [Considerations](#) for more information.

Potential use cases

The architecture may be appropriate for any application that uses massive amounts of data that must always be available. Examples include apps that:

- Track customer spending habits and shopping behavior (retail industry).
- Forecast weather (agriculture, environment, and media/news industries).
- Offer smart traffic systems or implement smart traffic systems or use smart technology to monitor traffic (automotive and transportation industries).
- Analyze manufacturing Internet of Things (IoT) data.
- Display smart meter data or use smart technology to monitor meter data (energy industry).

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. The client authenticates with Microsoft Entra ID and is granted access to web applications hosted on Azure App Service.
2. Azure Front Door, a firewall and layer 7 load balancer, switches user traffic to a different Azure region if there is a regional outage.
3. Azure App Service hosts websites and RESTful web APIs. Browser clients run AJAX applications that use the APIs.
4. Web APIs delegate function apps to handle background tasks. The tasks are queued in Azure Queue Storage queues.
5. The function apps hosted by Azure Functions perform the background tasks, triggered by the queued messages.
6. The custom replication software assures tables remain identical among regions.
7. Azure Cache for Redis caches table data for the function apps. This offloads database activity and speeds up the function apps and web apps.
8. Azure Table Storage holds the data used by the web applications.

Components

- [Microsoft Entra ID](#) is a multi-tenant identity and access management service that can synchronize with an on-premises directory. [Azure DNS](#) is a high-availability hosting service for DNS domains that provides apps with fast DNS queries and quick updates to DNS records. Managing Azure DNS is like managing other Azure services, and uses the same credentials, APIs, tools, and billing.
- [Azure Front Door](#) is a secure content delivery network (CDN) and load balancer with instant failover. It operates at the edge close to users, accelerating content delivery while protecting apps, APIs, and websites from cyber threats.
- [Azure App Service](#) is a fully managed service for building, deploying, and scaling web apps. You can build apps using .NET, .NET Core, Node.js, Java, Python, or PHP. Apps can run in containers or on Windows or Linux. In a mainframe migration, the front-end screens or web interface can be coded as HTTP-based REST APIs. They can be segregated and can be stateless to orchestrate a microservices-based system. For more information on web APIs, see [RESTful web API design](#).
- [Azure Functions](#) provides an environment for running small pieces of code, called functions, without having to establish an application infrastructure. You can use it to process bulk data, integrate systems, work with IoT, and build simple APIs and microservices. With microservices, you can create servers that connect to Azure services and are always up to date.
- [Azure Storage](#) is a set of massively scalable and secure cloud services for data, apps, and workloads. It includes [Azure Files](#), [Azure Table Storage](#), and [Azure](#)

[Queue Storage](#) . Azure Files is often an effective tool for migrating mainframe workloads.

- [Azure Queue Storage](#) provides simple, cost-effective, durable message queueing for large workloads.
- [Azure Table Storage](#) is a NoSQL key-value store for rapid development that uses massive semi-structured datasets. The tables are schemaless and adapt readily as needs change. Access is fast and cost-effective for many types of applications, and typically costs less than other types of keyed storage.
- [Azure Cache for Redis](#) is a fully managed in-memory caching service and message broker for sharing data and state among compute resources. It includes both the open-source Redis and a commercial product from Redis Labs as managed services. You can improve performance of high-throughput online transaction processing applications by designing them to scale and to make use of an in-memory data store such as Azure Cache for Redis.

Alternatives

- [Azure Traffic Manager](#) directs incoming DNS requests across the global Azure regions based on your choice of traffic routing methods. It also provides automatic failover and performance routing.
- [Azure Content Delivery Network](#) (CDN) caches static content in edge servers for quick response, and uses network optimizations to improve response for dynamic content. CDN is especially useful when the user base is global.
- [Azure Kubernetes Service \(AKS\)](#) is a fully managed Kubernetes service for deploying and managing containerized applications. You can use it to implement a microservices architecture whose components scale independently on demand.
- [Azure Container Instances](#) provides a quick and simple way to run tasks without having to manage infrastructure. It's useful during development or for running unscheduled tasks.
- [Azure Service Fabric](#) is a platform for scaling and orchestrating containers and microservices.
- [Azure Service Bus](#) is a reliable cloud messaging service for simple hybrid integration. It can be used instead of Queue Storage in this architecture. For more information, see [Storage queues and Service Bus queues - compared and contrasted](#).

Considerations

- The architecture requires custom replication software. This can be challenging to create, depending on the applications and the configuration. The possible difficulty

of implementing custom replication and the time required are important considerations with this architecture.

- Because replication is custom-designed, developers have great flexibility in implementing a data consistency strategy.
- There are performance limits on Table Storage that can be overcome by adding Storage accounts. The following circumstances may require more accounts:
 - To implement multi-tenancy to support multiple customers
 - To support customers with higher transaction rates
 - To support customers with large datasets
 - To speed up data access by distributing data across multiple storage accounts
 - To segregate data into hot, cold, and archive tiers
 - To make copies of data for backup and reporting purposes

For more information, see [Scalability and performance targets for Table Storage](#).

- If your application already contains data, then you need to write routines to copy old data to storage accounts. Make sure that you have timestamp and copy flags to track the progress of migration of data.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

- [Nabil Siddiqui](#) | Cloud Solution Architect - Digital and Application Innovation

Next steps

- [Guidelines for table design](#)
- [Use geo-redundancy to design highly available applications](#)

Related resources

- [Web-Queue-Worker architecture style](#)
- [Data partitioning strategies](#)
- [Build scalable database solutions with Azure services](#)
- [RESTful web API design](#)

HIPAA and HITRUST compliant health data AI

Azure Blob Storage

Azure Event Grid

Azure Functions

Azure Machine Learning

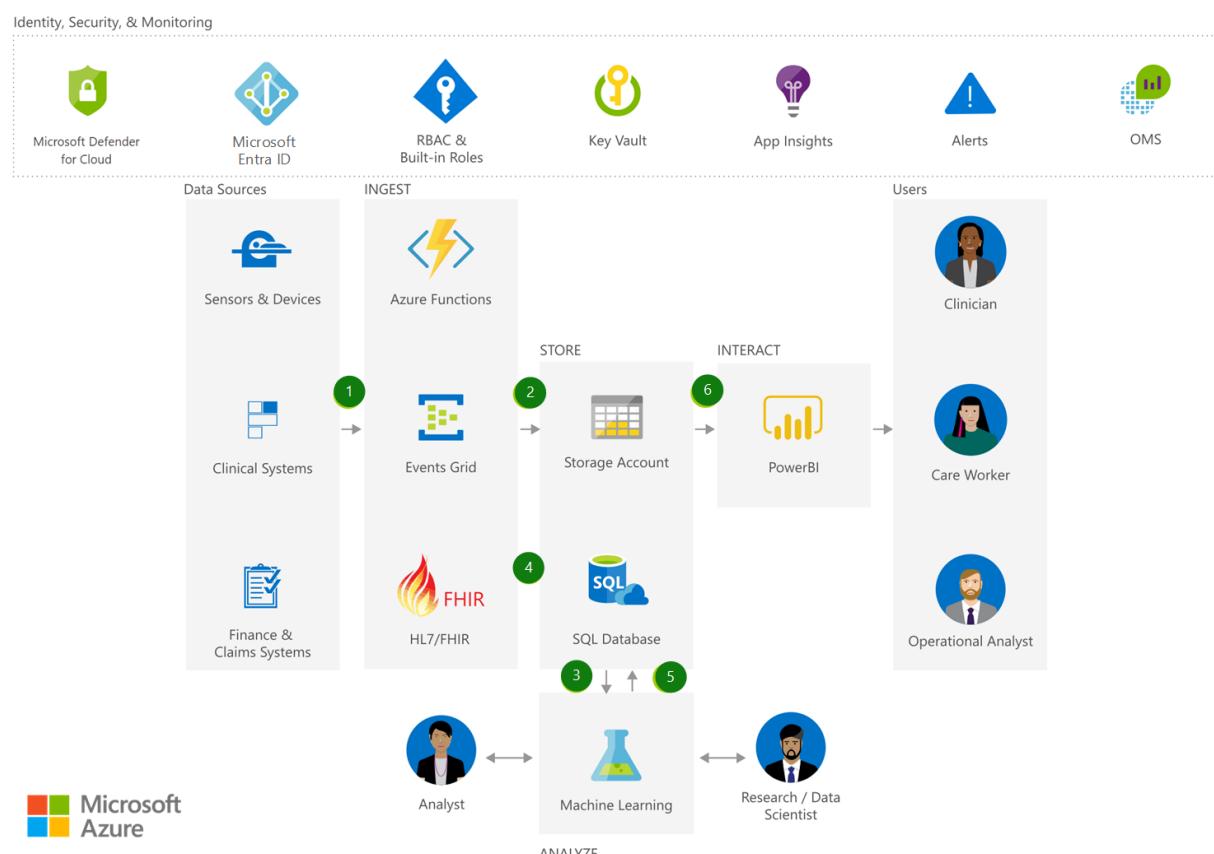
Power BI

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article describes how you can store, manage, and analyze HIPAA-compliant and HITRUST-compliant health data and medical records with a high level of built-in security.

Architecture



Download an [SVG](#) of this architecture.

Dataflow

1. Securely ingest bulk patient data into [Azure Blob storage](#).
2. [Event Grid](#) publishes patient data to [Azure Functions](#) for processing, and securely stores patient data in [SQL Database](#).
3. Analyze patient data using [Machine Learning](#), and create a Machine Learning-trained model.
4. Ingest new patient data in HL7/FHIR format and publish to [Azure Functions](#) for processing. Store in [SQL Database](#).
5. Analyze newly ingested data using the trained Machine Learning model.
6. Interact with patient data using [Power BI](#) while preserving Azure role-based access control (Azure RBAC).

Components

- [Azure Functions](#): Process events with serverless code
- [Event Grid](#): Get reliable event delivery at massive scale
- [Storage Accounts](#): Durable, highly available, and massively scalable cloud storage
- [Azure SQL Database](#): Managed, intelligent SQL in the cloud
- [Azure Machine Learning](#): Bring AI to everyone with an end-to-end, scalable, trusted platform with experimentation and model management
- [Power BI Embedded](#): Embed fully interactive, stunning data visualizations in your applications
- [Defender for Cloud](#): Unify security management and enable advanced threat protection across hybrid cloud workloads
- [Microsoft Entra ID](#): Synchronize on-premises directories and enable single sign-on
- [Key Vault](#): Safeguard and maintain control of keys and other secrets
- Application Insights: Detect, triage, and diagnose issues in your web apps and services
- [Azure Monitor](#): Full observability into your applications, infrastructure, and network
- [Operation Management Suite](#): A collection of management services that were designed in the cloud from the start
- [Azure RBAC and built-in roles](#): Azure role-based access control (Azure RBAC) has several built-in role definitions that you can assign to users, groups, and service principals.

Scenario details

This solution demonstrates how you can store, manage, and analyze HIPAA-compliant and HITRUST-compliant health data and medical records with a high level of built-in security.

Potential use cases

This solution is ideal for the medical and healthcare industry.

Next steps

- [Azure Functions Documentation](#)
- [Azure Event Grid Documentation](#)
- [Azure Storage Documentation](#)
- [Azure SQL Database Documentation](#)
- [Azure Machine Learning Documentation](#)
- [Power BI Embedded Documentation](#)
- [Microsoft Defender for Cloud Documentation](#)
- [Get started with Microsoft Entra ID](#)
- [What is Azure Key Vault?](#)
- [What is Application Insights?](#)
- [Monitoring Azure applications and resources](#)
- [What is Operations Management Suite \(OMS\)?](#)
- [Built-in roles for Azure role-based access control](#)

Related resources

- [Health data consortium on Azure](#)
- [Virtual health on Microsoft Cloud for Healthcare](#)
- [Confidential computing on a healthcare platform](#)

HPC media rendering

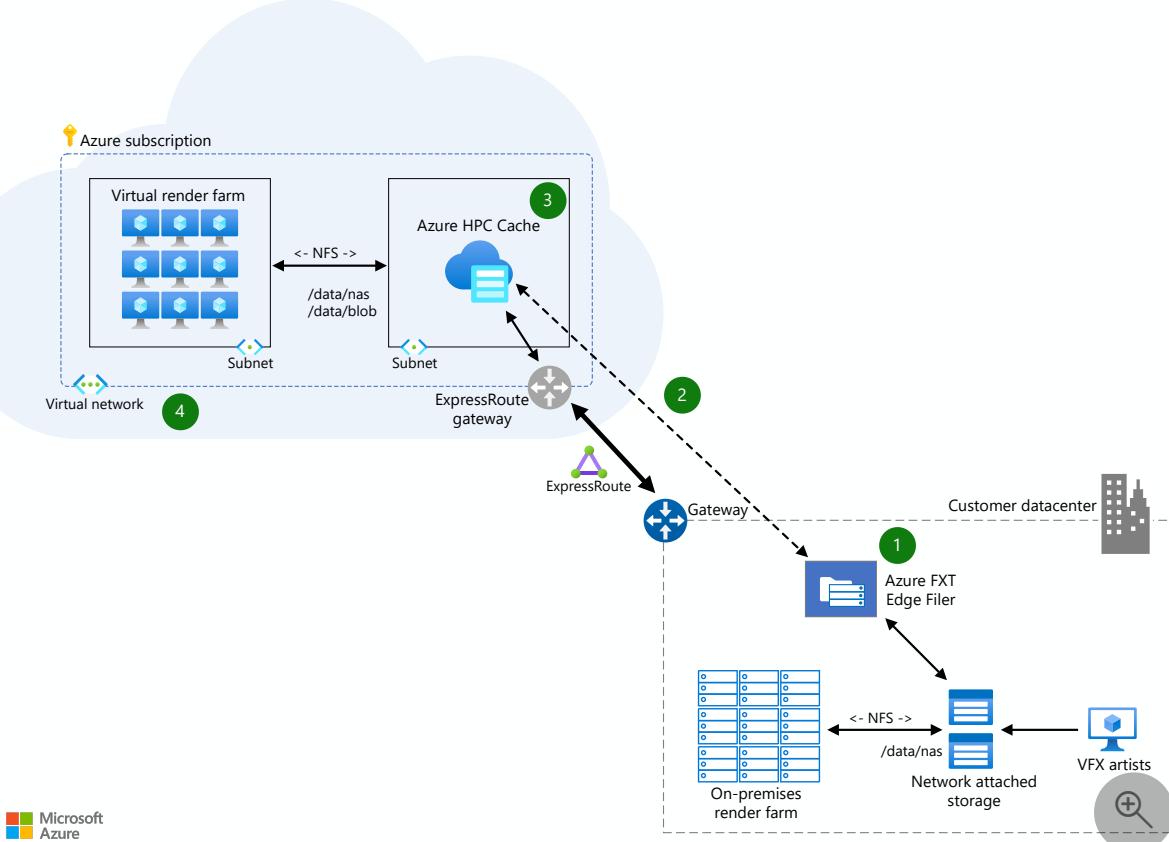
Azure Batch Azure CycleCloud Azure FXT Edge Filer Azure Virtual Machine Scale Sets

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution allows studios to leverage on-premises capacity to its fullest with the Azure FXT Edge Filer for NAS acceleration. When demand grows beyond on-premises capacity, burst render provides access to tens of thousands of cores using Azure Virtual Machine Scale Sets. An Express Route connection and HPC Cache minimize latency while studios securely manage storage in a single place without replication.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Optimize access to NAS files and support remote artists with the Azure FXT Edge Filer connecting artists to low-latency storage.
2. Connecting on-premises storage resources to Azure via Azure Express Route providing a secure, private link to additional render cores.
3. Azure HPC Cache provides low-latency access to tens of thousands of compute cores with burst rendering. Azure SDK support in HPC Cache enables automation for easy infrastructure management and cost efficiencies.
4. A virtual render farm is available in Azure using Virtual Machine Scale Sets that grow as you need it and provides capacity to meet fluctuating demand.

Components

- [N-Series VMs](#) : N-series virtual machines are ideal for compute and graphics-intensive workloads, helping customers to fuel innovation through scenarios like high-end remote visualization, deep learning, and predictive analytics.
- [H-Series VMs](#) : The H-series is a new family specifically designed to handle high performance computing workloads such as financial risk modeling, seismic and reservoir simulation, molecular modeling, and genomic research.
- Effectively manage common workloads with ease while creating and optimizing HPC clusters with Microsoft [Azure CycleCloud](#) .
- [Avere vFXT](#) : Faster, more accessible data storage for high-performance computing at the edge
- [Azure Batch](#) : Cloud-scale job scheduling and compute management

Scenario details

Potential use cases

Graphics designers, artists, and animation designers need high performance systems to make sure they deliver the best quality work and can accommodate change requests without waiting hours for the processing to finish. Areas that studios can see the benefits from high performance computing include:

- Animation and modeling.
- 3D Rendering.
- Compositing and color grading.

Next steps

- [N-Series Virtual Machines Documentation](#)
- [H-Series Virtual Machines Documentation](#)
- [Azure CycleCloud Documentation](#)
- [Avere vFXT Documentation](#)
- [Azure Batch Documentation](#)

Medical data storage solutions

Azure Blob Storage

Azure Data Factory

Azure Data Lake

Azure Machine Learning

Power BI

💡 Solution ideas

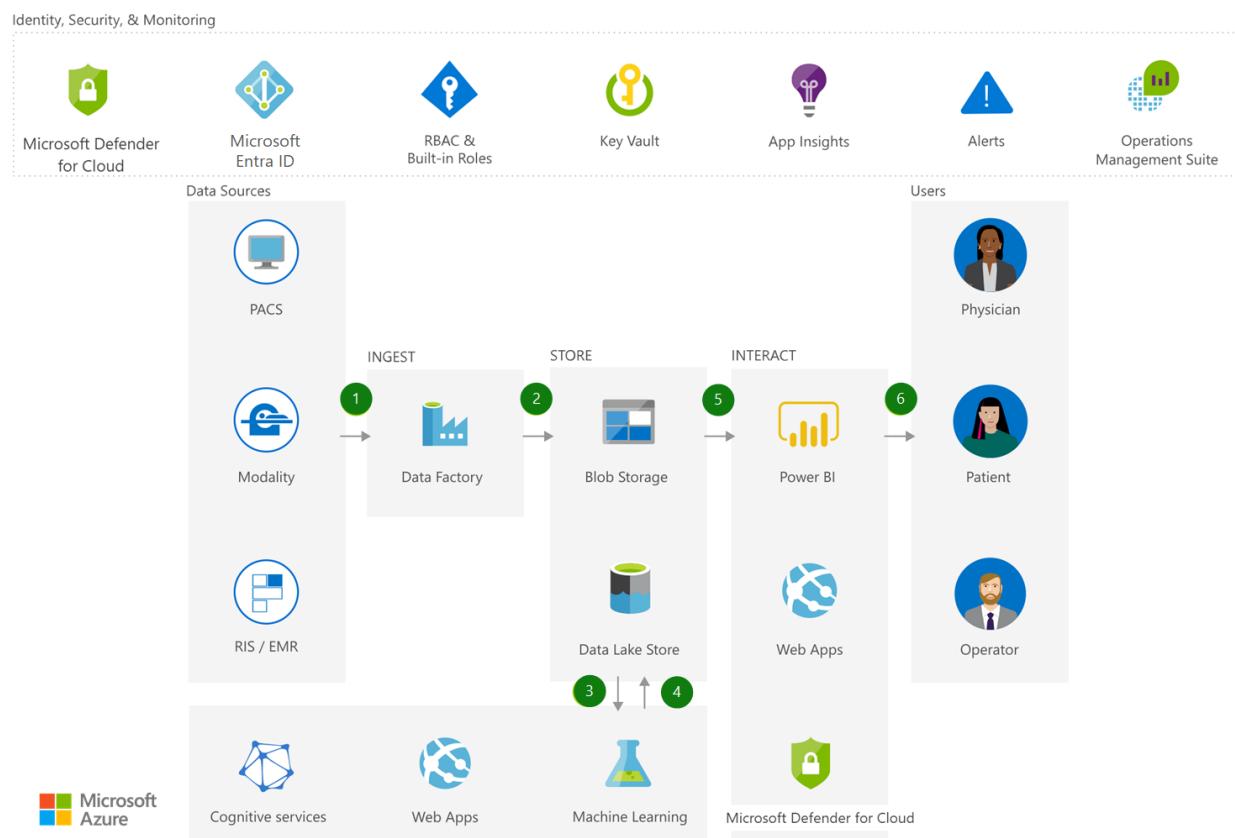
This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Cloud and hybrid solutions from Microsoft help you manage medical data storage efficiently and cost effectively, while infusing intelligence and maintaining compliance.

Potential use cases

This solution is primarily for the healthcare, medical, medical insurance, and finance industries.

Architecture



Download an [SVG](#) of this architecture.

Dataflow

1. Securely ingest medical image data using Azure Data Factory.
2. Securely store medical image data in Azure Data Lake Store and/or Azure Blob Storage.
3. Analyze medical image data using a pre-trained Azure Cognitive Services API or a custom developed Machine Learning model.
4. Store artificial intelligence (AI) and Machine Learning results in Azure Data Lake.
5. Interact AI and Machine Learning results using Power BI, while preserving Azure role-based access control (Azure RBAC).
6. Securely interact with medical image data via a web based vendor neutral archive (VNA) image viewer.

Components

- [Data Factory](#): Hybrid data integration at enterprise scale, made easy
- [Data Lake Storage](#): Hyperscale repository for big data analytics workloads
- [Cognitive Services](#): Add smart API capabilities to enable contextual interactions
- [Web Apps](#): Quickly create and deploy mission critical web apps at scale
- [Defender for Cloud](#): Unify security management and enable advanced threat protection across hybrid cloud workloads
- [Microsoft Entra ID](#): Synchronize on-premises directories and enable single sign-on
- [Key Vault](#): Safeguard and maintain control of keys and other secrets
- Application Insights: Detect, triage, and diagnose issues in your web apps and services
- [Azure Monitor](#): Full observability into your applications, infrastructure, and network
- [Machine Learning](#): Easily build, deploy, and manage predictive analytics solutions
- [Power BI Embedded](#): Embed fully interactive, stunning data visualizations in your applications

Next steps

- [Azure Data Factory V2 Preview Documentation](#)
- [Data Lake Store Documentation](#)
- [Get started with Azure](#)
- [Web Apps overview](#)
- [Microsoft Defender for Cloud Documentation](#)
- [Get started with Microsoft Entra ID](#)

- [What is Azure Key Vault?](#)
- [Application Insights Documentation](#)
- [Azure Monitor Documentation](#)
- [Azure Machine Learning Documentation](#)
- [Power BI Embedded Documentation](#)

Two-region web application with Table Storage failover

Azure Front Door

Azure App Service

Azure Functions

Azure Table Storage

Azure Cache for Redis

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This architecture provides a high-availability solution for a web application that uses massive amounts of data. A secondary region serves as a standby to the primary, improving availability. The primary region sends its data to the secondary by using the built-in replication capabilities of Azure Storage.

Data is stored in Azure Table Storage tables. As with any Azure Storage service, Table Storage data is replicated synchronously three times in the primary region. To make it available for standby use, it's also replicated asynchronously three times in the secondary region. For information about Azure Storage replication, see [Azure Storage redundancy](#).

The architecture includes a cache for the tables to reduce access load and improve application response.

⚠ Note

Your application may require multiple storage accounts under some circumstances. See [Considerations](#) for more information.

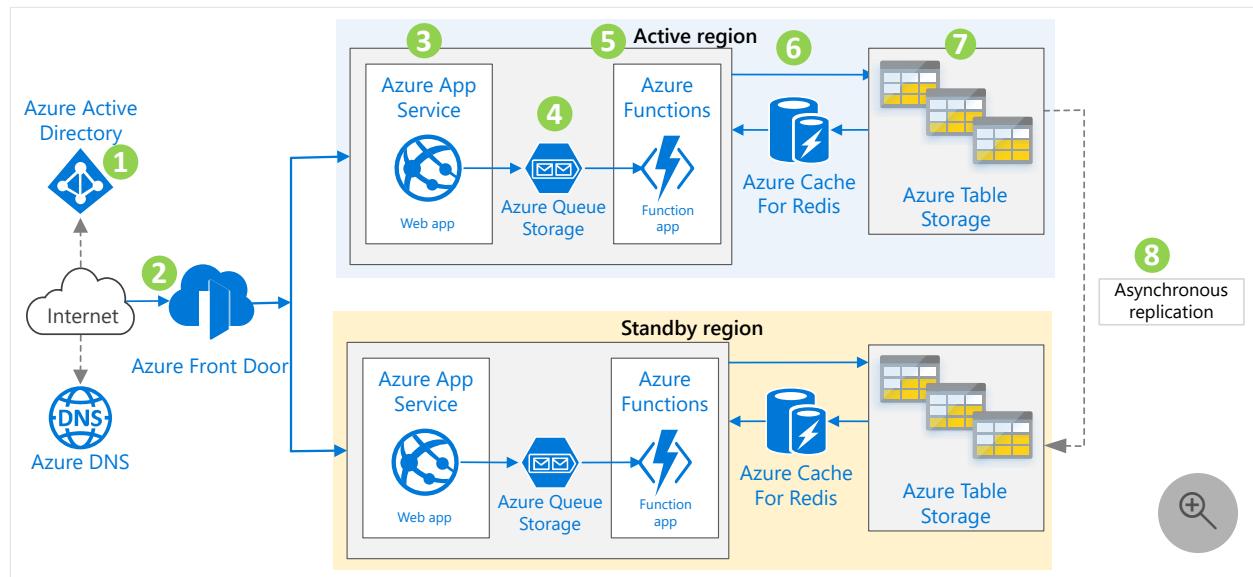
Potential use cases

The architecture may be appropriate for any application that uses massive amounts of data that must always be available. Examples include apps that:

- Track customer spending habits and shopping behavior.
- Forecast weather.

- Offer smart traffic systems or implement smart traffic systems or use smart technology to monitor traffic.
- Analyze manufacturing Internet of Things (IoT) data.
- Display smart meter data or use smart technology to monitor meter data.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. The client authenticates with Microsoft Entra ID and is granted access to web applications hosted on Azure App Service.
2. Azure Front Door, a firewall and layer 7 load balancer, switches user traffic to the standby region in case of a regional outage.
3. Azure App Service hosts websites and RESTful web APIs. Browser clients run AJAX applications that use the APIs.
4. Web APIs delegate function apps to handle background tasks. The tasks are queued in Azure Queue Storage queues.
5. The function apps hosted by Azure Functions perform the background tasks, triggered by the queued messages.
6. Azure Cache for Redis caches table data for the function apps. This offloads database activity and speeds up the function apps and web apps.
7. Azure Table Storage holds the data used by the web applications.
8. Table Storage supports synchronous replication of data across availability zones in the region to mitigate data center outages. It also uses asynchronous replication for replicating data across different Azure regions to remediate regional outages and improve application availability.

Components

- [Microsoft Entra ID](#) is a multi-tenant identity and access management service that can synchronize with an on-premises directory.
- [Azure DNS](#) is a high-availability hosting service for DNS domains that provides apps with fast DNS queries and quick updates to DNS records. Managing Azure DNS is like managing other Azure services, and uses the same credentials, APIs, tools, and billing.
- [Azure Front Door](#) is a secure content delivery network (CDN) and load balancer with instant failover. It operates at the edge close to users, accelerating content delivery while protecting apps, APIs, and websites from cyber threats.
- [Azure App Service](#) is a fully managed service for building, deploying, and scaling web apps. You can build apps using .NET, .NET Core, Node.js, Java, Python, or PHP. Apps can run in containers or on Windows or Linux. In a mainframe migration, the front-end screens or web interface can be coded as HTTP-based REST APIs. They can be segregated and can be stateless to orchestrate a microservices-based system. For more information on web APIs, see [RESTful web API design](#).
- [Azure Functions](#) provides an environment for running small pieces of code, called functions, without having to establish an application infrastructure. You can use it to process bulk data, integrate systems, work with IoT, and build simple APIs and microservices. With microservices, you can create servers that connect to Azure services and are always up to date.
- [Azure Storage](#) is a set of massively scalable and secure cloud services for data, apps, and workloads. It includes [Azure Files](#), [Azure Table Storage](#), and [Azure Queue Storage](#). Azure Files is often an effective tool for migrating mainframe workloads.
- [Azure Queue Storage](#) provides simple, cost-effective, durable message queueing for large workloads.
- [Azure Table Storage](#) is a NoSQL key-value store for rapid development that uses massive semi-structured datasets. The tables are schemaless and adapt readily as needs change. Access is fast and cost-effective for many types of applications, and typically costs less than other types of keyed storage.
- [Azure Cache for Redis](#) is a fully managed in-memory caching service and message broker for sharing data and state among compute resources. It includes both the open-source Redis and a commercial product from Redis Labs as managed services. You can improve performance of high-throughput online transaction processing applications by designing them to scale and to make use of an in-memory data store such as Azure Cache for Redis.

Alternatives

- [Azure Traffic Manager](#) directs incoming DNS requests across the global Azure regions based on your choice of traffic routing methods. It also provides automatic failover and performance routing.
- [Azure Content Delivery Network](#) caches static content in edge servers for quick response, and uses network optimizations to quicken response for dynamic content. Content Delivery Network is especially useful when the user base is global.
- [Azure Kubernetes Service \(AKS\)](#) is a fully managed Kubernetes service for deploying and managing containerized applications. You can use it to implement a microservices architecture whose components scale independently on demand.
- [Azure Container Instances](#) provides a quick and simple way to run tasks without having to manage infrastructure. It's useful during development or for running unscheduled tasks.
- [Azure Service Fabric](#) is a platform for scaling and orchestrating containers and microservices.
- [Azure Service Bus](#) is a reliable cloud messaging service for simple hybrid integration. It can be used instead of Queue Storage in this architecture. For more information, see [Storage queues and Service Bus queues - compared and contrasted](#).

Considerations

- There are performance limits on Table Storage that can be overcome by adding Storage accounts. The following circumstances may require additional accounts:
 - To implement multi-tenancy to support multiple customers
 - To support customers with higher transaction rates
 - To support customers with large datasets
 - To speed up data access by distributing data across multiple storage accounts
 - To segregate data into hot, cold, and archive tiers
 - To make copies of data for backup and reporting purposes

For more information, see [Scalability and performance targets for Table Storage](#).

- Table Storage replication isn't available in some Azure regions.
- The data in a secondary region has eventual consistency, which means that there's a lag between the time an update occurs in a primary region and when it's seen in the secondary region. Because replication from the primary region to the secondary region is asynchronous, data can be lost if the primary region fails and does not recover. There is currently no service level agreement (SLA) on how long it takes to replicate data to the secondary region. For more information, see [Azure Storage redundancy](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Nabil Siddiqui](#) | Cloud Solution Architect - Digital and Application Innovation

Next steps

- [Use geo-redundancy to design highly available applications](#)
- [Guidelines for table design](#)

Related resources

- [Web-Queue-Worker architecture style](#)
- [Data partitioning strategies](#)
- [Build a scalable system for massive data](#)
- [RESTful web API design](#)

Virtual desktop architecture design

Article • 10/10/2023

Migrating end-user desktops to the cloud helps improve employee productivity and enables employees to work from anywhere on a high-security cloud-based virtual desktop infrastructure.

Azure provides these virtual desktop solutions:

- [Azure Virtual Desktop](#) is a desktop and application virtualization service.
- [VMware Horizon Cloud on Microsoft Azure](#) is a VMware service that simplifies the delivery of virtual desktops and applications on Azure by extending Azure Virtual Desktop.
- [Citrix Virtual Apps and Desktops for Azure](#) is a desktop and app virtualization service that you can use to provision Windows desktops and apps on Azure with Citrix and Azure Virtual Desktop.
- [Azure Lab Services](#) provides computer labs in the cloud.
- [Microsoft Dev Box Preview](#) is a service that gives developers access to ready-to-code, project-specific workstations that are preconfigured and centrally managed in the cloud.

Introduction to virtual desktop architecture on Azure

If you're new to virtual desktops on Azure, the best way to learn more is [Microsoft Learn training](#), a free online platform. Here's a learning path to get you started:

- [Deliver remote desktops and apps with Azure Virtual Desktop](#)

Path to production

Cloud Adoption Framework for Azure provides an end-to-end scenario to guide you through your virtual desktop migration or deployment. Start with [Migrate or deploy Azure Virtual Desktop instances to Azure](#), and check out the other articles below that one in the table of contents.

These are two more key Cloud Adoption Framework articles:

- [Azure Virtual Desktop planning](#)
- [Azure Virtual Desktop Azure landing zone review](#)

See [Understanding Azure Virtual Desktop network connectivity](#) for a high-level overview of the network connections used by Azure Virtual Desktop.

Best practices

- [Security best practices for Azure Virtual Desktop](#)
- [Azure security baseline for Azure Virtual Desktop](#)
- [Session host virtual machine sizing guidelines](#)
- [Configure device redirection](#)
- [Set up scaling tool using Azure Automation and Azure Logic Apps for Azure Virtual Desktop](#)

More virtual desktop resources

The following sections, organized by category, provide links to example scenarios and other articles.

Identity

- [Authentication in Azure Virtual Desktop](#)
- [Microsoft Entra join for Azure Virtual Desktop](#)
- [Multiple forests with AD DS and Microsoft Entra ID](#)
- [Multiple forests with AD DS, Microsoft Entra ID, and Microsoft Entra Domain Services](#)

Azure Virtual Desktop for the enterprise

- [Azure Virtual Desktop for the enterprise](#)

FSLogix

FSLogix is designed for roaming profiles in remote computing environments like Azure Virtual Desktop. It stores a complete user profile in a single container. At sign-in, this container is dynamically attached to the computing environment. For more information, see these resources:

- [FSLogix configuration examples](#)
- [FSLogix profile containers and Azure Files](#)
- [Storage options for FSLogix profile containers in Azure Virtual Desktop](#)

Stay current with virtual desktop technologies on Azure

Get the [latest updates on Azure Virtual Desktop technologies](#).

Additional resources

Example solutions

These are some additional articles about Azure Virtual Desktop:

- [Azure Virtual Desktop RDP Shortpath for managed networks](#)
- [Multiregion Business Continuity and Disaster Recovery \(BCDR\) for Azure Virtual Desktop](#)
- [Deploy Esri ArcGIS Pro in Azure Virtual Desktop](#)

AWS professionals

- [AWS to Azure services comparison - End-user computing](#)

Windows 365 Azure network connection

Azure ExpressRoute

Azure Virtual Desktop

Azure Virtual Machines

Azure Virtual Network

Windows 365 is a cloud-based service that you can use to deliver highly optimized and personalized Windows computing instances called Cloud PCs that are purpose-built for each user's requirements. Cloud PCs use a combination of the following services:

- Intune to customize, secure, and manage Cloud PCs
- Entra ID for identity and access control
- Azure Virtual Desktop for remote connectivity

A Cloud PC is a highly available, optimized, and personalized computing instance that provides you with a rich Windows desktop experience. It's hosted in the Windows 365 service and is accessible from anywhere, on any device.

The Windows 365 shared responsibility model

Windows 365 is a software as a service (SaaS) solution. Microsoft manages some components in Windows 365 services, and you manage other components. The amount of responsibility you have depends on the architecture pattern you choose for deployment. The responsibilities to manage Windows 365 are split into three parts:

- **Deployment:** Planning and deploying the components of the service.
- **Lifecycle:** Management of the component throughout its lifecycle, such as patching and securing.
- **Configuration:** Configuring the component to apply settings as needed for a scenario.

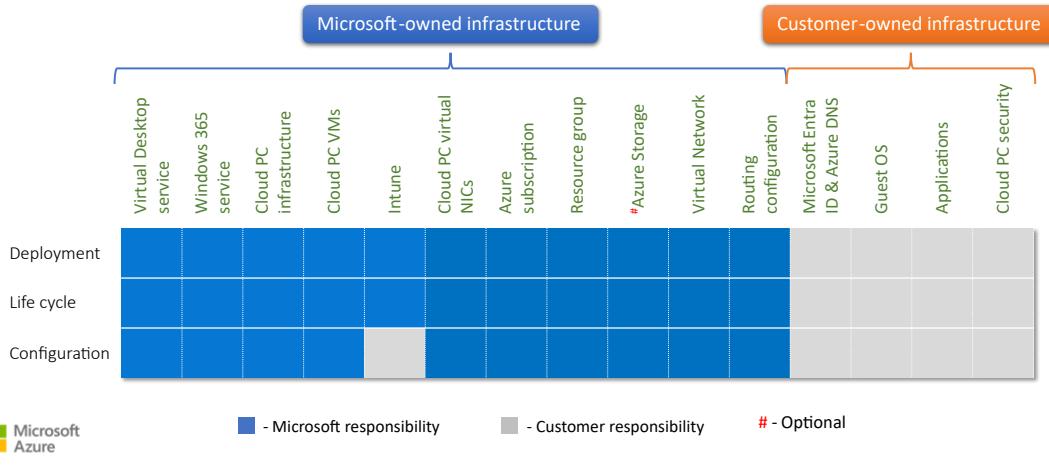
The following diagram shows the responsibility matrix of a Windows 365 deployment by using the recommended Microsoft-hosted network, Microsoft Entra join, and gallery images with Windows Autopatch. With this configuration, you don't have to manage many components and lifecycle stages. This configuration translates to the benefits listed in [Recommended architecture patterns](#).

Note

The following diagram represents the responsibilities from the infrastructure perspective, such as setting up the hardware and network, and maintaining them. It

doesn't include the Windows 365 or Intune tenant subscription setup.

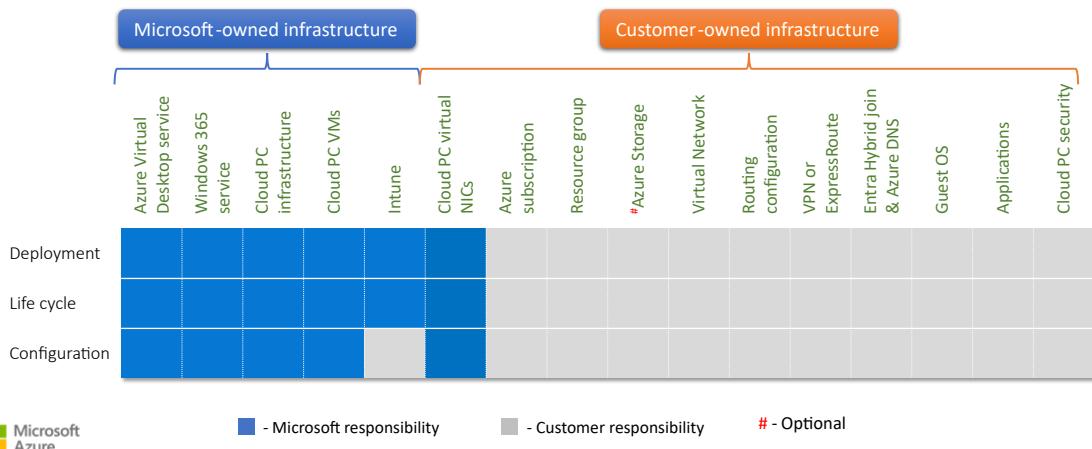
Windows 365 Microsoft-hosted network architecture pattern: Responsibility matrix



Download a [PowerPoint file](#) of this architecture.

The following diagram shows a typical Windows 365 deployment that uses an Azure network connection and shows the components that Microsoft manages and the components that you manage across the lifecycle stages of a Cloud PC.

Windows 365 Azure network connection architecture pattern: Responsibility matrix



Download a [PowerPoint file](#) of this architecture.

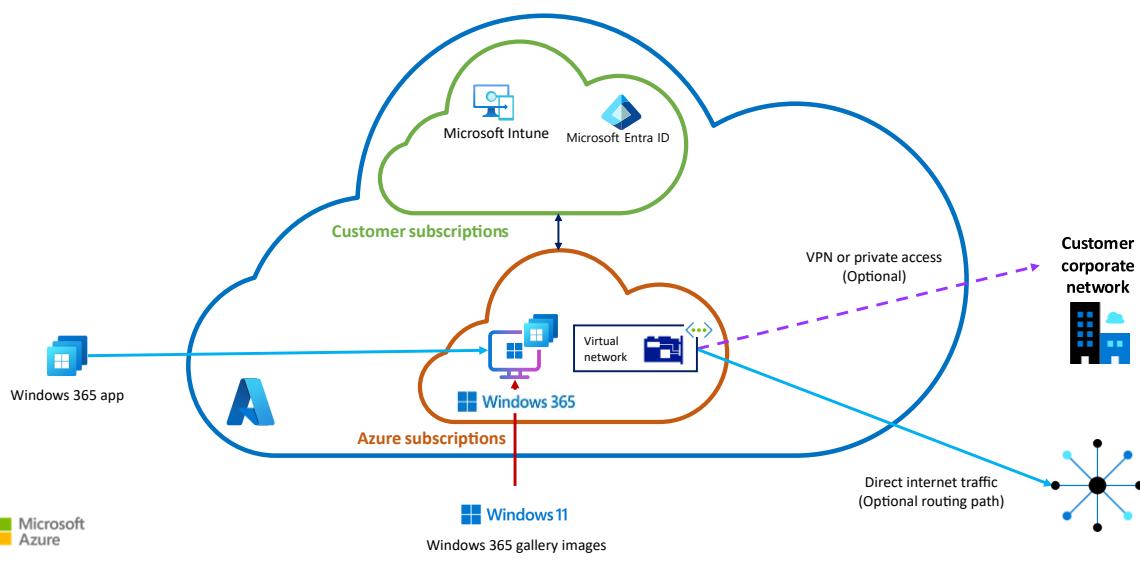
Recommended architecture pattern

Microsoft recommends deploying Windows 365 with the following components to get a SaaS experience, enabling you to have the maximum benefits of the service:

- Microsoft Entra join
- A Microsoft-hosted network
- Gallery images
- Intune-based mobile device management (MDM) service with app and OS configuration
- A Windows 365 app for Cloud PC access

Microsoft-hosted network - Cloud PC's Azure virtual network that's fully managed by Microsoft

Microsoft Entra join



Download a [PowerPoint file](#) of this architecture.

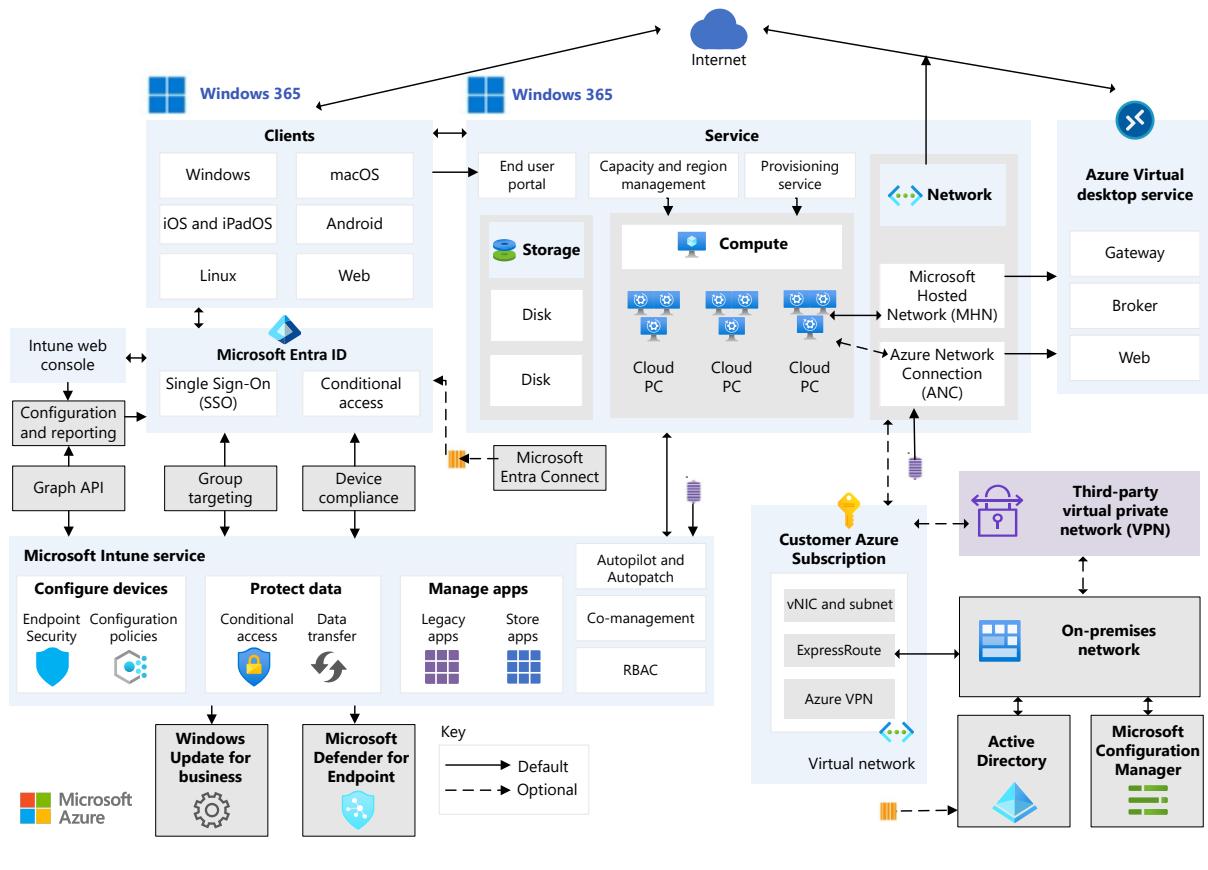
The preceding architecture pattern allows you to get the most out of the Windows 365 service and provides the following benefits:

- Simplified and faster deployment
- Minimal to zero dependencies
- Full Zero Trust framework support
- Simplified troubleshooting flows
- Self-service user troubleshooting
- Low overhead and management
- Highest maturity model of software and application delivery

The Windows 365 service architecture

The following diagram is a representation of all the components that are part of the Windows 365 service. This architecture uses Intune and Microsoft Entra ID, which are

core requirements of Windows 365. There are also optional components such as Azure Virtual Network.



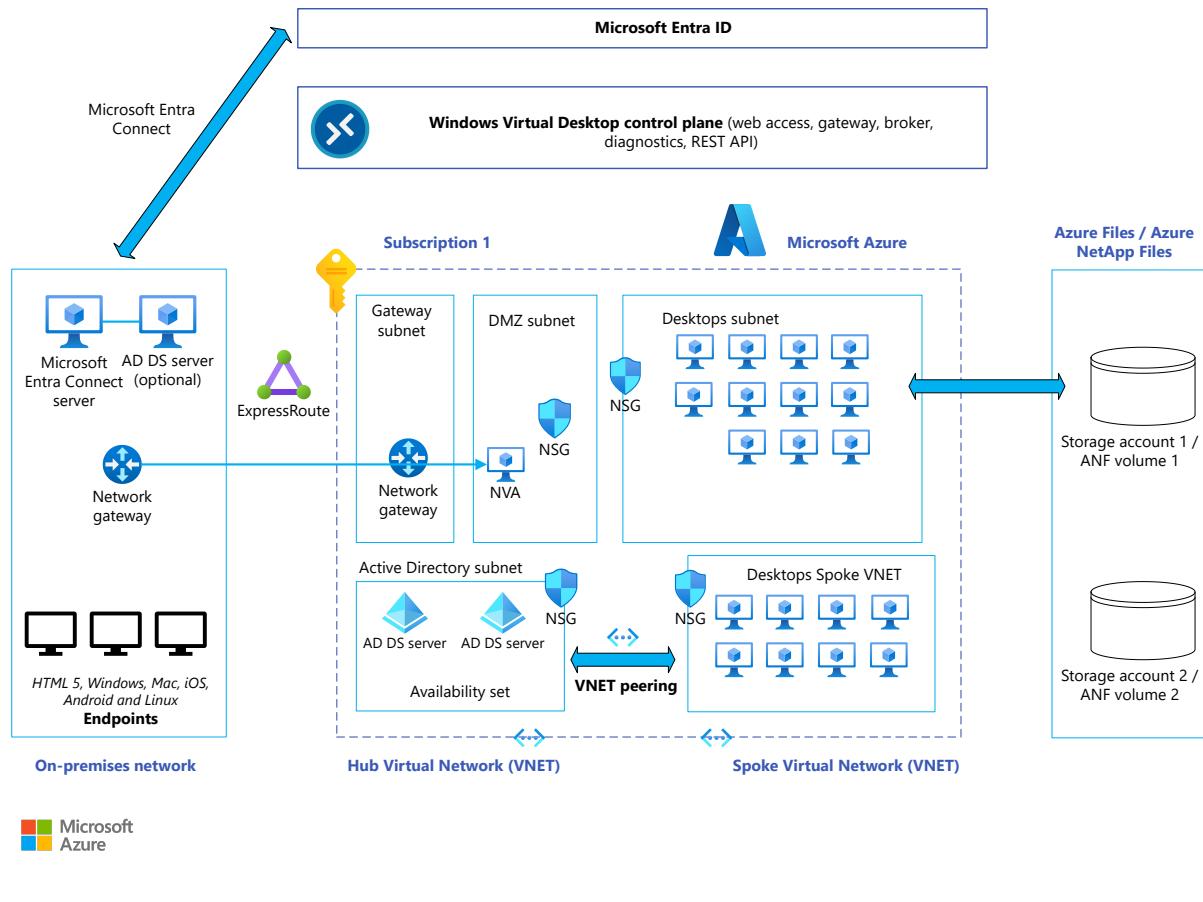
Download a [Visio file](#) of this architecture.

The previous diagram shows Azure network connection and Microsoft-hosted network options. They're mutually exclusive architecture options. The following sections elaborate on the Azure network connection options.

Virtual Desktop

Virtual Desktop is an Azure-based virtual desktop infrastructure (VDI) solution. Microsoft manages Virtual Desktop. It provides a platform-as-a-service (PaaS) style solution. Windows 365 uses the network management components required for you to connect to their Cloud PCs. Components include the Virtual Desktop gateway service, a connection broker service, and a web client service. These services allow for seamless connection to Windows 365 Cloud PCs.

For more information, see [Azure Virtual Desktop for the enterprise](#).



Download a [Visio file](#) of this architecture.

ⓘ Note

Windows 365 utilizes the components entitled "Windows Virtual Desktop Control Plane" in the previous diagram for facilitating user and Cloud PC connections and as such inherits most of the connection related capabilities of Azure Virtual Desktop. Familiarizing with how Virtual Desktop networking operates then becomes essential to designing the Azure network connection architecture detailed in this document.

Microsoft Intune

Intune is a cloud-based endpoint management solution that allows you to view and consume reports and manage:

- App delivery
- Windows updates
- Device management configurations
- Security policies

Intune simplifies app and device management across many devices, including mobile devices, desktop computers, and virtual endpoints.

You can protect access and data on organization-owned and personal devices. Intune also has compliance and reporting features that support the Zero Trust security model. For more information, see [Create a device configuration profile](#).

Architecture patterns

An architecture pattern describes components and illustrates the configurations with which a service or product is deployed. For more information, see [Hosted on behalf of architecture](#).

See the following Azure network connection patterns:

Azure network connection with Microsoft Entra join – In this pattern, Microsoft Entra joined Cloud PCs use Azure network connection to connect to resources in on-premises environments, such as line-of-business (LOB) applications, file shares, and other applications that don't need Kerberos or Windows New Technology LAN Manager (NTLM) authentication.

Azure network connection with Microsoft Entra hybrid join – In this pattern, Microsoft Entra hybrid joined Cloud PCs use Azure network connection to domain join with an on-premises Microsoft Entra ID domain controller. The Cloud PC authenticates with the on-premises domain controller when users access the Cloud PC, on-premises apps, or cloud apps that need Kerberos or NTLM authentication.

Azure network connection architecture patterns

For some patterns, the Windows 365 service connects to on-premises environments via Virtual Network by using Azure ExpressRoute or a site-to-site VPN. This connectivity method is represented by Azure network connection, which is an Intune object. This connection allows the Cloud PCs to connect to on-premises resources such as Active Directory or LOB apps.

This network connection, that's represented by Azure network connection, is used by the Windows 365 service during Cloud PC provisioning for on-premises Microsoft Entra domain joining, [health checks](#) for Cloud PC provisioning readiness.

The following tables list dependencies for Azure network connection. Windows 365 runs automatic health checks on these dependencies.

 [Expand table](#)

Dependency	Microsoft Entra Connect - Checks if Microsoft Entra Connect is set up and finishes successfully.
Architecture patterns	Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Set up the Microsoft Entra Connect sync interval with the default or lowest value. Longer sync intervals increase the possibility of the Cloud PC provisioning failing in production due to a timeout. For more information, see Microsoft Entra hybrid join failing. - Set up Active Directory Domain Controller replication from a server in the same datacenter as the Windows 365 Azure network connection to provide faster replication. - Set up Microsoft Entra ID domain controller replication with a default value.

 [Expand table](#)

Dependency	Azure tenant readiness - Checks if Azure subscription is enabled, with no blocking restrictions, and is ready for use.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Use an account with the right privileges to manage the Azure, Intune, and Windows 365 subscriptions. For more information, see Role-based access control(RBAC). - Disable or modify any Azure policies that prevent the creation of Cloud PCs. For more information, see Restrict allowed VM SKUs. - Make sure the subscription has sufficient resource quotas for networking and general limits based on the maximum number of Cloud PCs to be created. Examples include the network gateway size, IP address space, size of the virtual network, and bandwidth required. For more information, see Networking limits and General limits.

 [Expand table](#)

Dependency	Azure virtual network readiness – Checks if the virtual network is in a supported Windows 365 region.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Create the virtual network in a Windows 365 Supported Azure regions for Cloud PC provisioning. - Create at least one subnet, in addition to the default subnet, to deploy the Cloud PC virtual network adaptors. - Where possible, create shared network services, such as Azure Firewall, VPN gateways, or ExpressRoute gateways, in a separate virtual network to allow for routing controls and expansion of deployment. <p>In virtual networks, apply network security groups (NSG) with appropriate exclusions to allow the required URLs for the Windows 365 service. For more information, see Networking requirements and Network security groups.</p>

Expand table

Dependency	Azure subnet IP address usage – Checks if there are sufficient IP addresses available.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Create the virtual network with sufficient IP addresses to handle the Cloud PC creation and temporary IP address reservation during reprovision. It's recommended that you use an IP address space that's 1.5 to 2 times the maximum Cloud PCs that you deploy for the cloud. For more information, see General network requirements. - Treat the Azure virtual network as a logical extension of your on-premises network, and assign unique IP address space across all your networks to avoid routing conflicts.

Expand table

Dependency	Endpoint connectivity – Checks if the external URLs needed for the Cloud PC provisioning are reachable from the virtual network.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Allow all the URLs needed for the Cloud PC provisioning via the Azure virtual network. For more information, see Allow network connectivity. - Use Azure Firewall to take advantage of Windows 365, Azure Virtual Desktop, and Intune FQDN tags to create application rules and allow URLs needed for the Windows 365 Cloud PC provisioning. For more information, see Use Azure Firewall to manage and secure Windows 365 environments. - Bypass or exclude Remote Desktop Protocol (RDP) traffic from any network inspection, proxying, or manipulation device to avoid latency and routing issues. For more information, see Traffic interception technologies. - From the end user device and network side, allow the Windows 365 service URLs and ports for proxy and network inspections. - Allow Azure internal IP addresses 168.63.129.16 and 169.254.169.254, as these IP addresses are used for communication with Azure platform services such as metadata or heartbeat. For more information, see What is IP address 168.63.129.16?, Azure Instance Metadata Service, and Virtual Network FAQ.

Expand table

Dependency	Intune enrollment – Checks if Intune allows Windows enrollment.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Ensure that Intune device type enrollment restrictions are set to allow the Windows mobile device management (MDM) platform for corporate enrollment. - For Microsoft Entra hybrid join, set up devices automatically by configuring the service connection point (SCP) for each domain in Microsoft Entra Connect or by using the targeted deployment model. For more information, see Configure Microsoft hybrid join and Microsoft Entra hybrid join targeted deployment.

[+] [Expand table](#)

Dependency	First-party app permissions – Checks the Windows 365 app for permissions on the customer Azure subscription, resource group, and virtual network levels.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none">- Ensure the account used for setting up the Azure network connection has read permissions on the Azure subscription in which the Azure virtual network is created.- Ensure in the Azure subscription that there are no policies in place that block permissions for the Windows 365 first-party app. The app must have permissions at the subscription, resource group, and virtual network level. For more information, see Azure requirements.

[+] [Expand table](#)

Dependency	Localization language pack – Checks if the language pack download locations are reachable.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none">- Ensure the URLs needed for the appropriate version of Windows images are allowed via firewall rules used in the Azure virtual network. For more information, see Provide a localized Windows experience.

[+] [Expand table](#)

Dependency	RDP Shortpath – Checks if the User Datagram Protocol (UDP) configurations are in place for you to connect.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none">- Enable RDP Shortpath for Cloud PC access to take advantage of UDP's resilience. For more information, see Use RDP Shortpath for

Dependency	RDP Shortpath – Checks if the User Datagram Protocol (UDP) configurations are in place for you to connect.
	public networks with Windows 365 and Use RDP Shortpath for private networks with Windows 365 .

[+] [Expand table](#)

Dependency	Intune license – Checks if the tenant has appropriate Intune licenses to use Windows.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Ensure Intune licenses are assigned to you in accordance with the licensing requirements.

[+] [Expand table](#)

Dependency	Single sign-on (SSO) check – Checks if the Kerberos server object is created in Active Directory and synced to Microsoft Entra ID.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Ensure that the SSO option is selected in the provisioning policy. This option enables you to connect to the policy's Cloud PC by using sign in credentials from an Intune-managed physical device that's domain joined or Microsoft Entra joined. For more information, see Continue creating provisioning policies.

[+] [Expand table](#)

Dependency	DNS name resolution – Checks if the DNS in the Azure network connection can resolve on-premises Active Directory domain.
Architecture	Azure network connection for Microsoft Entra join, Azure network

Dependency	DNS name resolution – Checks if the DNS in the Azure network connection can resolve on-premises Active Directory domain.
patterns	connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Ensure the Azure virtual network is configured with the name resolution of an on-premises Microsoft Entra domain by using a custom DNS, a private DNS, or a private resolver. For more information, see What is Azure DNS? - Ensure the DNS servers configured in the virtual network are in the same geography and have the ability to register newly provisioned Cloud PCs without delays. Avoid DNS referral or redirections to prevent propagation delays, which can result in provisioning delays or failures.

[+] [Expand table](#)

Dependency	Microsoft Entra domain join – Checks that the credentials provided for Microsoft Entra domain join are valid and Cloud PCs can be domain joined.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Ensure the account provided for Microsoft Entra domain join has permissions on the Microsoft Entra organizational unit specified in the Azure network connection configuration. - Ensure the account provided isn't a standard user account with a domain join limitation. For more information, see Default limit to number of workstations a user can join to the domain. - Ensure the account specified is synced to Microsoft Entra ID. - Ensure the OU specified in the Azure network connection doesn't have any object limits. For more information, see Increase the computer account limit in the organizational unit.

For more information, see [Azure network connection health checks in Windows 365](#).

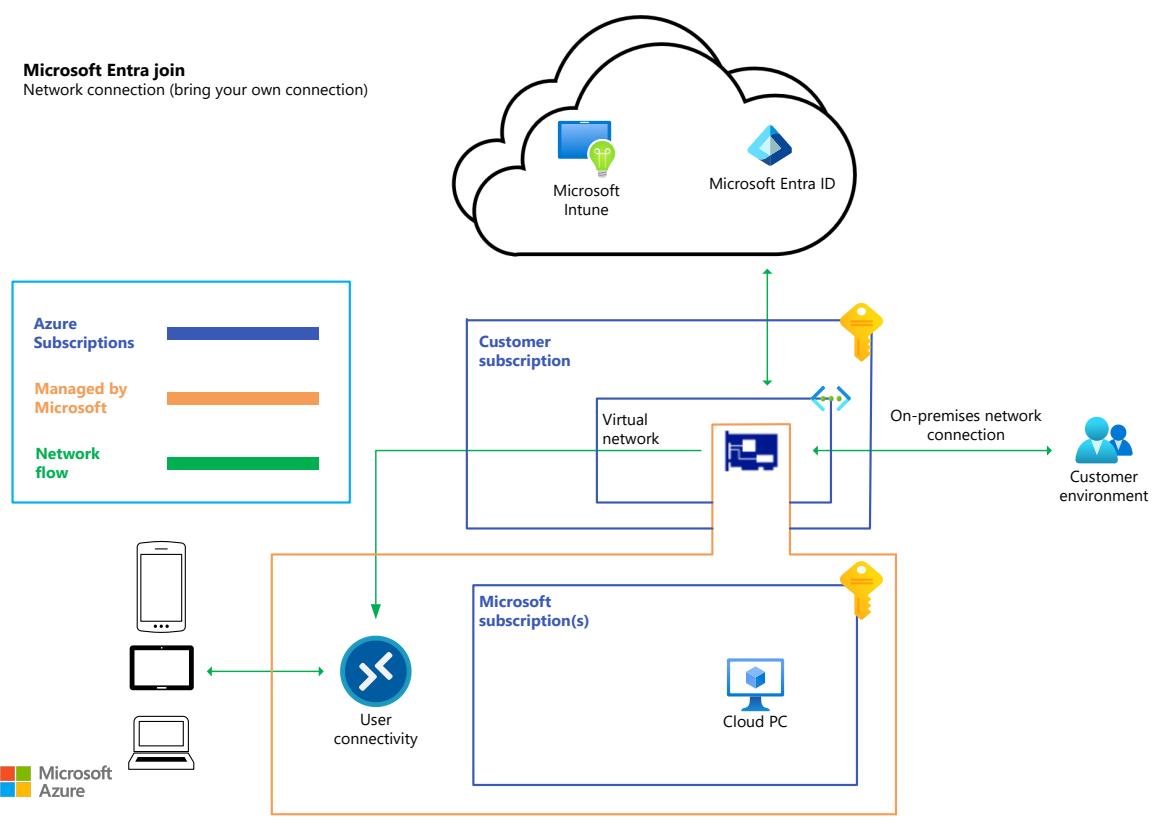
Azure network connection building blocks recommendations

This section provides the breakdown of building blocks of the Windows 365 Azure network connection architecture pattern.

Azure subscription

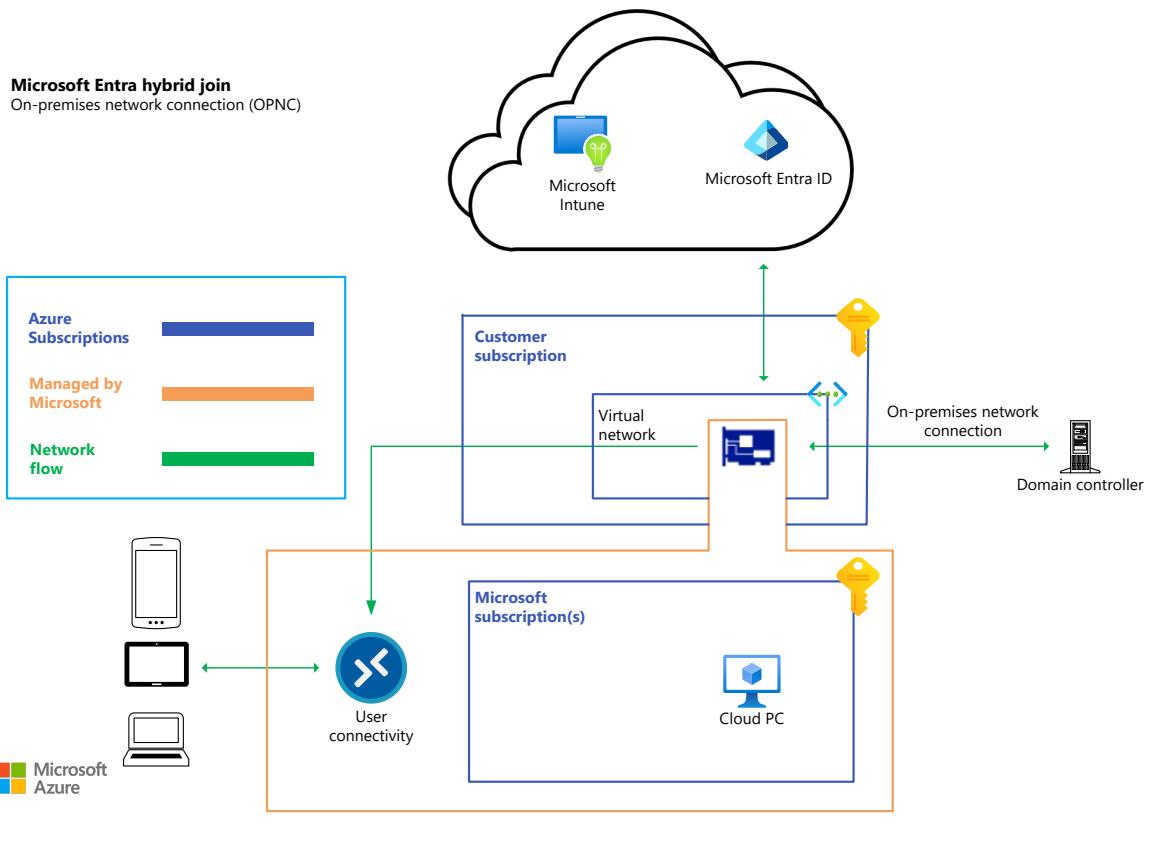
Windows 365 usage in an Azure network connection architecture pattern involves two types of Azure subscriptions, a Microsoft subscription and a customer subscription

Windows 365 uses the *Hosted on behalf of* model to deliver services to Windows 365 customers. In this model, the Cloud PC is provisioned and run in Azure subscriptions owned by Microsoft, while the network adapter of the Cloud PC is provisioned in a customer's Azure subscription. The following diagrams show two Azure network connection architecture patterns. Customers use their own Azure subscription and virtual network.



Download a [Visio file](#) of this architecture.

The previous architecture pattern uses the Microsoft Entra join identity to manage the Cloud PC.



Download a [Visio file](#) of this architecture.

The previous architecture pattern uses Microsoft Entra hybrid join identity to manage the Cloud PC and requires a *line of sight* network communication with Active Directory Domain Services (AD DS) domain controllers in on-premises environments.

[\[\] Expand table](#)

Component	Azure subscription – Azure subscription that hosts the virtual network used for providing connectivity for a Cloud PC to an on-premises environment and the internet.
Architecture patterns	Azure network connection for Microsoft Entra join, Azure network connection for Microsoft Entra hybrid join
Recommendations	<ul style="list-style-type: none"> - Create or use a subscription that has a virtual network and ExpressRoute or VPN gateways to provide a connection back to an on-premises environment. - Create a dedicated resource group for a Cloud PC to provide permission and resources management. - Exclude Cloud PC resource groups and virtual network from Azure policies that prevent automatic creation and deletion of virtual network interface card (vNIC) objects, and IP address assignment or release. For more information, see Lock your resources to protect your infrastructure and Azure requirements.

Component	Azure subscription – Azure subscription that hosts the virtual network used for providing connectivity for a Cloud PC to an on-premises environment and the internet.
	<ul style="list-style-type: none"> - Create dedicated virtual networks for better IP address management and routing controls.

Virtual Network and hybrid connection

Windows 365 Azure network connection-based architecture patterns require one or more Azure virtual networks. The virtual networks provide connectivity to on-premises environments and over the internet for provisioning a Cloud PC. The virtual network adapter of the Cloud PC is provisioned in the Azure virtual network of the customer-owned subscription as described in the [Azure subscription](#) section.

Azure networking can be deployed with varying design sophistication, based on the existing on-premises networking or Azure networking. To get started with a basic hybrid network design, see [Implement a secure hybrid network](#).

Consider the following factors when you design an Azure virtual network architecture:

- *IP address space*: The size of IP address space depends on the number of Cloud PCs to support. Plan for at least 1.5 times the maximum number of Cloud PCs that are deployed. The additional IP addresses account for IP addresses used during provisioning and deprovisioning of Cloud PCs.
- *Name resolution*: The DNS process used by the Cloud PC to resolve the on-premises domain name in a Microsoft Entra hybrid join deployment or to resolve internet resources or Azure resources in a Microsoft Entra join deployment model.
 - To use your existing on-premises DNS infrastructure, configure the IP addresses of one or more DNS servers for name resolution. For more information, see [DNS requirements](#).
 - Ensure the DNS server IP used in the Azure virtual network belong to the same geography as the Cloud PC and that it doesn't redirect DNS registration requests to another region. Otherwise, it results in delayed or failed deployments and Azure network connection health checks.
 - For Azure DNS-based name resolution, use the public or private Azure DNS or the private resolver option. For more information, see [Azure DNS documentation](#).
- *Network topology*: Azure networking supports topologies to accommodate different use cases.

- *Hub-spoke topology with virtual network peering*: This topology is the simplest way to provide an isolation of services with their own spoke and hub virtual networks. Shared services include Azure Firewall and network gateways. Choose this topology if you have a simple, single-site design to deploy a Cloud PC in one or more spoke virtual networks. For more information, see [Hub-and-spoke network topology](#).
- *Hub-spoke topology with Azure Virtual WAN*: Virtual WAN is an Azure networking service that brings together networking, security, and management capabilities that enable complex network requirements. Use this topology for multi-site, multi-region deployments with specific firewalling and routing requirements. For more information, see [Hub-spoke network topology with Virtual WAN](#).
- *Network gateway*: Azure network gateways provide connectivity from a virtual network to an on-premises network. There are VPN and ExpressRoute network gateways. Ensure that the maximum bandwidth requirements of a Cloud PC are considered before deciding on the ExpressRoute or VPN method of connectivity. Both VPN and ExpressRoute gateways are offered in tiers, or SKUs, that differ in the amount of bandwidth provided and other metrics. For more information, see [Extend an on-premises network using ExpressRoute](#) and [Connect an on-premises network to Azure using ExpressRoute](#).

Routing configurations

Windows 365 Azure network connection service uses automated health checks to determine the health and readiness of the customer's environment to provision Microsoft Entra join or Microsoft Entra hybrid join Cloud PCs in an Azure network connection-based architecture. Without proper routing configurations in your Azure virtual network and associated networking services, there's a high likelihood of failures or delays in your Cloud PC deployment. Consider the following recommendations to optimize routing for the Windows 365 network architecture:

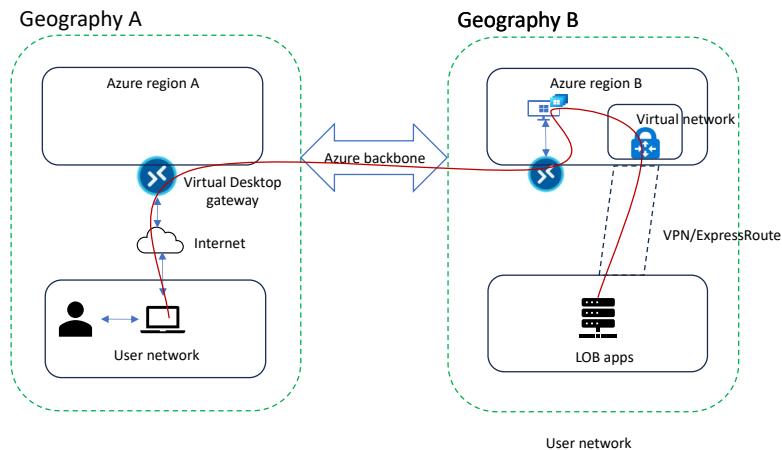
- *Allowlist required URLs*: Each Cloud PC deployed in Microsoft Entra hybrid join and Microsoft Entra join Azure network connection model requires several URLs to be allowed through OS anti-virus, network firewalls, and load balancers. Ensure all the URLs are allowed. For more information, see [Allow network connectivity](#).
- *Use Azure FQDN tags*: When you use the Azure Firewall service, use Azure FQDN tags to allow required URLs for Azure Virtual Desktop, Windows 365, and Intune. For more information, see [Use Azure Firewall to manage and secure Windows 365 environments](#).

- *Enable pass-through:* Windows 365 uses the RDP protocol, which is sensitive to latency introduced by traffic inspection devices such as a firewall or an SSL decryption appliance. Such latency can result in a poor experience, so disable traffic inspection of these URLs and instead enable pass-through. For more information, see [Traffic interception technologies](#).
- *Bypass proxy:* Cloud and traditional proxy services, while suitable for internet access, introduce latency in RDP connections. This latency happens when the connection from the end user's physical device or from the Cloud PC is forced through a proxy and results in frequent disconnections, lags, and sluggish response times. Set `*.wvd.microsoft.com` and [Windows 365 gateway IP ranges](#) to bypass proxy services on the user's physical device, the network the physical device is connected to, and in the Cloud PC.

For more information, see [Optimizing RDP connectivity for Windows 365](#).

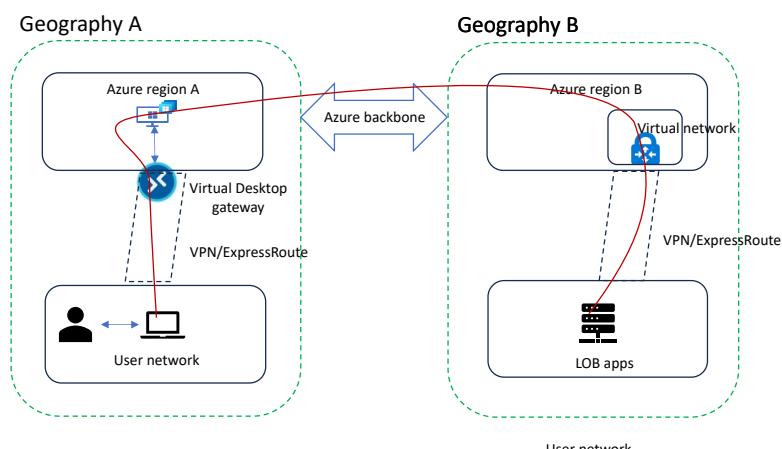
- *Shortest path routing:* Ensure RDP traffic from a Cloud PC reaches Virtual Desktop service endpoints via the shortest path. The ideal path is from a virtual network, directly to the Virtual Desktop gateway IP via the internet. Also ensure RDP traffic from the end user's physical device reaches the Virtual Desktop gateway IP directly. This configuration ensures optimal routing and doesn't degrade the user experience. Avoid routing RDP traffic to the internet via cloud proxy services or on-premises networks.
- *RDP Shortpath:* Enable RDP Shortpath-based access for end user networks, Azure networks, and Cloud PCs. RDP Shortpath uses UDP to transmit RDP traffic. Unlike TCP, it's resilient to high latency network connections. UDP also takes maximum advantage of the available network bandwidth to efficiently transfer RDP packets, which leads to an improved user experience. For more information, see [Use RDP Shortpath for public networks with Windows 365](#).
- *Cloud PC placement:* For an optimal user experience and routing performance, determine where customers are in relation to the work apps or network they access. Also consider the time customers spend accessing the LOB apps compared to the overall time that they access other apps. See the following two possible deployment options:
 - The following deployment model might be optimal if customers spend most of their work time accessing the LOB apps rather than work on locally installed apps, like apps in Microsoft 365. This model optimizes latency for LOB apps vs. Cloud PC access latency by placing the Cloud PC in the same region as the LOB app (Geography B). This optimization occurs even though the gateway is

geographically closer to the end user (Geography A). The following diagram shows the possible traffic flow from the end user to the LOB apps.



[Download a PowerPoint file](#) of this architecture.

- If customers occasionally access the LOB apps in Geography B, then deploying a Cloud PC closer to the customers might be optimal because it optimizes the Cloud PC access latency over LOB apps access latency. The following diagram shows how the traffic might flow in such a scenario.



[Download a PowerPoint file](#) of this architecture.

AD DS recommendations

In a Microsoft Entra hybrid join architecture, an on-premises AD DS infrastructure acts as the identity source of authority. Having a properly configured and healthy AD DS infrastructure is a crucial step to make the Windows 365 deployment successful.

On-premises AD DS supports many configurations and varying levels of complexity, so the recommendations provided only cover the baseline best practices.

- For Microsoft Entra hybrid join scenario, you can deploy AD DS in Azure VMs as described in the architecture reference in [Deploy AD DS in a virtual network](#). You can also use a hybrid network connection to provide a direct line of sight to your on-premises Microsoft Entra domain controller. For more information, see [Implement a secure hybrid network](#).
- For Microsoft Entra join deployment, follow the reference architecture in [Integrate on-premises Microsoft Entra domains with Microsoft Entra ID](#).
- Windows 365 uses a watchdog service as part of automated testing that creates a test VM account. That account shows as disabled in the organizational unit specified in Azure network connection configuration. Don't delete this account.
- Any Cloud PC that's decommissioned in the Microsoft Entra hybrid join model leaves behind a disabled computer account, which needs to be cleaned manually in AD DS.
- Microsoft Entra Domain Services isn't supported as an identity source because it doesn't support Microsoft Entra hybrid join.

DNS recommendations

In an Azure network connection deployment architecture, DNS servers or another DNS service used by an Azure virtual network is a crucial dependency. It's important to have a healthy infrastructure in place.

- For a Microsoft Entra hybrid join configuration, DNS should be able to resolve the domain to which the Cloud PC needs to be joined. There are multiple configuration options available, the simplest of them being specifying your DNS server IP in the Azure virtual network configuration. For more information, see [Name resolution that uses your own DNS server](#).
- Depending on the complexity of the infrastructure, such as a multi-region, multi-domain setup in Azure and on-premises environments, you should use a service like Azure DNS private zones or [Azure DNS Private Resolver](#).

Cloud PC connection recommendations

Deployed Cloud PCs should be configured to allow uninterrupted connection flow to and from the Virtual Desktop gateway service. Consider the following recommendations when you deploy apps as part of a Windows operating system configuration:

- Ensure that the VPS client doesn't launch when the user signs in because it can disconnect the session when the VPN tunnel establishes. The user would have to sign in a second time.
- Configure the VPN, proxy, firewall, and antivirus and antimalware apps to allow or bypass traffic bound for IP addresses 168.63.129.16 and 169.254.169.254. These IP addresses are used for communication with Azure platform services such as metadata and heartbeat. For more information, see [What is IP address 168.63.129.16?](#), [Azure Instance Metadata Service for virtual machines](#), and [Virtual Network FAQ](#).
- Don't manually modify the IP addresses of Cloud PCs because it might result in permanent disconnection. IP addresses are assigned with an indefinite lease and managed throughout the lifecycle of the Cloud PC by Azure networking services. For more information, see [Allocation methods](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ravishankar Nandagopalan](#) | Senior Product Manager

Other contributors

- [Paul Collinge](#) | Principal Product Manager
- [Claus Emerich](#) | Principal Product Manager
- [David Falkus](#) | Principal Product Manager
- [Bob Roudebush](#) | Technical Leader and Cloud/Developer Technologist
- [Matt Shadbolt](#) | Principal Product Manager, Windows Cloud Experiences

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

[Plan your Cloud PC deployment](#)

[Windows 365 architecture](#)

[Windows 365 identity and authentication](#)

[Cloud PC lifecycle in Windows 365](#)

Related resources

[Active Directory Domain Services overview](#)

[Data encryption in Windows 365](#)

[Understanding virtual desktop network connectivity](#)

[Web applications architecture design](#)

Azure Virtual Desktop terminology

Article • 11/01/2023

ⓘ Important

This content applies to Azure Virtual Desktop with Azure Resource Manager Azure Virtual Desktop objects. If you're using Azure Virtual Desktop (classic) without Azure Resource Manager objects, see [this article](#).

Azure Virtual Desktop is a service that gives users easy and secure access to their virtualized desktops and applications. This topic will tell you a bit more about the terminology and general structure of Azure Virtual Desktop.

Host pools

A host pool is a collection of Azure virtual machines that register to Azure Virtual Desktop as session hosts when you run the Azure Virtual Desktop agent. All session host virtual machines in a host pool should be sourced from the same image for a consistent user experience. You control the resources published to users through application groups.

A host pool can be one of two types:

- Personal, where each session host is assigned to an individual user. Personal host pools provide dedicated desktops to end-users that optimize environments for performance and data separation.
- Pooled, where user sessions can be load balanced to any session host in the host pool. There can be multiple different users on a single session host at the same time. Pooled host pools provide a shared remote experience to end-users, which ensures lower costs and greater efficiency.

The following table goes into more detail about the differences between each type of host pool:

[\[+\] Expand table](#)

Feature	Personal host pools	Pooled host pools
Load balancing	User sessions are always load balanced to the session host the user is assigned to. If	User sessions are load balanced to session hosts in the host pool based

Feature	Personal host pools	Pooled host pools
	the user isn't currently assigned to a session host, the user session is load balanced to the next available session host in the host pool.	on user session count. You can choose which load balancing algorithm to use: breadth-first or depth-first.
Maximum session limit	One.	As configured by the maximum session limit value of the properties of a host pool. Under high concurrent connection load when multiple users connect to the host pool at the same time, the number of sessions created on a session host can exceed the maximum session limit.
User assignment process	Users can either be directly assigned to session hosts or be automatically assigned to the first available session host. Users always have sessions on the session hosts they are assigned to.	Users aren't assigned to session hosts. After a user signs out and signs back in, their user session might get load balanced to a different session host. To learn more, see Configure personal desktop assignment .
Scaling	Autoscale for personal host pools starts session host virtual machines according to schedule or using Start VM on Connect and then deallocates/hibernates session host virtual machines based on the user session state (log off/disconnect).	Autoscale for pooled host pools turns VMs on and off based on the capacity thresholds and schedules the customer defines.
Windows Updates	Updated with Windows Updates, Microsoft Configuration Manager (ConfigMgr) , or other software distribution configuration tools.	Updated by redeploying session hosts from updated images instead of traditional updates.
User data	Each user only ever uses one session host, so they can store their user profile data on the operating system (OS) disk of the VM.	Users can connect to different session hosts every time they connect, so they should store their user profile data in FSLogix .

Validation environment

You can set a host pool to be a [validation environment](#). Validation environments let you monitor service updates before the service applies them to your production or non-validation environment. Without a validation environment, you may not discover

changes that introduce errors, which could result in downtime for users in your production environment.

To ensure your apps work with the latest updates, the validation environment should be as similar to host pools in your non-validation environment as possible. Users should connect as frequently to the validation environment as they do to the production environment. If you have automated testing on your host pool, you should include automated testing on the validation environment.

Application groups

An [application group](#) is a logical grouping of applications installed on session hosts in the host pool.

An application group can be one of two types:

- RemoteApp, where users access the applications you individually select and publish to the application group. Available with pooled host pools only.
- Desktop, where users access the full desktop. Available with pooled or personal host pools.

Pooled host pools have a preferred application group type that dictates whether users see RemoteApp or Desktop apps in their feed if both resources have been published to the same user. By default, Azure Virtual Desktop automatically creates a Desktop application group with the friendly name **Default Desktop** whenever you create a host pool and sets the host pool's preferred application group type to **Desktop**. You can remove the Desktop application group at any time. If you want your users to only see applications in their feed, you should set the **preferred application group type** value to **RemoteApp**. If you want your users to only see session desktops in their feed, you should set the **preferred application group type** value to **Desktop**. You can't create another Desktop application group in a host pool while a Desktop application group exists.

To publish resources to users, you must assign them to application groups. When assigning users to application groups, consider the following things:

- We don't support assigning both the RemoteApp and desktop application groups in a single host pool to the same user. Doing so will cause a single user to have two user sessions in a single host pool. Users aren't supposed to have two active user sessions at the same time, as this can cause the following things to happen:
 - The session hosts become overloaded
 - Users get stuck when trying to login

- Connections won't work
- The screen turns black
- The application crashes
- Other negative effects on end-user experience and session performance
- A user can be assigned to multiple application groups within the same host pool, and their feed will be an accumulation of all application groups.
- Personal host pools only allow and support Desktop application groups.

ⓘ Note

If your host pool's *preferred application group type* is set to **Undefined**, that means you haven't set the value yet. You must finish configuring your host pool by setting its *preferred application group type* before you start using it to prevent app incompatibility and session host overload issues.

Workspaces

A [workspace](#) is a logical grouping of application groups in Azure Virtual Desktop. Each Azure Virtual Desktop application group must be associated with a workspace for users to see the desktops and applications published to them.

End users

After you've assigned users to their application groups, they can connect to an Azure Virtual Desktop deployment with any of the Azure Virtual Desktop clients.

User sessions

In this section, we'll go over each of the three types of user sessions that end users can have.

Active user session

A user session is considered *active* when a user signs in and connects to their desktop or RemoteApp resource.

Disconnected user session

A disconnected user session is an inactive session that the user hasn't signed out of yet. When a user closes the remote session window without signing out, the session becomes disconnected. When a user reconnects to their remote resources, they'll be redirected to their disconnected session on the session host they were working on. At this point, the disconnected session becomes an active session again.

Pending user session

A pending user session is a placeholder session that reserves a spot on the load-balanced virtual machine for the user. Because the sign-in process can take anywhere from 30 seconds to five minutes depending on the user profile, this placeholder session ensures that the user won't be kicked out of their session if another user completes their sign-in process first.

Next steps

Learn more about delegated access and how to assign roles to users at [Delegated Access in Azure Virtual Desktop](#).

To learn how to set up your Azure Virtual Desktop host pool, see [Create a host pool with the Azure portal](#).

To learn how to connect to Azure Virtual Desktop, see one of the following articles:

- [Connect with Windows](#)
- [Connect with the Azure Virtual Desktop Store app for Windows](#)
- [Connect with a web browser](#)
- [Connect with the Android client](#)
- [Connect with the macOS client](#)
- [Connect with the iOS client](#)
- [Connect with the Remote Desktop app for Windows](#)

Get started with the Azure Virtual Desktop Agent

Article • 05/15/2023

In the Azure Virtual Desktop Service framework, there are three main components: the Remote Desktop client, the service, and the virtual machines. These virtual machines live in the customer subscription where the Azure Virtual Desktop agent and agent bootloader are installed. The agent acts as the intermediate communicator between the service and the virtual machines, enabling connectivity. Therefore, if you're experiencing any issues with the agent installation, update, or configuration, your virtual machines won't be able to connect to the service. The agent bootloader is the executable that loads the agent.

This article will give you a brief overview of the agent installation and update processes.

ⓘ Note

This documentation is not for the FSLogix agent or the Remote Desktop Client agent.

Initial installation process

The Azure Virtual Desktop agent is initially installed in one of two ways. If you provision virtual machines (VMs) in the Azure portal and Azure Marketplace, the agent and agent bootloader are automatically installed. If you provision VMs using PowerShell, you must manually download the agent and agent bootloader .msi files when [creating a Azure Virtual Desktop host pool with PowerShell](#). Once the agent is installed, it installs the Azure Virtual Desktop side-by-side stack and Geneva Monitoring agent. The side-by-side stack component is required for users to securely establish reverse server-to-client connections. The Geneva Monitoring agent monitors the health of the agent. All three of these components are essential for end-to-end user connectivity to function properly.

ⓘ Important

To successfully install the Azure Virtual Desktop agent, side-by-side stack, and Geneva Monitoring agent, you must unblock all the URLs listed in the [Required URL list](#). Unblocking these URLs is required to use the Azure Virtual Desktop service.

Agent update process

The Azure Virtual Desktop service updates the agent whenever an update becomes available. Agent updates can include new functionality or fixes for previous issues. You must always have the latest stable version of the agent installed so your VMs don't lose connectivity or security. After you've installed the initial version of the Azure Virtual Desktop agent, the agent will regularly query the Azure Virtual Desktop service to determine if there's a newer version of the agent, stack, or monitoring agent available. If a newer version exists, the updated component is automatically installed by the flighting system, unless you've configured the Scheduled Agent Updates feature. If you've already configured the Scheduled Agent Updates feature, the agent will only install the updated components during the maintenance window that you specify. For more information, see [Scheduled Agent Updates](#).

New versions of the agent are deployed at regular intervals in five-day periods to all Azure subscriptions. These update periods are called "flights". It takes 24 hours for all VMs in a single broker region to receive the agent update in a flight. Because of this, when a flight happens, you may see VMs in your host pool receive the agent update at different times. Also, if the VMs are in different regions, they might update on different days in the five-day period. The flight will update all VM agents in all subscriptions by the end of the deployment period. The Azure Virtual Desktop flighting system enhances service reliability by ensuring the stability and quality of the agent update.

Other important things you should keep in mind:

- The agent update isn't connected to Azure Virtual Desktop infrastructure build updates. When the Azure Virtual Desktop infrastructure updates, that doesn't mean that the agent has updated along with it.
- Because VMs in your host pool may receive agent updates at different times, you'll need to be able to tell the difference between flighting issues and failed agent updates. If you go to the event logs for your VM at **Event Viewer > Windows Logs > Application** and see an event labeled "ID 3277," that means the Agent update didn't work. If you don't see that event, then the VM is in a different flight and will be updated later. See [Set up diagnostics to monitor agent updates](#) for more information about how to set up diagnostic logs to track updates and make sure they've been installed correctly.
- When the Geneva Monitoring agent updates to the latest version, the old GenevaTask task is located and disabled before creating a new task for the new monitoring agent. The earlier version of the monitoring agent isn't deleted in case that the most recent version of the monitoring agent has a problem that requires reverting to the earlier version to fix. If the latest version has a problem, the old monitoring agent will be re-enabled to continue delivering monitoring data. All

versions of the monitor that are earlier than the last one you installed before the update will be deleted from your VM.

- Your VM keeps three versions of the agent and of the side-by-side stack at a time. This allows for quick recovery if something goes wrong with the update. The earliest version of the agent or stack is removed from the VM whenever the agent or stack updates. If you delete these components prematurely and the agent or stack has a failure, the agent or stack won't be able to roll back to an earlier version, which will put your VM in an unavailable state.

The agent update normally lasts 2-3 minutes on a new VM and shouldn't cause your VM to lose connection or shut down. This update process applies to both Azure Virtual Desktop (classic) and the latest version of Azure Virtual Desktop with Azure Resource Manager.

Next steps

Now that you have a better understanding of the Azure Virtual Desktop agent, here are some resources that might help you:

- If you're experiencing agent or connectivity-related issues, check out the [Azure Virtual Desktop Agent issues troubleshooting guide](#).
- To schedule agent updates, see the [Scheduled Agent Updates document](#).
- To set up diagnostics for this feature, see the [Scheduled Agent Updates Diagnostics guide](#).
- To find information about the latest and previous agent versions, see the [Agent Updates version notes](#).

Supported identities and authentication methods

Article • 11/14/2023

In this article, we'll give you a brief overview of what kinds of identities and authentication methods you can use in Azure Virtual Desktop.

Identities

Azure Virtual Desktop supports different types of identities depending on which configuration you choose. This section explains which identities you can use for each configuration.

Important

Azure Virtual Desktop doesn't support signing in to Microsoft Entra ID with one user account, then signing in to Windows with a separate user account. Signing in with two different accounts at the same time can lead to users reconnecting to the wrong session host, incorrect or missing information in the Azure portal, and error messages appearing while using MSIX app attach.

On-premises identity

Since users must be discoverable through Microsoft Entra ID to access the Azure Virtual Desktop, user identities that exist only in Active Directory Domain Services (AD DS) aren't supported. This includes standalone Active Directory deployments with Active Directory Federation Services (AD FS).

Hybrid identity

Azure Virtual Desktop supports [hybrid identities](#) through Microsoft Entra ID, including those federated using AD FS. You can manage these user identities in AD DS and sync them to Microsoft Entra ID using [Microsoft Entra Connect](#). You can also use Microsoft Entra ID to manage these identities and sync them to [Microsoft Entra Domain Services](#).

When accessing Azure Virtual Desktop using hybrid identities, sometimes the User Principal Name (UPN) or Security Identifier (SID) for the user in Active Directory (AD) and Microsoft Entra ID don't match. For example, the AD account user@contoso.local may

correspond to user@contoso.com in Microsoft Entra ID. Azure Virtual Desktop only supports this type of configuration if either the UPN or SID for both your AD and Microsoft Entra ID accounts match. SID refers to the user object property "ObjectSID" in AD and "OnPremisesSecurityIdentifier" in Microsoft Entra ID.

Cloud-only identity

Azure Virtual Desktop supports cloud-only identities when using [Microsoft Entra joined VMs](#). These users are created and managed directly in Microsoft Entra ID.

 **Note**

You can also assign hybrid identities to Azure Virtual Desktop Application groups that host Session hosts of join type Microsoft Entra joined.

Third-party identity providers

If you're using an Identity Provider (IdP) other than Microsoft Entra ID to manage your user accounts, you must ensure that:

- Your IdP is [federated with Microsoft Entra ID](#).
- Your session hosts are Microsoft Entra joined or [Microsoft Entra hybrid joined](#).
- You enable [Microsoft Entra authentication](#) to the session host.

External identity

Azure Virtual Desktop currently doesn't support [external identities](#).

Service authentication

To access Azure Virtual Desktop resources, you must first authenticate to the service by signing in with a Microsoft Entra account. Authentication happens whenever you subscribe to a workspace to retrieve your resources and connect to apps or desktops. You can use [third-party identity providers](#) as long as they federate with Microsoft Entra ID.

Multifactor authentication

Follow the instructions in [Enforce Microsoft Entra multifactor authentication for Azure Virtual Desktop using Conditional Access](#) to learn how to enforce Microsoft Entra

multifactor authentication for your deployment. That article will also tell you how to configure how often your users are prompted to enter their credentials. When deploying Microsoft Entra joined VMs, note the extra steps for [Microsoft Entra joined session host VMs](#).

Passwordless authentication

You can use any authentication type supported by Microsoft Entra ID, such as [Windows Hello for Business](#) and other [passwordless authentication options](#) (for example, FIDO keys), to authenticate to the service.

Smart card authentication

To use a smart card to authenticate to Microsoft Entra ID, you must first [configure AD FS for user certificate authentication](#) or [configure Microsoft Entra certificate-based authentication](#).

Session host authentication

If you haven't already enabled [single sign-on](#) or saved your credentials locally, you'll also need to authenticate to the session host when launching a connection. The following list describes which types of authentication each Azure Virtual Desktop client currently supports.

Client	Supported authentication type(s)
Windows Desktop client	Username and password Smart card Windows Hello for Business certificate trust Windows Hello for Business key trust with certificates Microsoft Entra authentication
Azure Virtual Desktop Store app	Username and password Smart card Windows Hello for Business certificate trust Windows Hello for Business key trust with certificates Microsoft Entra authentication
Remote Desktop app	Username and password
Web client	Username and password
Android client	Username and password

Client	Supported authentication type(s)
iOS client	Username and password
macOS client	Username and password Smart card: support for smart card-based sign in using smart card redirection at the Winlogon prompt when NLA is not negotiated.

ⓘ Important

In order for authentication to work properly, your local machine must also be able to access the [required URLs for Remote Desktop clients](#).

Single sign-on (SSO)

SSO allows the connection to skip the session host credential prompt and automatically sign the user in to Windows. For session hosts that are Microsoft Entra joined or Microsoft Entra hybrid joined, it's recommended to enable [SSO using Microsoft Entra authentication](#). Microsoft Entra authentication provides other benefits including passwordless authentication and support for third-party identity providers.

Azure Virtual Desktop also supports [SSO using Active Directory Federation Services \(AD FS\)](#) for the Windows Desktop and web clients.

Without SSO, the client will prompt users for their session host credentials for every connection. The only way to avoid being prompted is to save the credentials in the client. We recommend you only save credentials on secure devices to prevent other users from accessing your resources.

Smart card and Windows Hello for Business

Azure Virtual Desktop supports both NT LAN Manager (NTLM) and Kerberos for session host authentication, however Smart card and Windows Hello for Business can only use Kerberos to sign in. To use Kerberos, the client needs to get Kerberos security tickets from a Key Distribution Center (KDC) service running on a domain controller. To get tickets, the client needs a direct networking line-of-sight to the domain controller. You can get a line-of-sight by connecting directly within your corporate network, using a VPN connection or setting up a [KDC Proxy server](#).

In-session authentication

Once you're connected to your RemoteApp or desktop, you may be prompted for authentication inside the session. This section explains how to use credentials other than username and password in this scenario.

In-session passwordless authentication

Azure Virtual Desktop supports in-session passwordless authentication using [Windows Hello for Business](#) or security devices like FIDO keys when using the [Windows Desktop client](#). Passwordless authentication is enabled automatically when the session host and local PC are using the following operating systems:

- Windows 11 single or multi-session with the [2022-10 Cumulative Updates for Windows 11 \(KB5018418\)](#) or later installed.
- Windows 10 single or multi-session, versions 20H2 or later with the [2022-10 Cumulative Updates for Windows 10 \(KB5018410\)](#) or later installed.
- Windows Server 2022 with the [2022-10 Cumulative Update for Microsoft server operating system \(KB5018421\)](#) or later installed.

To disable passwordless authentication on your host pool, you must [customize an RDP property](#). You can find the **WebAuthn redirection** property under the **Device redirection** tab in the Azure portal or set the **redirectwebauthn** property to **0** using PowerShell.

When enabled, all WebAuthn requests in the session are redirected to the local PC. You can use Windows Hello for Business or locally attached security devices to complete the authentication process.

To access Microsoft Entra resources with Windows Hello for Business or security devices, you must enable the FIDO2 Security Key as an authentication method for your users. To enable this method, follow the steps in [Enable FIDO2 security key method](#).

In-session smart card authentication

To use a smart card in your session, make sure you've installed the smart card drivers on the session host and enabled [smart card redirection](#). Review the [client comparison chart](#) to make sure your client supports smart card redirection.

Next steps

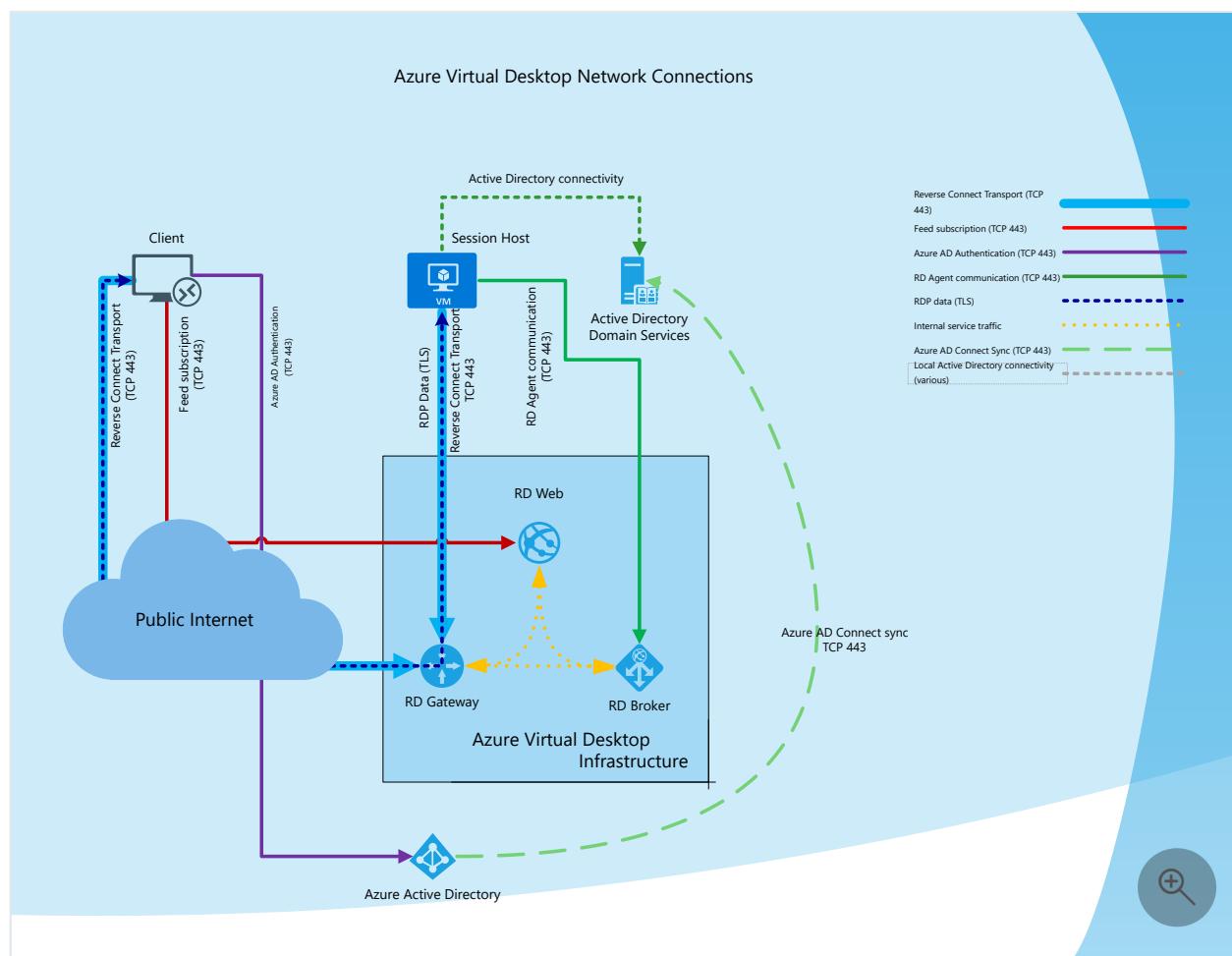
- Curious about other ways to keep your deployment secure? Check out [Security best practices](#).

- Having issues connecting to Microsoft Entra joined VMs? Look at [Troubleshoot connections to Microsoft Entra joined VMs](#).
- Having issues with in-session passwordless authentication? See [Troubleshoot WebAuthn redirection](#).
- Want to use smart cards from outside your corporate network? Review how to set up a [KDC Proxy server](#).

Understanding Azure Virtual Desktop network connectivity

Article • 06/12/2023

Azure Virtual Desktop provides the ability to host client sessions on the session hosts running on Azure. Microsoft manages portions of the services on the customer's behalf and provides secure endpoints for connecting clients and session hosts. The diagram below gives a high-level overview of the network connections used by Azure Virtual Desktop



Session connectivity

Azure Virtual Desktop uses Remote Desktop Protocol (RDP) to provide remote display and input capabilities over network connections. RDP was initially released with Windows NT 4.0 Terminal Server Edition and was continuously evolving with every Microsoft Windows and Windows Server release. From the beginning, RDP developed to be independent of its underlying transport stack, and today it supports multiple types of transport.

Reverse connect transport

Azure Virtual Desktop is using reverse connect transport for establishing the remote session and for carrying RDP traffic. Unlike the on-premises Remote Desktop Services deployments, reverse connect transport doesn't use a TCP listener to receive incoming RDP connections. Instead, it is using outbound connectivity to the Azure Virtual Desktop infrastructure over the HTTPS connection.

Session host communication channel

Upon startup of the Azure Virtual Desktop session host, the Remote Desktop Agent Loader service establishes the Azure Virtual Desktop broker's persistent communication channel. This communication channel is layered on top of a secure Transport Layer Security (TLS) connection and serves as a bus for service message exchange between session host and Azure Virtual Desktop infrastructure.

Client connection sequence

Client connection sequence described below:

1. Using supported Azure Virtual Desktop client user subscribes to the Azure Virtual Desktop Workspace
2. Azure Active Directory authenticates the user and returns the token used to enumerate resources available to a user
3. Client passes token to the Azure Virtual Desktop feed subscription service
4. Azure Virtual Desktop feed subscription service validates the token
5. Azure Virtual Desktop feed subscription service passes the list of available desktops and RemoteApps back to the client in the form of digitally signed connection configuration
6. Client stores the connection configuration for each available resource in a set of .rdp files
7. When a user selects the resource to connect, the client uses the associated .rdp file and establishes the secure TLS 1.2 connection to an Azure Virtual Desktop gateway instance with the help of [Azure Front Door](#) and passes the connection information. The latency from all gateways is evaluated, and the gateways are put into groups of 10ms. The gateway with the lowest latency and then lowest number of existing connections is chosen.
8. Azure Virtual Desktop gateway validates the request and asks the Azure Virtual Desktop broker to orchestrate the connection

9. Azure Virtual Desktop broker identifies the session host and uses the previously established persistent communication channel to initialize the connection
10. Remote Desktop stack initiates the TLS 1.2 connection to the same Azure Virtual Desktop gateway instance as used by the client
11. After both client and session host connected to the gateway, the gateway starts relaying the raw data between both endpoints, this establishes the base reverse connect transport for the RDP
12. After the base transport is set, the client starts the RDP handshake

Connection security

TLS 1.2 is used for all connections initiated from the clients and session hosts to the Azure Virtual Desktop infrastructure components. Azure Virtual Desktop uses the same TLS 1.2 ciphers as [Azure Front Door](#). It's important to make sure both client computers and session hosts can use these ciphers. For reverse connect transport, both client and session host connect to the Azure Virtual Desktop gateway. After establishing the TCP connection, the client or session host validates the Azure Virtual Desktop gateway's certificate. After establishing the base transport, RDP establishes a nested TLS connection between client and session host using the session host's certificates. By default, the certificate used for RDP encryption is self-generated by the OS during the deployment. If desired, customers may deploy centrally managed certificates issued by the enterprise certification authority. For more information about configuring certificates, see [Windows Server documentation](#).

Next steps

- To learn about bandwidth requirements for Azure Virtual Desktop, see [Understanding Remote Desktop Protocol \(RDP\) Bandwidth Requirements for Azure Virtual Desktop](#).
- To get started with Quality of Service (QoS) for Azure Virtual Desktop, see [Implement Quality of Service \(QoS\) for Azure Virtual Desktop](#).

RDP Shortpath for Azure Virtual Desktop

Article • 03/10/2023

ⓘ Important

Using RDP Shortpath for public networks with TURN for Azure Virtual Desktop is currently in PREVIEW. See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Connections to Azure Virtual Desktop use Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). RDP Shortpath is a feature of Azure Virtual Desktop that establishes a direct UDP-based transport between a supported Windows Remote Desktop client and session host. By default, Remote Desktop Protocol (RDP) tries to establish connection using UDP and uses a TCP-based reverse connect transport as a fallback connection mechanism. TCP-based reverse connect transport provides the best compatibility with various networking configurations and has a high success rate for establishing RDP connections. UDP-based transport offers better connection reliability and more consistent latency.

RDP Shortpath can be used in two ways:

1. **Managed networks**, where direct connectivity is established between the client and the session host when using a private connection, such as a virtual private network (VPN).
2. **Public networks**, where direct connectivity is established between the client and the session host when using a public connection. There are two connection types when using a public connection, which are listed here in order of preference:
 - a. A *direct* UDP connection using the Simple Traversal Underneath NAT (STUN) protocol between a client and session host.
 - b. An *indirect* UDP connection using the Traversal Using Relay NAT (TURN) protocol with a relay between a client and session host. This is in preview.

The transport used for RDP Shortpath is based on the [Universal Rate Control Protocol \(URCP\)](#). URCP enhances UDP with active monitoring of the network conditions and

provides fair and full link utilization. URCP operates at low delay and loss levels as needed.

ⓘ Important

- During the preview, TURN is only available for connections to session hosts in a validation host pool. To configure your host pool as a validation environment, see [Define your host pool as a validation environment](#).
- RDP Shortpath for public networks with TURN is only available in the Azure public cloud.

Key benefits

Using RDP Shortpath has the following key benefits:

- Using URCP to enhance UDP achieves the best performance by dynamically learning network parameters and providing the protocol with a rate control mechanism.
- The removal of extra relay points reduces round-trip time, which improves connection reliability and user experience with latency-sensitive applications and input methods.
- In addition, for managed networks:
 - RDP Shortpath brings support for configuring Quality of Service (QoS) priority for RDP connections through Differentiated Services Code Point (DSCP) marks.
 - The RDP Shortpath transport allows limiting outbound network traffic by specifying a throttle rate for each session.

How RDP Shortpath works

To learn how RDP Shortpath works for managed networks and public networks, select each of the following tabs.

Managed networks

You can achieve the direct line of sight connectivity required to use RDP Shortpath with managed networks using the following methods.

- ExpressRoute private peering
- Site-to-site or Point-to-site VPN (IPsec), such as [Azure VPN Gateway](#)

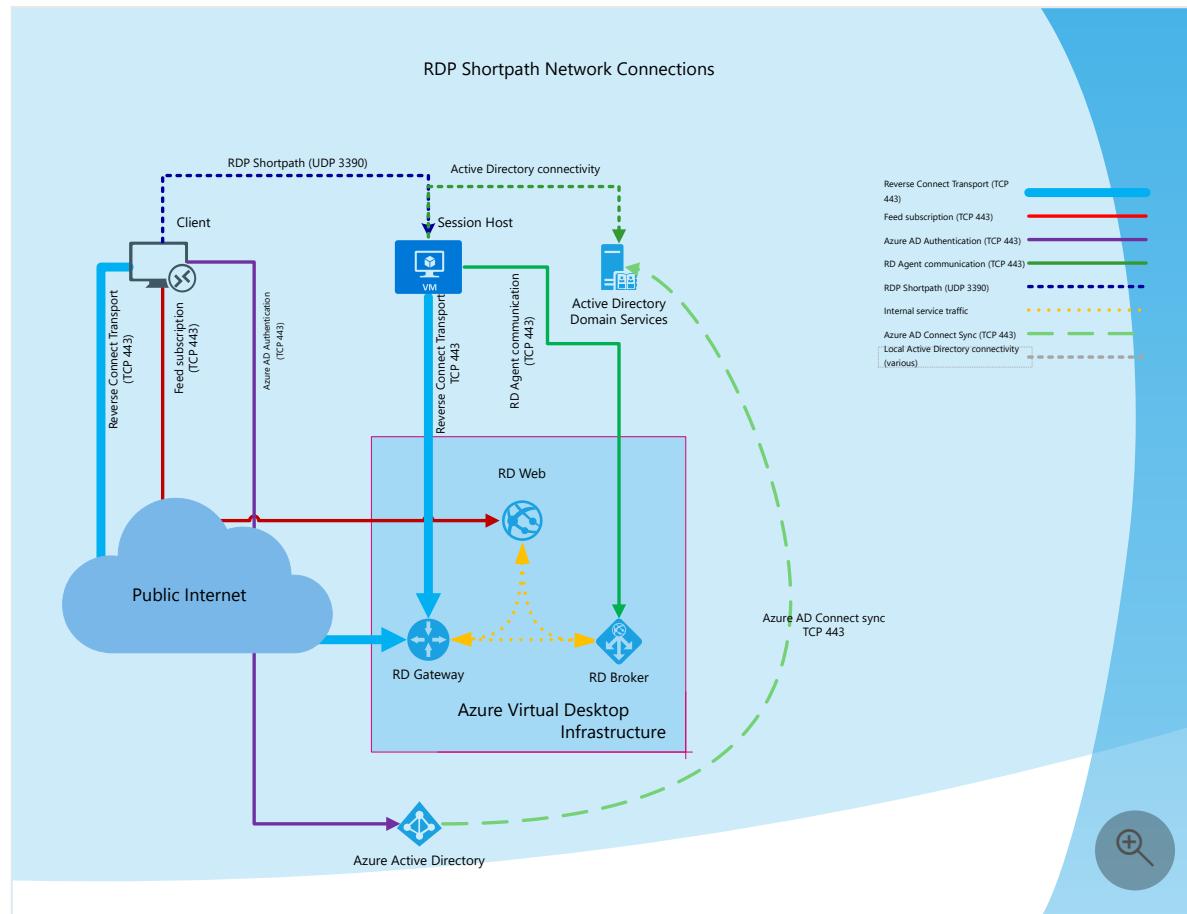
Having direct line of sight connectivity means that the client can connect directly to the session host without being blocked by firewalls.

 **Note**

If you're using other VPN types to connect to Azure, we recommend using a UDP-based VPN. While most TCP-based VPN solutions support nested UDP, they add inherited overhead of TCP congestion control, which slows down RDP performance.

To use RDP Shortpath for managed networks, you must enable a UDP listener on your session hosts. By default, port **3390** is used, although you can use a different port.

The following diagram gives a high-level overview of the network connections when using RDP Shortpath for managed networks and session hosts joined to an Active Directory domain.



Connection sequence

All connections begin by establishing a TCP-based [reverse connect transport](#) over the Azure Virtual Desktop Gateway. Then, the client and session host establish the initial RDP transport, and start exchanging their capabilities. These capabilities are negotiated using the following process:

1. The session host sends the list of its IPv4 and IPv6 addresses to the client.
2. The client starts the background thread to establish a parallel UDP-based transport directly to one of the session host's IP addresses.
3. While the client is probing the provided IP addresses, it continues to establish the initial connection over the reverse connect transport to ensure there's no delay in the user connection.
4. If the client has a direct connection to the session host, the client establishes a secure connection using TLS over reliable UDP.
5. After establishing the RDP Shortpath transport, all Dynamic Virtual Channels (DVCs), including remote graphics, input, and device redirection, are moved to the new transport. However, if a firewall or network topology prevents the client from establishing direct UDP connectivity, RDP continues with a reverse connect transport.

If your users have both RDP Shortpath for managed network and public networks available to them, then the first-found algorithm will be used. The user will use whichever connection gets established first for that session.

Connection security

RDP Shortpath extends RDP multi-transport capabilities. It doesn't replace the reverse connect transport but complements it. Initial session brokering is managed through the Azure Virtual Desktop service and the reverse connect transport. All connection attempts are ignored unless they match the reverse connect session first. RDP Shortpath is established after authentication, and if successfully established, the reverse connect transport is dropped and all traffic flows over the RDP Shortpath.

RDP Shortpath uses a secure connection using TLS over reliable UDP between the client and the session host using the session host's certificates. By default, the certificate used for RDP encryption is self-generated by the operating system during the deployment. You can also deploy centrally managed certificates issued by an enterprise certification

authority. For more information about certificate configurations, see [Remote Desktop listener certificate configurations](#).

ⓘ Note

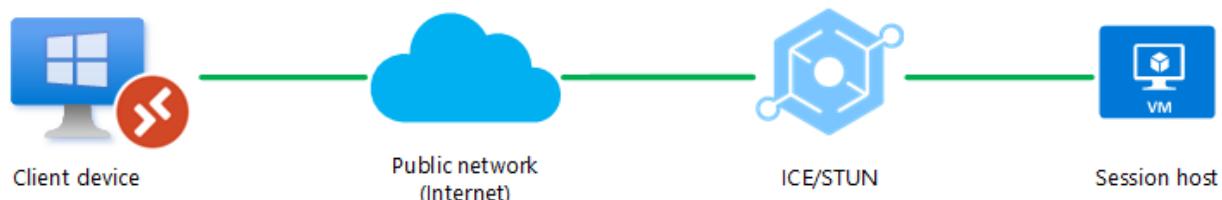
The security offered by RDP Shortpath is the same as that offered by TCP reverse connect transport.

Example scenarios

Here are some example scenarios to show how connections are evaluated to decide whether RDP Shortpath is used across different network topologies.

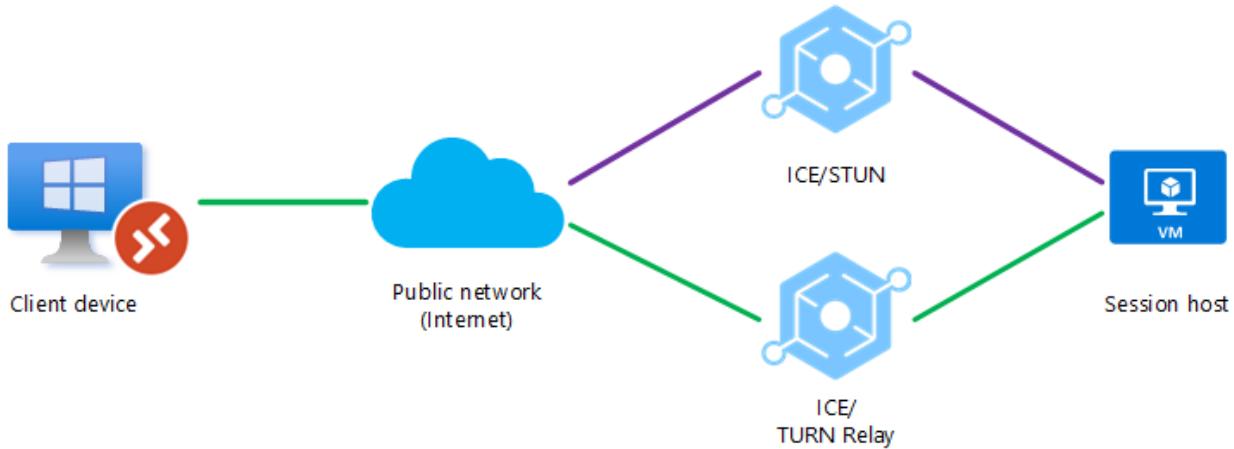
Scenario 1

A UDP connection can only be established between the client device and the session host over a public network (internet). A direct connection, such as a VPN, isn't available. UDP is allowed through firewall or NAT device.



Scenario 2

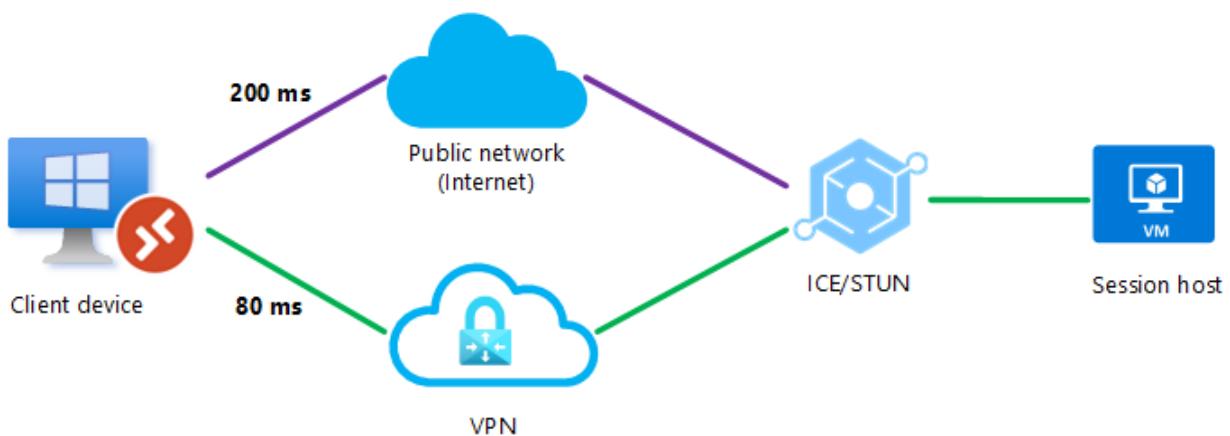
A firewall or NAT device is blocking a direct UDP connection, but an indirect UDP connection can be relayed using TURN between the client device and the session host over a public network (internet). Another direct connection, such as a VPN, isn't available.



Scenario 3

A UDP connection can be established between the client device and the session host over a public network or over a direct VPN connection, but RDP Shortpath for managed networks isn't enabled. When the client initiates the connection, the ICE/STUN protocol can see multiple routes and will evaluate each route and choose the one with the lowest latency.

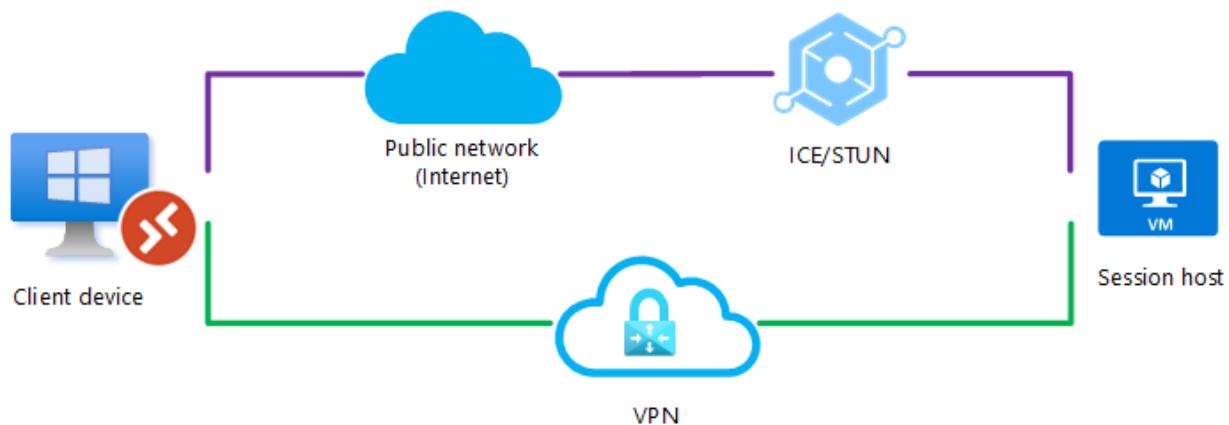
In this example, a UDP connection using RDP Shortpath for public networks over the direct VPN connection will be made as it has the lowest latency, as shown by the green line.



Scenario 4

Both RDP Shortpath for public networks and managed networks are enabled. A UDP connection can be established between the client device and the session host over a public network or over a direct VPN connection. When the client initiates the connection, there are simultaneous attempts to connect using RDP Shortpath for managed networks through port 3390 (by default) and RDP Shortpath for public networks through the ICE/STUN protocol. The first-found algorithm will be used and the user will use whichever connection gets established first for that session.

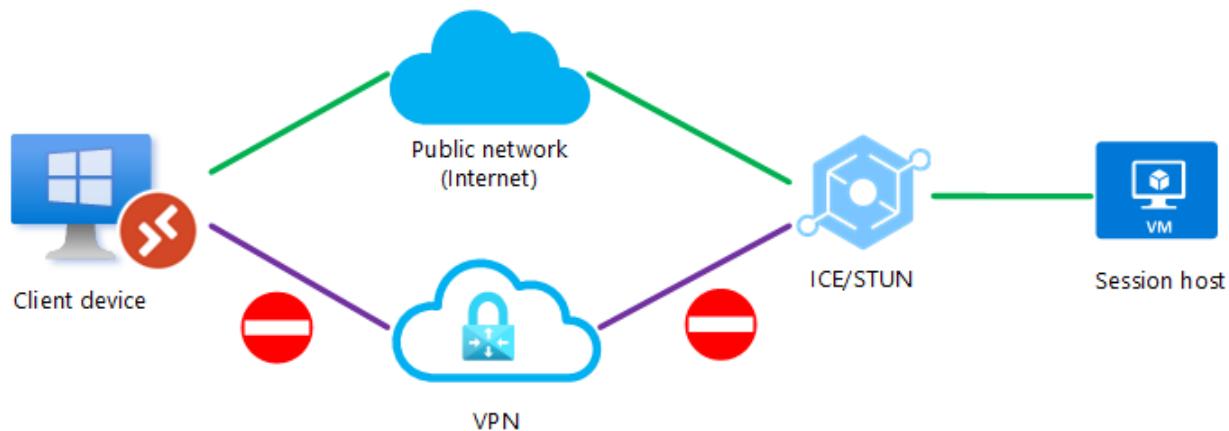
Since going over a public network has more steps, for example a NAT device, a load balancer, or a STUN server, it's likely that the first-found algorithm will select the connection using RDP Shortpath for managed networks and be established first.



Scenario 5

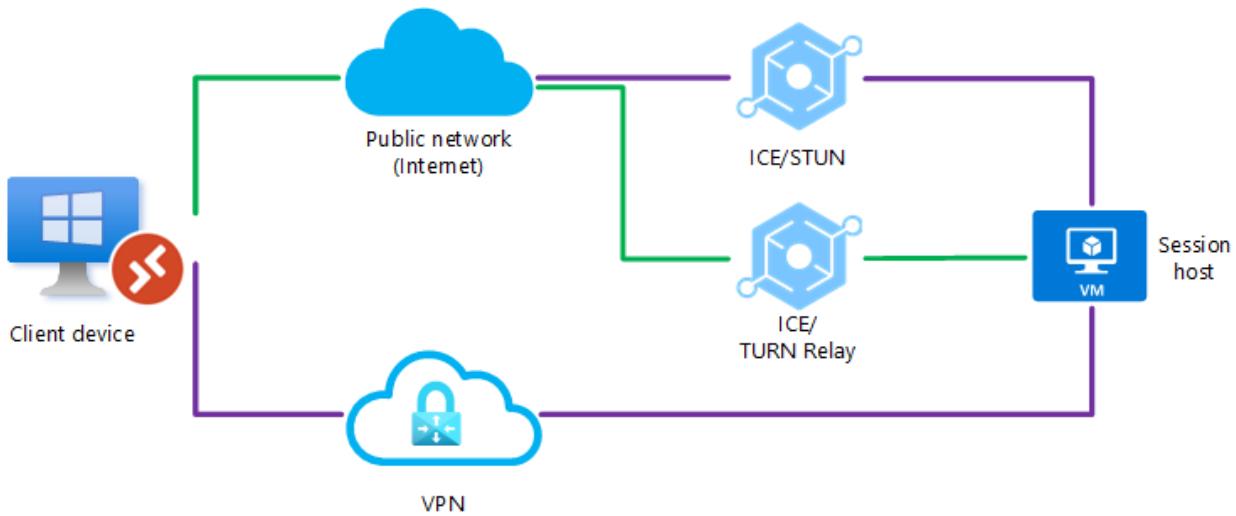
A UDP connection can be established between the client device and the session host over a public network or over a direct VPN connection, but RDP Shortpath for managed networks isn't enabled. To prevent ICE/STUN from using a particular route, an admin can block one of the routes for UDP traffic. Blocking a route would ensure the remaining path is always used.

In this example, UDP is blocked on the direct VPN connection and the ICE/STUN protocol establishes a connection over the public network.



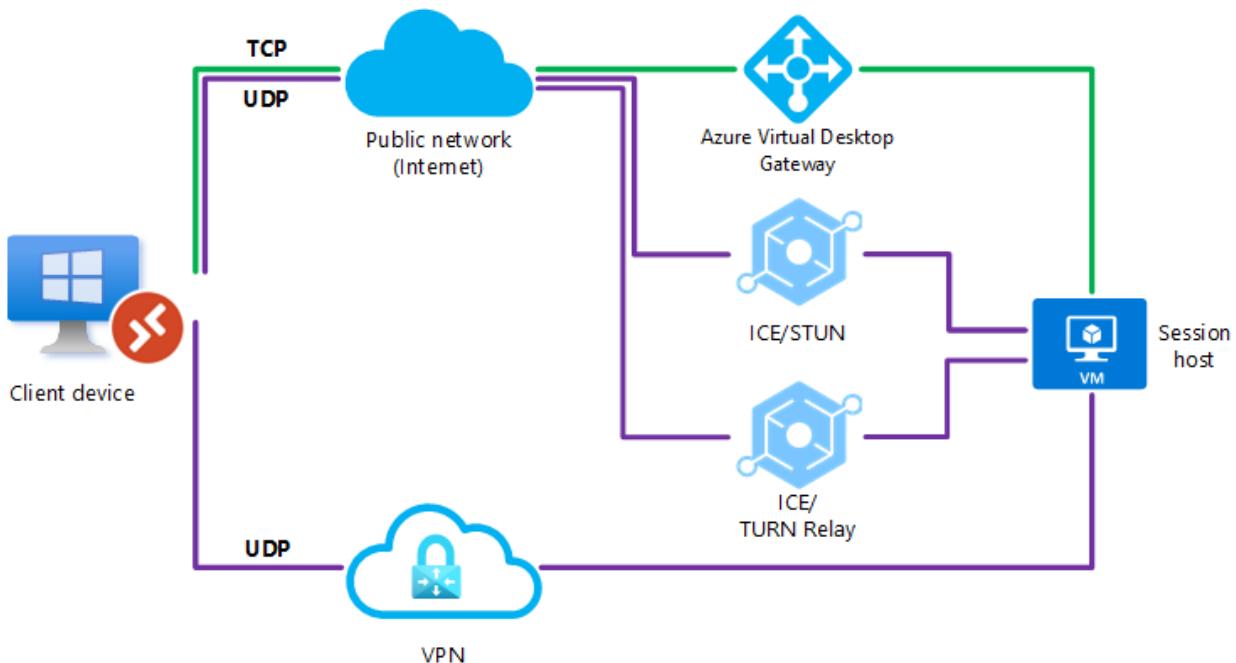
Scenario 6

Both RDP Shortpath for public networks and managed networks are configured, however a UDP connection couldn't be established using direct VPN connection. A firewall or NAT device is also blocking a direct UDP connection using the public network (internet), but an indirect UDP connection can be relayed using TURN between the client device and the session host over a public network (internet).



Scenario 7

Both RDP Shortpath for public networks and managed networks are configured, however a UDP connection couldn't be established. In this instance, RDP Shortpath will fail and the connection will fall back to TCP-based reverse connect transport.



Next steps

- Learn how to [Configure RDP Shortpath](#).
- Learn more about Azure Virtual Desktop network connectivity at [Understanding Azure Virtual Desktop network connectivity](#).
- Understand [Azure egress network charges](#).
- To understand how to estimate the bandwidth used by RDP, see [RDP bandwidth requirements](#).

Implement Quality of Service (QoS) for Azure Virtual Desktop

Article • 05/25/2022

[RDP Shortpath for managed networks](#) provides a direct UDP-based transport between Remote Desktop Client and Session host. RDP Shortpath for managed networks enables configuration of Quality of Service (QoS) policies for the RDP data. QoS in Azure Virtual Desktop allows real-time RDP traffic that's sensitive to network delays to "cut in line" in front of traffic that's less sensitive. Example of such less sensitive traffic would be a downloading a new app, where an extra second to download isn't a large deal. QoS uses Windows Group Policy Objects to identify and mark all packets in real-time streams and help your network to give RDP traffic a dedicated portion of bandwidth.

If you support a large group of users experiencing any of the problems described in this article, you probably need to implement QoS. A small business with few users might not need QoS, but it should be helpful even there.

Without some form of QoS, you might see the following issues:

- Jitter – RDP packets arriving at different rates, which can result in visual and audio glitches
- Packet loss – packets dropped, which results in retransmission that requires additional time
- Delayed round-trip time (RTT) – RDP packets taking a long time to reach their destinations, which result in noticeable delays between input and reaction from the remote application.

The least complicated way to address these issues is to increase the data connections' size, both internally and out to the internet. Since that is often cost-prohibitive, QoS provides a way to manage the resources you have instead of adding bandwidth more effectively. To address quality issues, we recommend that you first use QoS, then add bandwidth only where necessary.

For QoS to be effective, you must apply consistent QoS settings throughout your organization. Any part of the path that fails to support your QoS priorities can degrade the quality RDP session.

Introduction to QoS queues

To provide QoS, network devices must have a way to classify traffic and must be able to distinguish RDP from other network traffic.

When network traffic enters a router, the traffic is placed into a queue. If a QoS policy isn't configured, there is only one queue, and all data is treated as first-in, first-out with the same priority. That means RDP traffic might get stuck behind traffic where a few extra milliseconds delay wouldn't be a problem.

When you implement QoS, you define multiple queues using one of several congestion management features, such as Cisco's priority queuing and [Class-Based Weighted Fair Queueing \(CBWFQ\)](#) and congestion avoidance features, such as [weighted random early detection \(WRED\)](#).

A simple analogy is that QoS creates virtual "carpool lanes" in your data network. So some types of data never or rarely encounter a delay. Once you create those lanes, you can adjust their relative size and much more effectively manage the connection bandwidth you have while still delivering business-grade experiences for your organization's users.

QoS implementation checklist

At a high level, do the following to implement QoS:

1. [Make sure your network is ready](#)
2. [Make sure that RDP Shortpath for managed networks is enabled](#) - QoS policies are not supported for reverse connect transport
3. [Implement insertion of DSCP markers on session hosts](#)

As you prepare to implement QoS, keep the following guidelines in mind:

- The shortest path to session host is best
- Any obstacles in between, such as proxies or packet inspection devices, aren't recommended

Make sure your network is ready

If you're considering a QoS implementation, you should already have determined your bandwidth requirements and other [network requirements](#).

Traffic congestion across a network will significantly impact media quality. A lack of bandwidth leads to performance degradation and a poor user experience. As Azure

Virtual Desktop adoption and usage grows, use [Log Analytics](#) to identify problems and then make adjustments using QoS and selective bandwidth additions.

VPN considerations

QoS only works as expected when implemented on all links between clients and session hosts. If you use QoS on an internal network and a user signs in from a remote location, you can only prioritize within your internal, managed network. Although remote locations can receive a managed connection by implementing a virtual private network (VPN), a VPN inherently adds packet overhead and creates delays in real-time traffic.

In a global organization with managed links that span continents, we strongly recommend QoS because bandwidth for those links is limited compared to the LAN.

Insert DSCP markers

You could implement QoS using a Group Policy Object (GPO) to direct session hosts to insert a DSCP marker in IP packet headers identifying it as a particular type of traffic. Routers and other network devices can be configured to recognize these markings and put the traffic in a separate, higher-priority queue.

You can compare DSCP markings to postage stamps that indicate to postal workers how urgent the delivery is and how best to sort it for speedy delivery. Once you've configured your network to give priority to RDP streams, lost packets and late packets should diminish significantly.

Once all network devices are using the same classifications, markings, and priorities, it's possible to reduce or eliminate delays, dropped packets, and jitter. From the RDP perspective, the essential configuration step is the classification and marking of packets. However, for end-to-end QoS to be successful, you also need to align the RDP configuration with the underlying network configuration carefully. The DSCP value tells a correspondingly configured network what priority to give a packet or stream.

We recommend using DSCP value 46 that maps to **Expedited Forwarding (EF)** DSCP class.

Implement QoS on session host using Group Policy

You can use policy-based Quality of Service (QoS) within Group Policy to set the predefined DSCP value.

To create a QoS policy for domain-joined session hosts, first, sign in to a computer on which Group Policy Management has been installed. Open Group Policy Management (select Start, point to Administrative Tools, and then select Group Policy Management), and then complete the following steps:

1. In Group Policy Management, locate the container where the new policy should be created. For example, if all your session hosts computers are located in an OU named "**session hosts**", the new policy should be created in the Session Hosts OU.
2. Right-click the appropriate container, and then select **Create a GPO in this domain, and Link it here**.
3. In the **New GPO** dialog box, type a name for the new Group Policy object in the **Name** box, and then select **OK**.
4. Right-click the newly created policy, and then select **Edit**.
5. In the Group Policy Management Editor, expand **Computer Configuration**, expand **Windows Settings**, right-click **Policy-based QoS**, and then select **Create new policy**.
6. In the **Policy-based QoS** dialog box, on the opening page, type a name for the new policy in the **Name** box. Select **Specify DSCP Value** and set the value to **46**. Leave **Specify Outbound Throttle Rate** unselected, and then select **Next**.
7. On the next page, select **Only applications with this executable name** and enter the name **svchost.exe**, and then select **Next**. This setting instructs the policy to only prioritize matching traffic from the Remote Desktop Service.
8. On the third page, make sure that both **Any source IP address** and **Any destination IP address** are selected, and then select **Next**. These two settings ensure that packets will be managed regardless of which computer (IP address) sent the packets and which computer (IP address) will receive the packets.
9. On page four, select **UDP** from the **Select the protocol this QoS policy applies to** drop-down list.
10. Under the heading **Specify the source port number**, select **From this source port or range**. In the accompanying text box, type **3390**. Select **Finish**.

The new policies you've created won't take effect until Group Policy has been refreshed on your session host computers. Although Group Policy periodically refreshes on its own, you can force an immediate refresh by following these steps:

1. On each session host for which you want to refresh Group Policy, open a Command Prompt as administrator (*Run as administrator*).
2. At the command prompt, enter

```
Console
```

```
gpupdate /force
```

Implement QoS on session host using PowerShell

You can set QoS for RDP Shortpath for managed networks using the PowerShell cmdlet below:

```
PowerShell
```

```
New-NetQosPolicy -Name "RDP Shortpath for managed networks" -  
AppPathNameMatchCondition "svchost.exe" -IPProtocolMatchCondition UDP -  
IPSrcPortStartMatchCondition 3390 -IPSrcPortEndMatchCondition 3390 -  
DSCPAction 46 -NetworkProfile All
```

Related articles

- [Quality of Service \(QoS\) Policy](#)

Next steps

- To learn about bandwidth requirements for Azure Virtual Desktop, see [Understanding Remote Desktop Protocol \(RDP\) Bandwidth Requirements for Azure Virtual Desktop](#).
- To learn about Azure Virtual Desktop network connectivity, see [Understanding Azure Virtual Desktop network connectivity](#).

Remote Desktop Protocol (RDP) bandwidth requirements

Article • 05/25/2022

Remote Desktop Protocol (RDP) is a sophisticated technology that uses various techniques to perfect the server's remote graphics' delivery to the client device. Depending on the use case, availability of computing resources, and network bandwidth, RDP dynamically adjusts various parameters to deliver the best user experience.

Remote Desktop Protocol multiplexes multiple Dynamic Virtual Channels (DVCs) into a single data channel sent over different network transports. There are separate DVCs for remote graphics, input, device redirection, printing, and more. Azure Virtual Desktop partners can also use their extensions that use DVC interfaces.

The amount of the data sent over RDP depends on the user activity. For example, a user may work with basic textual content for most of the session and consume minimal bandwidth, but then generate a printout of a 200-page document to the local printer. This print job will use a significant amount of network bandwidth.

When using a remote session, your network's available bandwidth dramatically impacts the quality of your experience. Different applications and display resolutions require different network configurations, so it's essential to make sure your network configuration meets your needs.

Estimating bandwidth utilization

RDP uses various compression algorithms for different types of data. The table below guides estimating of the data transfers:

Type of Data	Direction	How to estimate
Remote Graphics	Session host to client	See the detailed guidelines
Heartbeats	Both directions	~ 20 bytes every 5 seconds
Input	Client to session Host	Amount of data is based on the user activity, less than 100 bytes for most of the operations

Type of Data	Direction	How to estimate
File transfers	Both directions	File transfers are using bulk compression. Use .zip compression for approximation
Printing	Session host to client	Print job transfer depends on the driver and using bulk compression, use .zip compression for approximation

Other scenarios can have their bandwidth requirements change depending on how you use them, such as:

- Voice or video conferencing
- Real-time communication
- Streaming 4K video

Estimating bandwidth used by remote graphics

It's tough to predict bandwidth use by the remote desktop. The user activities generate most of the remote desktop traffic. Every user is unique, and differences in their work patterns may significantly change network use.

The best way to understand bandwidth requirements is to monitor real user connections. Monitoring can be performed by the built-in performance counters or by the network equipment.

However, in many cases, you may estimate network utilization by understanding how Remote Desktop Protocol works and by analyzing your users' work patterns.

The remote protocol delivers the graphics generated by the remote server to display it on a local monitor. More specifically, it provides the desktop bitmap entirely composed on the server. While sending a desktop bitmap seems like a simple task at first approach, it requires a significant amount of resources. For example, a 1080p desktop image in its uncompressed form is about 8Mb in size. Displaying this image on the locally connected monitor with a modest screen refresh rate of 30 Hz requires bandwidth of about 237 MB/s.

To reduce the amount of data transferred over the network, RDP uses the combination of multiple techniques, including but not limited to

- Frame rate optimizations
- Screen content classification
- Content-specific codecs
- Progressive image encoding

- Client-side caching

To better understand remote graphics, consider the following:

- The richer the graphics, more bandwidth it will take
 - Text, window UI elements, and solid color areas are consuming less bandwidth than anything else.
 - Natural images are the most significant contributors to bandwidth use. But client-side caching helps with its reduction.
- Only changed parts of the screen are transmitted. If there are no visible updates on the screen, no updates are sent.
- Video playback and other high-frame-rate content are essentially an image slideshow. RDP dynamically uses appropriate video codecs to deliver them with the close to original frame rate. However, it's still graphics, and it's still the most significant contributor to bandwidth utilization.
- Idle time in remote desktop means no or minimal screen updates; so, network use is minimal during idle times.
- When remote desktop client window is minimized, no graphical updates are sent from the session host.

Keep in mind that the stress put on your network depends on both your app workload's output frame rate and your display resolution. If either the frame rate or display resolution increases, the bandwidth requirement will also rise. For example, a light workload with a high-resolution display requires more available bandwidth than a light workload with regular or low resolution. Different display resolutions require different available bandwidths.

The table below guides estimating of the data used by the different graphic scenarios. These numbers apply to a single monitor configuration with 1920x1080 resolution and with both default graphics mode and H.264/AVC 444 graphics mode.

Scenario	Default mode	H.264/AVC 444 mode	Thumbnail	Description of the scenario
Idle	0.3 Kbps	0.3 Kbps		User is paused their work and there's no active screen updates
Microsoft Word	100-150 Kbps	200-300 Kbps		User is actively working with Microsoft Word, typing, pasting graphics and switching between documents
Microsoft Excel	150-200 Kbps	400-500 Kbps		User is actively working with Microsoft Excel, multiple cells with formulas and charts are updated simultaneously

Scenario	Default mode	H.264/AVC 444 mode	Thumbnail	Description of the scenario
Microsoft PowerPoint	4-4.5 Mbps	1.6-1.8 Mbps		User is actively working with Microsoft PowerPoint, typing, pasting. User also modifying rich graphics, and using slide transition effects
Web Browsing	6-6.5 Mbps	0.9-1 Mbps		User is actively working with a graphically rich website that contains multiple static and animated images. User scrolls the pages both horizontally and vertically
Image Gallery	3.3-3.6 Mbps	0.7-0.8 Mbps		User is actively working with the image gallery application. browsing, zooming, resizing and rotating images
Video playback	8.5-9.5 Mbps	2.5-2.8 Mbps		User is watching a 30 FPS video that consumes 1/2 of the screen
Fullscreen Video playback	7.5-8.5 Mbps	2.5-3.1 Mbps		User is watching a 30 FPS video that maximized to a fullscreen

Dynamic bandwidth allocation

Remote Desktop Protocol is a modern protocol designed to adjust to the changing network conditions dynamically. Instead of using the hard limits on bandwidth utilization, RDP uses continuous network detection that actively monitors available network bandwidth and packet round-trip time. Based on the findings, RDP dynamically selects the graphic encoding options and allocates bandwidth for device redirection and other virtual channels.

This technology allows RDP to use the full network pipe when available and rapidly back off when the network is needed for something else. RDP detects that and adjusts image quality, frame rate, or compression algorithms if other applications request the network.

Limit network bandwidth use with throttle rate

In most scenarios, there's no need to limit bandwidth utilization as limiting may affect user experience. Yet in the constrained networks you may want to limit network utilization. Another example is leased networks that are charged for the amount of traffic used.

In such cases, you could limit an RDP outbound network traffic by specifying a throttle rate in QoS Policy.

 **Note**

Make sure that RDP Shortpath for managed networks is enabled - throttle rate-limiting are not supported for reverse connect transport.

Implement throttle rate limiting on session host using Group Policy

You can use policy-based Quality of Service (QoS) within Group Policy to set the predefined throttle rate.

To create a QoS policy for domain-joined session hosts, first, sign in to a computer on which Group Policy Management has been installed. Open Group Policy Management (select Start, point to Administrative Tools, and then select Group Policy Management), and then complete the following steps:

1. In Group Policy Management, locate the container where the new policy should be created. For example, if all your session hosts computers are located in an OU named **Session Hosts**, the new policy should be created in the Session Hosts OU.
2. Right-click the appropriate container, and then select **Create a GPO in this domain, and Link it here**.
3. In the **New GPO** dialog box, type a name for the new Group Policy object in the **Name** box, and then select **OK**.
4. Right-click the newly created policy, and then select **Edit**.
5. In the Group Policy Management Editor, expand **Computer Configuration**, expand **Windows Settings**, right-click **Policy-based QoS**, and then select **Create new policy**.
6. In the **Policy-based QoS** dialog box, on the opening page, type a name for the new policy in the **Name** box. Select **Specify Outbound Throttle Rate** and set the required value, and then select **Next**.
7. On the next page, select **Only applications with this executable name** and enter the name **svchost.exe**, and then select **Next**. This setting instructs the policy to only prioritize matching traffic from the Remote Desktop Service.

8. On the third page, make sure that both **Any source IP address** and **Any destination IP address** are selected. Select **Next**. These two settings ensure that packets will be managed regardless of which computer (IP address) sent the packets and which computer (IP address) will receive the packets.
9. On page four, select **UDP** from the **Select the protocol this QoS policy applies to** drop-down list.
10. Under the heading **Specify the source port number**, select **From this source port or range**. In the accompanying text box, type **3390**. Select **Finish**.

The new policies you've created won't take effect until Group Policy has been refreshed on your session host computers. Although Group Policy periodically refreshes on its own, you can force an immediate refresh by following these steps:

1. On each session host for which you want to refresh Group Policy, open a Command Prompt as administrator (*Run as administrator*).
2. At the command prompt, enter

```
Console  
gpupdate /force
```

Implement throttle rate limiting on session host using PowerShell

You can set throttle rate for RDP Shortpath for managed networks using the PowerShell cmdlet below:

```
PowerShell  
New-NetQoSPolicy -Name "RDP Shortpath for managed networks" -  
AppPathNameMatchCondition "svchost.exe" -IPProtocolMatchCondition UDP -  
IPSrcPortStartMatchCondition 3390 -IPSrcPortEndMatchCondition 3390 -  
ThrottleRateActionBitsPerSecond 10mb -NetworkProfile All
```

Next steps

- To learn about bandwidth requirements for Azure Virtual Desktop, see [Understanding Remote Desktop Protocol \(RDP\) Bandwidth Requirements for Azure Virtual Desktop](#).

- To learn about Azure Virtual Desktop network connectivity, see [Understanding Azure Virtual Desktop network connectivity](#).
- To get started with Quality of Service (QoS) for Azure Virtual Desktop, see [Implement Quality of Service \(QoS\) for Azure Virtual Desktop](#).

Proxy server guidelines for Azure Virtual Desktop

Article • 06/29/2023

This article will show you how to use a proxy server with Azure Virtual Desktop. The recommendations in this article only apply to connections between Azure Virtual Desktop infrastructure, client, and session host agents. This article doesn't cover network connectivity for Office, Windows 10, FSLogix, or other Microsoft applications.

What are proxy servers?

We recommend bypassing proxies for Azure Virtual Desktop traffic. Proxies don't make Azure Virtual Desktop more secure because the traffic is already encrypted. To learn more about connection security, see [Connection security](#).

Most proxy servers aren't designed for supporting long running WebSocket connections and may affect connection stability. Proxy server scalability also causes issues because Azure Virtual Desktop uses multiple long-term connections. If you do use proxy servers, they must be the right size to run these connections.

If the proxy server's geography is far from the host, then this distance will cause more latency in your user connections. More latency means slower connection time and worse user experience, especially in scenarios that need graphics, audio, or low-latency interactions with input devices. If you must use a proxy server, keep in mind that you need to place the server in the same geography as the Azure Virtual Desktop Agent and client.

If you configure your proxy server as the only path for Azure Virtual Desktop traffic to take, the Remote Desktop Protocol (RDP) data will be forced over Transmission Control Protocol (TCP) instead of User Datagram Protocol (UDP). This move lowers the visual quality and responsiveness of the remote connection.

In summary, we don't recommend using proxy servers on Azure Virtual Desktop because they cause performance-related issues from latency degradation and packet loss.

Bypassing a proxy server

If your organization's network and security policies require proxy servers for web traffic, you can configure your environment to bypass Azure Virtual Desktop connections while still routing the traffic through the proxy server. However, each organization's policies

are unique, so some methods may work better for your deployment than others. Here are some configuration methods you can try to prevent performance and reliability loss in your environment:

- Azure service tags with Azure Firewall
- Proxy server bypass using Proxy Auto Configuration (.PAC) files
- Bypass list in the local proxy configuration
- Using proxy servers for per-user configuration
- Using RDP Shortpath for the RDP connection while keeping the service traffic over the proxy

Recommendations for using proxy servers

Some organizations require that all user traffic goes through a proxy server for tracking or packet inspection. This section describes how we recommend configuring your environment in these cases.

Use proxy servers in the same Azure geography

When you use a proxy server, it handles all communication with the Azure Virtual Desktop infrastructure and performs DNS resolution and Anycast routing to the nearest Azure Front Door. If your proxy servers are distant or distributed across an Azure geography, your geographical resolution will be less accurate. Less accurate geographical resolution means connections will be routed to a more distant Azure Virtual Desktop cluster. To avoid this issue, only use proxy servers that are geographically close to your Azure Virtual Desktop cluster.

Use RDP Shortpath for managed networks for desktop connectivity

When you enable RDP Shortpath for managed networks, RDP data will bypass the proxy server, if possible. Bypassing the proxy server ensures optimal routing while using the UDP transport. Other Azure Virtual Desktop traffic, such as brokering, orchestration, and diagnostics will still go through the proxy server.

Don't use SSL termination on the proxy server

Secure Sockets Layer (SSL) termination replaces security certificates of the Azure Virtual Desktop components with certificates generated by proxy server. This proxy server feature enables packet inspection for HTTPS traffic on the proxy server. However, packet

inspection also increases the service response time, making it take longer for users to sign in. For reverse-connect scenarios, RDP traffic packet inspection isn't necessary because reverse-connect RDP traffic is binary and uses extra levels of encryption.

If you configure your proxy server to use SSL inspection, remember that you can't revert your server to its original state after the SSL inspection makes changes. If something in your Azure Virtual Desktop environment stops working while you have SSL inspection enabled, you must disable SSL inspection and try again before you open a support case. SSL inspection can also cause the Azure Virtual Desktop agent to stop working because it interferes with trusted connections between the agent and the service.

Don't use proxy servers that need authentication

Azure Virtual Desktop components on the session host run in the context of their operating system, so they don't support proxy servers that require authentication. If the proxy server requires authentication, the connection will fail.

Plan for the proxy server network capacity

Proxy servers have capacity limits. Unlike regular HTTP traffic, RDP traffic has long running, chatty connections that are bi-directional and consume lots of bandwidth. Before you set up a proxy server, talk to your proxy server vendor about how much throughput your server has. Also make sure to ask them how many proxy sessions you can run at one time. After you deploy the proxy server, carefully monitor its resource use for bottlenecks in Azure Virtual Desktop traffic.

Proxy servers and Microsoft Teams media optimization

Azure Virtual Desktop doesn't support proxy servers with [media optimization for Microsoft Teams](#).

Session host configuration recommendations

To configure a session host level proxy server, you need to enable a systemwide proxy. Remember that systemwide configuration affects all OS components and applications running on the session host. The following sections are recommendations for configuring systemwide proxies.

Use the Web Proxy Auto-Discovery (WPAD) protocol

The Azure Virtual Desktop agent automatically tries to locate a proxy server on the network using the Web Proxy Auto-Discovery (WPAD) protocol. During a location attempt, the agent searches the domain name server (DNS) for a file named `wpad.domainsuffix`. If the agent finds the file in the DNS, it makes an HTTP request for a file named `wpad.dat`. The response becomes the proxy configuration script that chooses the outbound proxy server.

To configure your network to use DNS resolution for WPAD, follow the instructions in [Auto detect settings Internet Explorer 11](#). Make sure the DNS server global query blocklist allows the WPAD resolution by following the directions in [Set-DnsServerGlobalQueryBlockList](#).

Manually set a device-wide proxy for Windows services

If you're specifying a proxy server manually, at a minimum you will need to set a proxy for the Windows services *RDAgent* and *Remote Desktop Services* on your session hosts. *RDAgent* runs with the account *Local System* and *Remote Desktop Services* runs with the account *Network Service*. You can set a proxy for these accounts using the `bitsadmin` command-line tool.

The following example configures the Local System and Network Service accounts to use a proxy `.pac` file. You'll need to run these commands from an elevated command prompt, changing the placeholder value for `<server>` with your own address:

Windows Command Prompt

```
bitsadmin /util /setieproxy LOCALSYSTEM AUTOSCRIPT http://<server>/proxy.pac
bitsadmin /util /setieproxy NETWORKSERVICE AUTOSCRIPT
http://<server>/proxy.pac
```

For a full reference and other examples, see [bitsadmin util and setieproxy](#).

You can also set a device-wide proxy or Proxy Auto Configuration (.PAC) file that applies to all interactive, Local System, and Network Service users. If your session hosts are enrolled with Intune, you can set a proxy with the [Network Proxy CSP](#), however, Windows multi-session client operating systems don't support Policy CSP as they only support the [settings catalog](#). Alternatively you can configure a device-wide proxy using the `netsh winhttp` command. For a full reference and examples, see [Netsh Commands for Windows Hypertext Transfer Protocol \(WINHTTP\)](#)

Client-side proxy support

The Azure Virtual Desktop client supports proxy servers configured with system settings or a [Network Proxy CSP](#).

Azure Virtual Desktop client support

The following table shows which Azure Virtual Desktop clients support proxy servers:

Client name	Proxy server support
Windows Desktop	Yes
Web client	Yes
Android	No
iOS	Yes
macOS	Yes
Windows Store	Yes

For more information about proxy support on Linux based thin clients, see [Thin client support](#).

Support limitations

There are many third-party services and applications that act as a proxy server. These third-party services include distributed next-gen firewalls, web security systems, and basic proxy servers. We can't guarantee that every configuration is compatible with Azure Virtual Desktop. Microsoft only provides limited support for connections established over a proxy server. If you're experiencing connectivity issues while using a proxy server, Microsoft support recommends you configure a proxy bypass and then try to reproduce the issue.

Next steps

For more information about keeping your Azure Virtual Desktop deployment secure, check out our [security guide](#).

Analyze connection quality in Azure Virtual Desktop

Article • 06/12/2023

Important

The Connection Graphics Data Logs are currently in preview. See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Azure Virtual Desktop helps users host client sessions on their session hosts running on Azure. When a user starts a session, they connect from their local device over a network to access the session host. It's important that the user experience feels as much like a local session on a physical device as possible. To understand the network connectivity from a user's device to a session host, see [Understanding Azure Virtual Desktop network connectivity](#).

You can analyze connection quality in your Azure Virtual Desktop deployment by using Azure Log Analytics. In this article, we'll talk about how you can measure your connection network and connection graphics to improve the connection quality of your end-users.

Connection network and graphics data

The connection network and graphics data that [Azure Log Analytics](#) collects can help you discover areas that impact your end-user's graphical experience. The service collects data for reports regularly throughout the session. You can also use [RemoteFX network performance counters](#) to get some graphics-related performance data from your deployment, but they're not quite as comprehensive as Azure Log Analytics. Azure Virtual Desktop connection network data reports have the following advantages over RemoteFX network performance counters:

- Each record is connection-specific and includes the correlation ID of the connection that can be tied back to the user.
- The round trip time measured in this table is protocol-agnostic and will record the measured latency for Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) connections.

Connection network data

The network data you collect for your data tables using the *NetworkData* table includes the following information:

- The **estimated available bandwidth (kilobytes per second)** is the average estimated available network bandwidth during each connection time interval.
- The **estimated round trip time (milliseconds)** is the average estimated round trip time during each connection time interval. Round trip time is how long a network request takes to go from the end-user's device to the session host through the network, then return from the session host to the end-user device.
- The **Correlation ID** is the ActivityId of a specific Azure Virtual Desktop connection that's assigned to every diagnostic within that connection.
- The **time generated** is a timestamp in Coordinated Universal Time (UTC) time that marks when an event the data counter is tracking happened on the virtual machine (VM). All averages are measured by the time window that ends at the marked timestamp.
- The **Resource ID** is a unique ID assigned to the Azure Virtual Desktop host pool associated with the data the diagnostics service collects for this table.
- The **source system, Subscription ID, Tenant ID, and type** (table name).

Frequency

The service generates these network data points every two minutes during an active session.

Connection graphics data (preview)

You should consult the *ConnectionGraphicsData* table (preview) when users report slow or choppy experiences in their Azure Virtual Desktop sessions. The *ConnectionGraphicsData* table will give you useful information whenever graphical indicators, end-to-end delay, and dropped frames percentage fall below the "healthy" threshold for Azure Virtual Desktop. This table will help your admins track and understand factors across the server, client, and network that could be contributing to the user's slow or choppy experience. However, while the *ConnectionGraphicsData* table is a useful tool for troubleshooting poor user experience, since it's not regularly populated throughout a session, it isn't a reliable environment baseline.

The Graphics table only captures performance data from the Azure Virtual Desktop graphics stream. This table doesn't capture performance degradation or "slowness" caused by application-specific factors or the virtual machine (CPU or storage constraints). You should use this table with other VM performance metrics to determine if the delay is caused by the remote desktop service (graphics and network) or something inherent in the VM or app itself.

The graphics data you collect for your data tables includes the following information:

- The **Last evaluated connection time interval** is the two minutes leading up to the time graphics indicators fell below the quality threshold.
- The **end-to-end delay (milliseconds)** is the delay in the time between when a frame is captured on the server until the time frame is rendered on the client, measured as the sum of the encoding delay on the server, network delay, the decoding delay on the client, and the rendering time on the client. The delay reflected is the highest (worst) delay recorded in the last evaluated connection time interval.
- The **compressed frame size (bytes)** is the compressed size of the frame with the highest end-to-end delay in the last evaluated connection time interval.
- The **encoding delay on the server (milliseconds)** is the time it takes to encode the frame with the highest end-to-end delay in the last evaluated connection time interval on the server.
- The **decoding delay on the client (milliseconds)** is the time it takes to decode the frame with the highest end-to-end delay in the last evaluated connection time interval on the client.
- The **rendering delay on the client (milliseconds)** is the time it takes to render the frame with the highest end-to-end delay in the last evaluated connection time interval on the client.
- The **percentage of frames skipped** is the total percentage of frames dropped by these three sources:
 - The client (slow client decoding).
 - The network (insufficient network bandwidth).
 - The server (the server is busy).

The recorded values (one each for client, server, and network) are from the second with the highest dropped frames in the last evaluated connection time interval.

- The **estimated available bandwidth (kilobytes per second)** is the average estimated available network bandwidth during the second with the highest end-to-end delay in the time interval.
- The **estimated round trip time (milliseconds)**, which is the average estimated round trip time during the second with the highest end-to-end delay in the time interval. Round trip time is how long a network request takes to go from the end-user's device to the session host through the network, then return from the session host to the end-user device.
- The **Correlation ID**, which is the ActivityId of a specific Azure Virtual Desktop connection that's assigned to every diagnostic within that connection.
- The **time generated**, which is a timestamp in UTC time that marks when an event the data counter is tracking happened on the virtual machine (VM). All averages are measured by the time window that ends that the marked timestamp.
- The **Resource ID** is a unique ID assigned to the Azure Virtual Desktop host pool associated with the data the diagnostics service collects for this table.
- The **source system, Subscription ID, Tenant ID, and type** (table name).

Frequency

In contrast to other diagnostics tables that report data at regular intervals throughout a session, the frequency of data collection for the graphics data varies depending on the graphical health of a connection. The table won't record data for "Good" scenarios, but will recording if any of the following metrics are recorded as "Poor" or "Okay," and the resulting data will be sent to your storage account. Data only records once every two minutes, maximum. The metrics involved in data collection are listed in the following table:

Metric	Bad	Okay	Good
Percentage of dropped frames with low frame rate (less than 15 fps)	Greater than 15%	10%–15%	less than 10%
Percentage of dropped frames with high frame rage (greater than 15 fps)	Greater than 50%	20%–50%	Less than 20%
End-to-end delay per frame	Greater than 300 ms	150 ms–300 ms	Less than 150 ms

① Note

For end-to-end delay per frame, if any frame in a single second is delayed by over 300 ms, the service registers it as "Bad". If all frames in a single second take between 150 ms and 300 ms, the service marks it as "Okay."

Next steps

- Learn more about how to monitor and run queries about connection quality issues at [Monitor connection quality](#).
- Troubleshoot connection and latency issues at [Troubleshoot connection quality for Azure Virtual Desktop](#).
- To check the best location for optimal latency, see the [Azure Virtual Desktop Experience Estimator tool](#).
- For pricing plans, see [Azure Log Analytics pricing](#).
- To get started with your Azure Virtual Desktop deployment, check out [our tutorial](#).
- To learn about bandwidth requirements for Azure Virtual Desktop, see [Understanding Remote Desktop Protocol \(RDP\) Bandwidth Requirements for Azure Virtual Desktop](#).
- To learn about Azure Virtual Desktop network connectivity, see [Understanding Azure Virtual Desktop network connectivity](#).
- Learn how to use Azure Virtual Desktop Insights at [Get started with Azure Virtual Desktop Insights](#).

Multiregion Business Continuity and Disaster Recovery (BCDR) for Azure Virtual Desktop

Azure Virtual Desktop

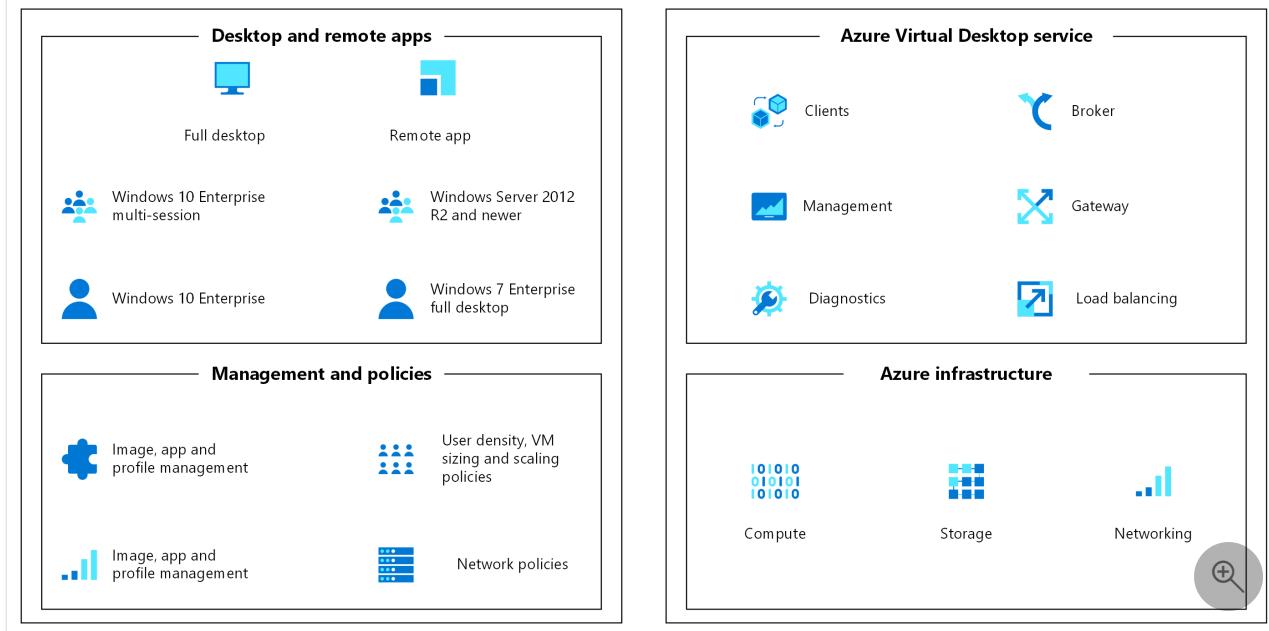
Azure Virtual Desktop is a comprehensive desktop and app virtualization service running on Microsoft Azure. Virtual Desktop helps enable a secure remote desktop experience that helps organizations strengthen business resilience. It delivers simplified management, Windows 10 and 11 Enterprise multi-session, and optimizations for Microsoft 365 Apps for enterprise. With Virtual Desktop, you can deploy and scale your Windows desktops and apps on Azure in minutes, providing integrated security and compliance features to help keep your apps and data secure.

As you continue to enable remote work for your organization with Virtual Desktop, it's important to understand its disaster recovery capabilities and best practices. These practices strengthen reliability across regions to help keep data safe and employees productive. This article provides you with considerations on business continuity and disaster recovery (BCDR) prerequisites, deployment steps, and best practices. You'll learn about options, strategies, and architecture guidance. The content in this document enables you to prepare a successful BCDR plan and can help you bring more resilience to your business during planned and unplanned downtime events.

There are several types of disasters and outages, and each can have a different impact. Resiliency and recovery are discussed in depth for both local and region-wide events, including recovery of the service in a different remote Azure region. This type of recovery is called geo disaster recovery. It's critical to build your Virtual Desktop architecture for resiliency and availability. You should provide maximum local resiliency to reduce the impact of failure events. This resiliency also reduces the requirements to execute recovery procedures. This article also provides information about high-availability and best practices.

Virtual Desktop control plane

Virtual Desktop offers BCDR for its control plane to preserve customer metadata during outages. When an outage occurs in a region, the service infrastructure components fail over to the secondary location and continue functioning as normal. You can still access service-related metadata, and users can still connect to available hosts. End-user connections stay online if the tenant environment or hosts remain accessible. [Data locations for Azure Virtual Desktop](#) are different from the location of the host pool session host virtual machines (VMs) deployment. It's possible to locate Virtual Desktop metadata in one of the supported regions, and then deploy VMs in a different location. No other action is required.

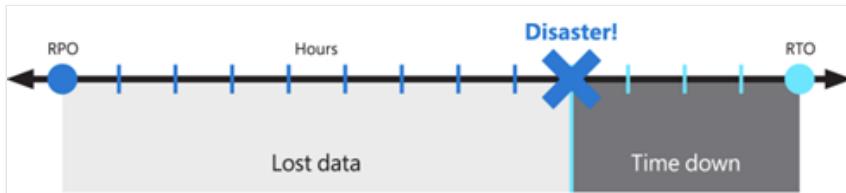


Goals and scope

The goals of this guide are to:

- Ensure maximum availability, resiliency, and geo-disaster recovery capability while minimizing data loss for important selected user data.
- Minimize recovery time.

These objectives are also known as the recovery point objective (RPO) and the Recovery Time Objective (RTO).



The proposed solution provides local high-availability, protection from a single availability zone failure, and protection from an entire Azure region failure. It relies on a redundant deployment in a different, or secondary, Azure region to recover the service. While it's still a good practice, Virtual Desktop and the technology used to build BCDR don't require Azure regions to be [paired](#). Primary and secondary locations can be any Azure region combination, if the network latency permits it.

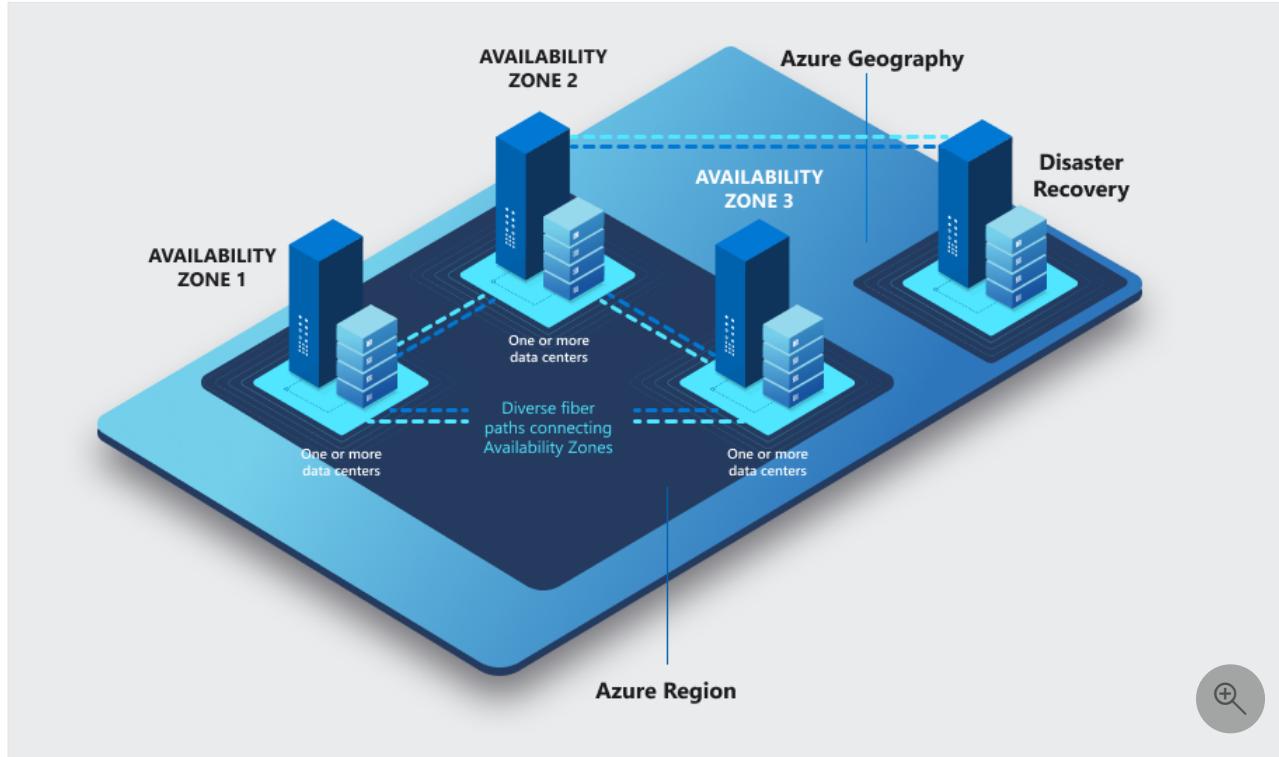
To reduce the impact of a single availability zone failure, use resiliency to improve high availability:

- At the [compute](#) layer, spread the Virtual Desktop session hosts across different availability zones.
- At the [storage](#) layer, use zone resiliency whenever possible.
- At the [networking](#) layer, deploy zone-resilient Azure ExpressRoute and virtual private network (VPN) gateways.
- For each dependency, review the impact of a single zone outage and plan mitigations. For example, deploy Active Directory Domain Controllers and other external resources accessed by Virtual Desktop users across multiple zones.

Depending on the number of availability zones you use, evaluate over-provisioning the number of session hosts to compensate for the loss of one zone. For example, even with (n-1) zones available, you can ensure user experience and performance.

! Note

Azure availability zones are a high-availability feature that can improve resiliency. However, do not consider them a disaster recovery solution able to protect from region-wide disasters.



Because of the possible combinations of types, replication options, service capabilities, and availability restrictions in some regions, the [Cloud Cache](#) component from [FSLogix](#) is used instead of specific storage replication mechanisms.

OneDrive isn't covered in this article. For more information on redundancy and high-availability, see [SharePoint and OneDrive data resiliency in Microsoft 365](#).

For the remainder of this article, you're going to learn about solutions for the two different Virtual Desktop host pool types. There are also observations provided so that you can compare this architecture with other solutions:

- **Personal:** In this type of host pool, a user has a permanently assigned session host, which should never change. Since it's personal, this VM can store user data. The assumption is to use replication and backup techniques to preserve and protect the state.
- **Pooled:** Users are temporarily assigned one of the available session host VMs from the pool, either directly through a desktop application group or by using remote apps. VMs are stateless and user data and profiles are stored in external storage or OneDrive.

Cost implications are discussed, but the primary goal is providing an effective geo disaster recovery deployment with minimal data loss. For more BCDR details, see the following resources:

- [BCDR considerations for Virtual Desktop](#)

- [Virtual Desktop disaster recovery](#)

Prerequisites

Deploy the core infrastructure and make sure it's available in the primary and the secondary Azure region. For guidance on your network topology, you can use the Azure Cloud Adoption Framework [Network topology and connectivity](#) models:

- [Traditional Azure networking topology](#)
- [Virtual WAN network topology \(managed by Microsoft\)](#)

In both models, deploy the primary Virtual Desktop host pool and the secondary disaster recovery environment inside different spoke virtual networks and connect them to each hub in the same region. Place one hub in the primary location, one hub in the secondary location, and then establish connectivity between the two.

The hub eventually provides hybrid connectivity to on-premises resources, firewall services, identity resources like Active Directory Domain Controllers, and management resources like Log Analytics.

You should consider any line-of-business applications and dependent resource availability when failed over to the secondary location.

Active-Active vs. Active-Passive

If distinct sets of users have different BCDR requirements, Microsoft recommends that you use multiple host pools with different configurations. For example, users with a mission critical application might assign a fully redundant host pool with geo disaster recovery capabilities. However, development and test users can use a separate host pool with no disaster recovery at all.

For each single Virtual Desktop host pool, you can base your BCDR strategy on an active-active or active-passive model. In this context, it's assumed that the same set of users in one geographic location is served by a specific host pool.

- **Active-Active**

- For each host pool in the primary region, you deploy a second host pool in the secondary region.
- This configuration provides almost zero RTO, and RPO has an extra cost.
- You don't require an administrator to intervene or fail over. During normal operations, the secondary host pool provides the user with Virtual Desktop resources.
- Each host pool has its own storage account for persistent user profiles.
- You should evaluate latency based on the user's physical location and connectivity available. For some Azure regions, such as Western Europe and Northern Europe, the difference can be negligible when accessing either the primary or secondary regions. You can validate this scenario using the [Azure Virtual Desktop Experience Estimator](#) tool.

- Users are assigned to different application groups, like desktop and remote apps, in both the primary and secondary host pools. In this case, they'll see duplicate entries in their Virtual Desktop client feed. To avoid confusion, use separate Virtual Desktop workspaces with clear names and labels that reflect the purpose of each resource. Inform your users about the usage of these resources.



- If you need storage to manage FSLogix Profile and Office containers, use Cloud Cache to ensure almost zero RPO.
 - To avoid profile conflicts, don't permit users to access both host pools at the same time.
 - Due to the active-active nature of this scenario, you should educate your users on how to use these resources in the proper way.

- **Active-Passive**

- Like active-active, for each host pool in the primary region, you deploy a second host pool in the secondary region.
- The amount of compute resources active in the secondary region is reduced compared to the primary region, depending on the budget available. You can use automatic scaling to provide more compute capacity, but it requires more time, and Azure capacity isn't guaranteed.
- This configuration provides higher RTO when compared to the active-active approach, but it's less expensive.
- You need administrator intervention to execute a failover procedure if there's an Azure outage. The secondary host pool doesn't normally provide the users access to Virtual Desktop resources.
- Each host pool has its own storage accounts for persistent user profiles.
- Users that consume Virtual Desktop services with optimal latency and performance are affected only if there's an Azure outage. You should validate this scenario by using the [Azure Virtual Desktop Experience Estimator](#) tool. Performance should be acceptable, even if degraded, for the secondary disaster recovery environment.
- Users are assigned to only one set of application groups, like Desktop and Remote Apps. During normal operations, these apps are in the primary host pool. During an outage, and after a failover, users are assigned to Application Groups in the secondary host pool. No duplicate entries are shown in the user's Virtual Desktop client feed, they can use the same workspace, and everything is transparent for them.
- If you need storage to manage FSLogix Profile and Office containers, use Cloud Cache to ensure almost zero RPO.

- To avoid profile conflicts, don't permit users to access both host pools at the same time. Since this scenario is active-passive, administrators can enforce this behavior at the application group level. Only after a failover procedure is the user able to access each application group in the secondary host pool. Access is revoked in the primary host pool application group and reassigned to an application group in the secondary host pool.
- Execute a failover for all application groups, otherwise users using different application groups in different host pools might cause profile conflicts if not effectively managed.
- It's possible to allow a specific subset of users to selectively fail over to the secondary host pool and provide limited active-active behavior and test failover capability. It's also possible to fail over specific application groups, but you should educate your users to not use resources from different host pools at the same time.

For specific circumstances, you can create a single host pool with a mix of session hosts located in different regions. The advantage of this solution is that if you have a single host pool, then there's no need to duplicate definitions and assignments for desktop and remote apps. Unfortunately, this solution has several disadvantages.

- For pooled host pools, it isn't possible to force a user to a session host in the same region.
- A user might experience higher latency and suboptimal performance when connecting to a session host in a remote region.
- If you require storage for user profiles, you need a complex configuration to manage assignments for session hosts in the primary and secondary regions.
- You can use drain mode to temporarily disable access to session hosts located in the secondary region. But this method introduces more complexity, management overhead, and inefficient use of resources.
- You can maintain session hosts in an offline state in the secondary regions, but it introduces more complexity and management overhead.

Considerations and recommendations

General

In order to deploy either an active-active or active-passive configuration using multiple host pools and an FSLogix cloud cache mechanism, you can create the host pool inside the same workspace or a different one, depending on the model. This approach requires you to maintain the alignment and updates, keeping both host pools in sync and at the same configuration level. In addition to a new host pool for the secondary disaster recovery region, you need:

- To create new distinct application groups and related applications for the new host pool.
- To revoke user assignments to the primary host pool, and then manually reassign them to the new host pool during the failover.

Review the [Business continuity and disaster recovery options for FSLogix](#).

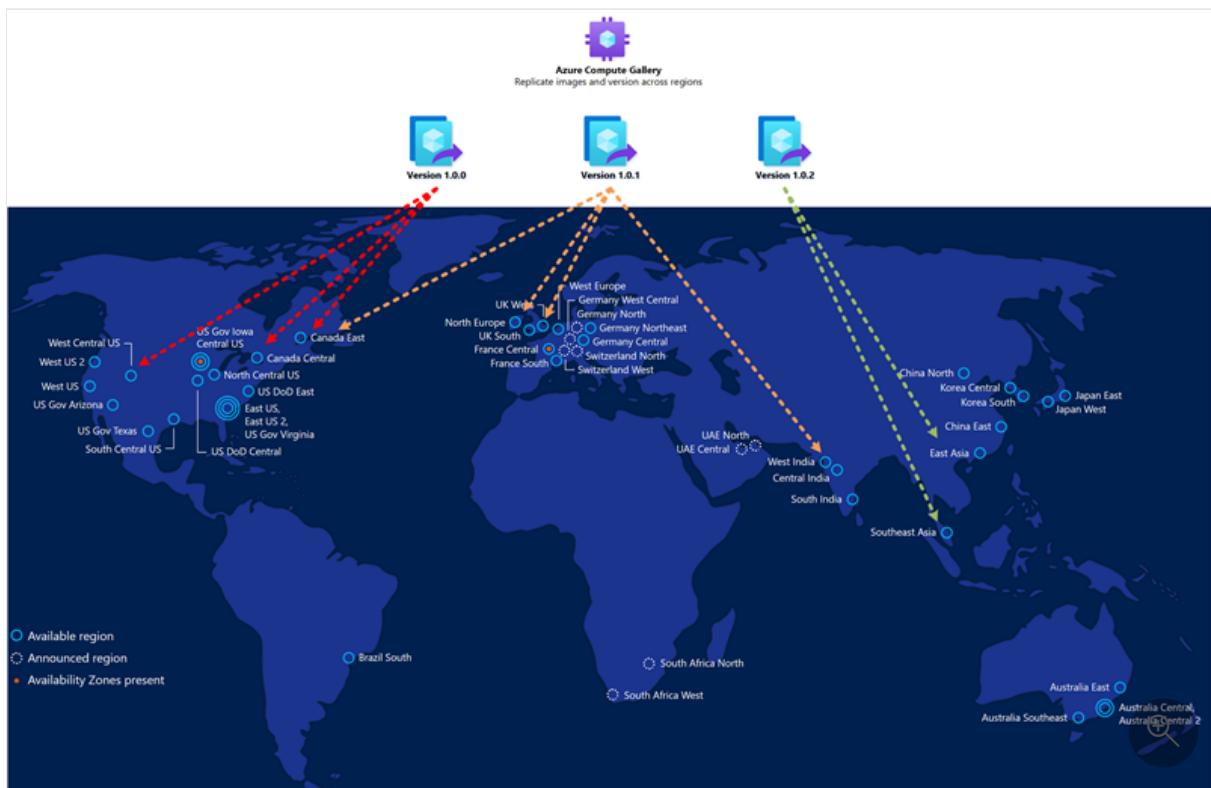
- [No profile recovery](#) is not covered in this document.
- [Cloud cache \(active/passive\)](#) is included in this document but is implemented it using the same host pool.
- [Cloud cache \(active/active\)](#) is covered in the remaining part of this document.

There are some limits for Virtual Desktop resources. For more information, see [Azure Virtual Desktop service limits](#).

For diagnostics and monitoring, use the same Log Analytics workspace for both the primary and secondary host pool.

Compute

- For deployment of both host pools in the primary and secondary disaster recovery regions, you should use Azure availability zones and spread your VM fleet over all available zones. If availability zones aren't available in the local region, you can use an Azure availability set.
- The golden image that you use for host pool deployment in the secondary disaster recovery region should be the same you use for the primary. You should store images in the Azure Compute Gallery and configure multiple image replicas in both the primary and the secondary locations. Each image replica can sustain a parallel deployment of a maximum number of VMs, and you might require more than one based on your desired deployment batch size. For more information, see [Store and share images in an Azure Compute Gallery](#).



- The Azure Compute Gallery is not a global resource, then it is recommended to have at least a secondary gallery in the secondary region. Once you have created in the primary region a Gallery, a VM Image Definition and a VM Image Version, you should create the same three objects also in the secondary region. When creating the VM Image Version, there is the possibility to copy the VM Image Version created in the primary region. To achieve this, from the secondary region, you have to specify the Gallery, the VM Image Definition and VM Image Version used in the primary region, and Azure will copy the image and create a local VM Image Version. It is possible to execute this operation using the Azure portal or AZ CLI command as outlined below:

[Create an image definition and an image version](#)

```
az sig image-version create
```

- Not all the session host VMs in the secondary disaster recovery locations must be active and running all the time. You must initially create a sufficient number of VMs, and after that, use an auto-scale mechanism like [scaling plans](#). With these mechanisms, it's possible to maintain most compute resources in an offline or deallocated state to reduce costs.
- It's also possible to use automation to create session hosts in the secondary region only when needed. This method optimizes costs, but depending on the mechanism you use, might require a longer RTO. This approach won't permit failover tests without a new deployment and won't permit selective failover for specific groups of users.

Important

You must power on each session host VM for a few hours at least one time every 90 days to refresh the Virtual Desktop token needed to connect to the Virtual Desktop control plane. You should also routinely apply security patches and application updates.

- Having session hosts in an offline, or *deallocated*, state in the secondary region won't guarantee that capacity is available if there's a primary region-wide disaster. It also applies if new sessions are deployed on-demand when needed, and with [Site Recovery](#) usage. Compute capacity can be guaranteed if:
 - Session hosts are kept in an active state in the secondary region.
 - You use the new Azure feature [On-demand Capacity Reservation](#).

Note

Azure Reserved Virtual Machine Instances doesn't provide guaranteed capacity, but they can integrate with On-demand Capacity Reservation to reduce the cost.

- Since you use Cloud Cache:
 - You should use the premium tier for the session host VM OS managed disk.
 - You should move the [Cloud Cache](#) to the temporary VM drive and use local SSD storage.

Storage

In this guide, you use at least two separate storage accounts for each Virtual Desktop host pool. One is for the FSLogix Profile container, and one is for the Office container data. You also need one more storage account for [MSIX](#) packages. The following considerations apply:

- You can use [Azure Files](#) share and [Azure NetApp Files](#) as storage alternatives.
- Azure Files share can provide zone resiliency by using the zone-replicated storage resiliency option, if it's available in the region.
- You can't use the geo-redundant storage feature in the following situations:
 - You require a non-paired region. The region pairs for geo-redundant storage are fixed and can't be changed.
 - You're using the premium tier.
- RPO and RTO are higher compared to FSLogix Cloud Cache mechanism.

- It isn't easy to test failover and failback in a production environment.
- Azure NetApp Files requires additional considerations:
 - Zone redundancy is not yet available. If the resiliency requirement is more important than performance, use Azure Files share.
 - Azure NetApp Files can be [zonal](#), that is customers can decide in which (single) Azure Availability Zone to allocate.
 - Cross-Zone replication can be established at the volume level. Replication is async (RPO>0) and requires manual failover (RTO>0). Before using this feature is recommended to review the requirements and considerations from this [article](#).
 - You can now use Azure NetApp Files with zone-redundant VPN and ExpressRoute gateways, if [Standard Networking](#) feature is used, which you might use for networking resiliency. For more information, see [Supported network topologies](#).
 - Azure Virtual WAN is now supported but requires Azure NetApp Files [Standard Networking](#) feature. For more information, see [Supported network topologies](#).
- Azure NetApp Files has a [cross-region replication mechanism](#), the following considerations apply:
 - It's not available in all regions.
 - Region pairs are fixed.
 - Failover isn't transparent, and failback requires storage reconfiguration.
- Limits
 - There are limits in the size, input/output operations per second (IOPS), bandwidth MB/s for both [Azure Files share](#) and [Azure NetApp Files](#) storage accounts and volumes. If necessary, it's possible to use more than one for the same host pool in Virtual Desktop by using [per-group settings](#) in FSLogix. However, this configuration requires more planning and configuration.

The storage account you use for MSIX application packages should be distinct from the other accounts for Profile and Office containers. The following Geo-disaster recovery options are available:

- **One storage account with geo-redundant storage enabled, in the primary region**
 - The secondary region is fixed. This option isn't suitable for local access if there's storage account failover.
- **Two separate storage accounts, one in the primary region and one in the secondary region (recommended)**
 - Use zone-redundant storage for at least the primary region.
 - Each host pool in each region has local storage access to MSIX packages with low latency.
 - Copy MSIX packages twice in both locations and register the packages twice in both host pools. Assign users to the application groups twice.

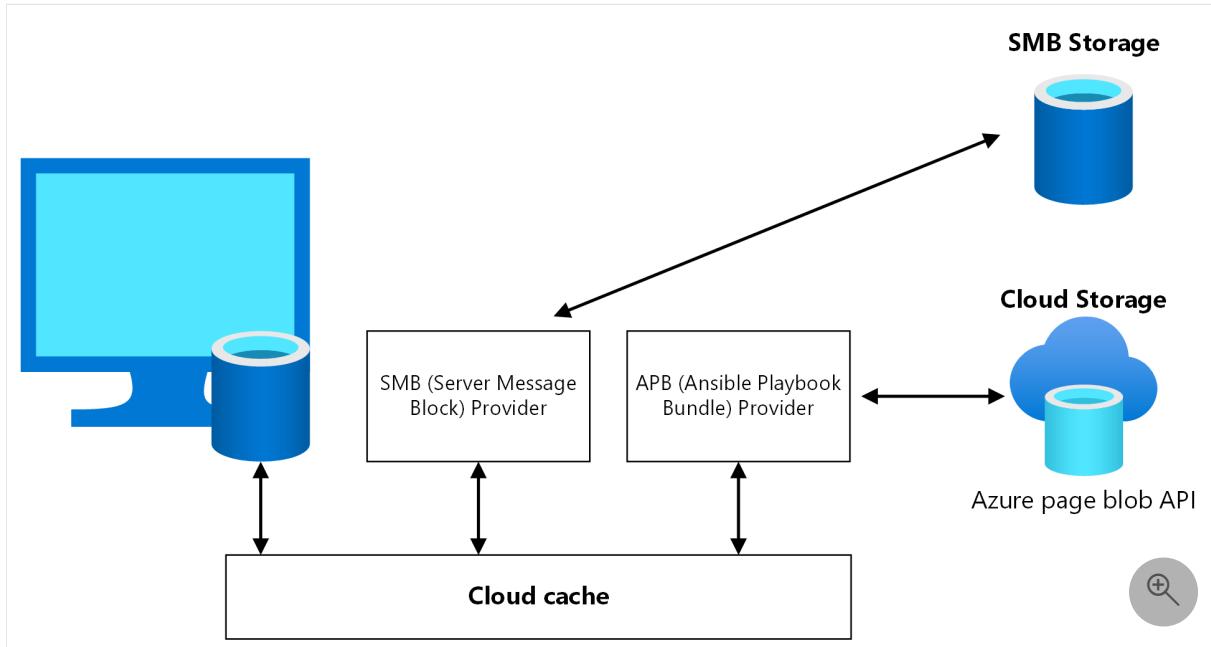
FSLogix

Microsoft recommends that you use the following FSLogix configuration and features:

- If the Profile container content needs to have separate BCDR management, and has different requirements compared to the Office container, you should split them.
 - Office Container only has cached content that can be rebuilt or repopulated from the source if there's a disaster. With Office Container, you might not need to keep backups, which can reduce costs.
 - When using different storage accounts, you can only enable backups on the profile container. Or, you must have different settings like retention period, storage used, frequency, and

RTO/RPO.

- **Cloud Cache** is an FSLogix component in which you can specify multiple profile storage locations and asynchronously replicate profile data, all without relying on any underlying storage replication mechanisms. If the first storage location fails or isn't reachable, Cloud Cache will automatically fail over to use the secondary, and effectively adds a resiliency layer. Use Cloud Cache to replicate both Profile and Office containers between different storage accounts in the primary and secondary regions.



- You must enable Cloud Cache twice in the session host VM registry, once for [Profile Container](#) and once for [Office Container](#). It's possible to not enable Cloud Cache for Office Container, but not enabling it might cause a data misalignment between the primary and the secondary disaster recovery region if there's failover and fallback. Test this scenario carefully before using it in production.
- Cloud Cache is compatible with both [profile split](#) and [per-group](#) settings. per-group requires careful design and planning of active directory groups and membership. You must ensure that every user is part of exactly one group, and that group is used to grant access to host pools.
- The *CCDLocations* parameter that's specified in the registry for the host pool in the secondary disaster recovery region is reverted in order, compared to the settings in the primary region. For more information, see [Tutorial: Configure Cloud Cache to redirect profile containers or office container to multiple Providers](#).

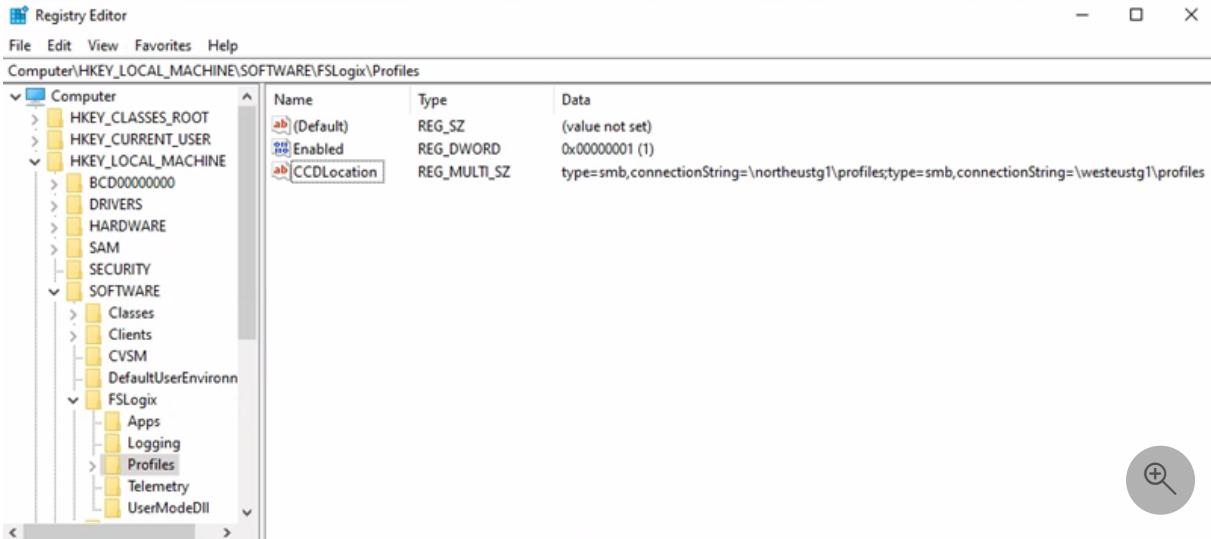
The following example shows a Cloud Cache configuration and related registry keys:

Primary Region = North Europe

- Profile container storage account URI = `\northeustg1\profiles`
 - Registry Key path = `HKEY_LOCAL_MACHINE > SOFTWARE > FSLogix > Profiles`
 - *CCDLocations* value =
`type=smb,connectionString=\northeustg1\profiles;type=smb,connectionString=\westeustg1\profiles`

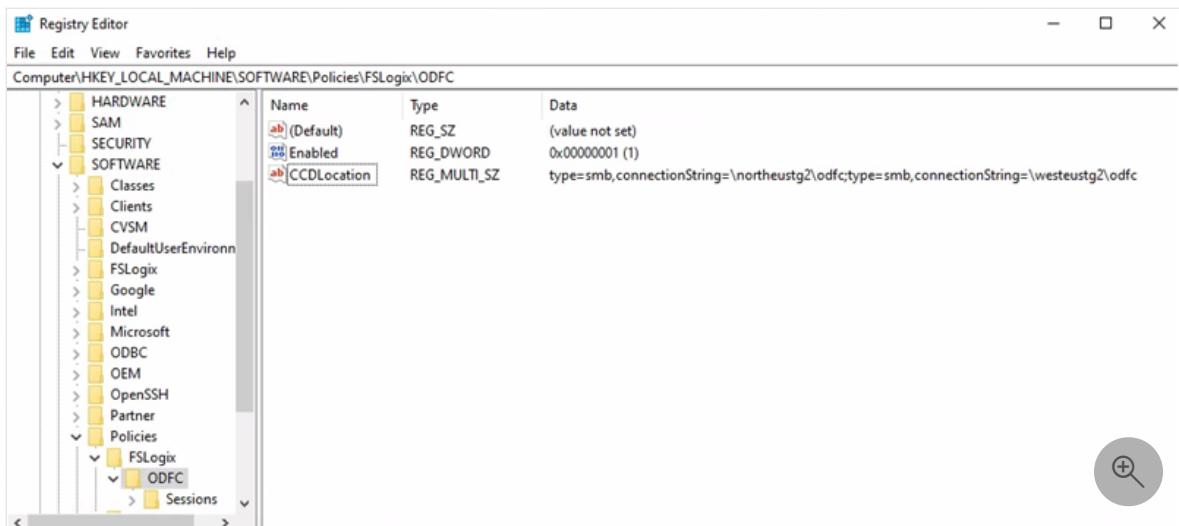
① Note

If you previously downloaded the FSLogix Templates, you can accomplish the same configurations through the Active Directory Group Policy Management Console. For more details on how to set up the Group Policy Object for FSLogix, refer to the guide, [Use FSLogix Group Policy Template Files](#).



- Office container storage account URI = \northeustg2\odcf
 - Registry Key path = HKEY_LOCAL_MACHINE > SOFTWARE > Policy > FSLogix > ODFC
 - CCDLocations* value =

type=smb,connectionString=\northeustg2\odfc;type=smb,connectionString=\westeustg2\odfc



! Note

In the screenshots above, not all the recommended registry keys for FSLogix and Cloud Cache are reported, for brevity and simplicity. For more information, see [FSLogix configuration examples](#).

Secondary Region = West Europe

- Profile container storage account URI = \westeustg1\profiles
 - Registry Key path = HKEY_LOCAL_MACHINE > SOFTWARE > FSLogix > Profiles

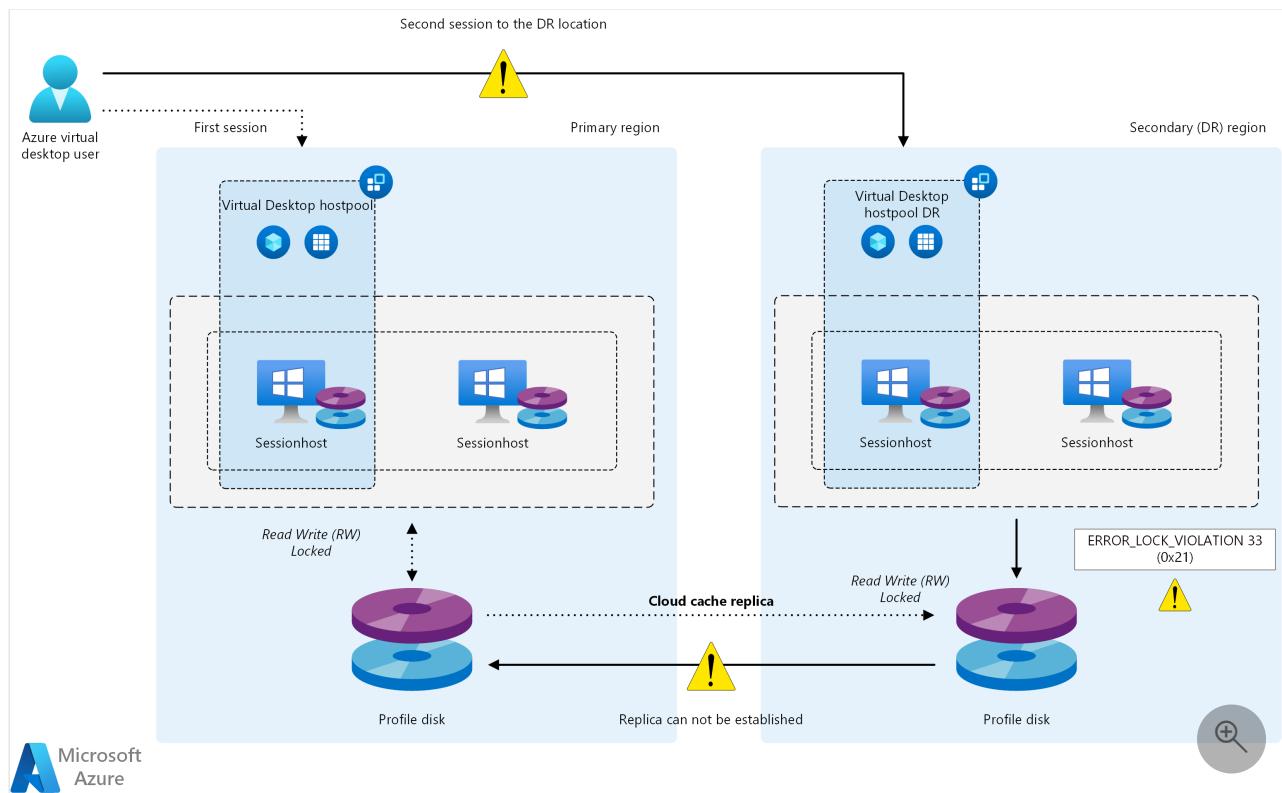
- CCDLocations value =


```
type=smb,connectionString=\westeustg1\profiles;type=smb,connectionString=\northeustg1\profiles
```
- Office container storage account URI = \westeustg2\odcf
 - Registry Key path = HKEY_LOCAL_MACHINE > SOFTWARE > Policy > FSLogix > ODFC
 - CCDLocations value =


```
type=smb,connectionString=\westeustg2\odfc;type=smb,connectionString=\northeustg2\odfc
```

Cloud Cache replication

The Cloud Cache configuration and replication mechanisms guarantee profile data replication between different regions with minimal data loss. Since the same user profile file can be opened in ReadWrite mode by only one process, concurrent access should be avoided, thus users shouldn't open a connection to both host pools at the same time.



Download a [Visio file](#) of this architecture.

Dataflow

1. The Virtual Desktop user launches Virtual Desktop client, and then opens a published Desktop or Remote App application assigned to the primary region host pool.
2. FSLogix retrieves the user Profile and Office containers, and then mounts the underlying storage VHD/X from the storage account located in the primary region.
3. At the same time, the Cloud Cache component initializes replication between the files in the primary region and the files in the secondary region. For this process, Cloud Cache in the primary region acquires an exclusive read-write lock on these files.
4. The same Virtual Desktop user now wants to launch another published application assigned on the secondary region host pool.

5. The FSLogix component running on the Virtual Desktop session host in the secondary region tries to mount the user profile VHD/X files from the local storage account. But the mounting fails since these files are locked by the Cloud Cache component running on the Virtual Desktop session host in the primary region.

6. In the default FSLogix and Cloud Cache configuration, the user can't sign in and an error is tracked in the FSLogix diagnostic logs, *ERROR_LOCK_VIOLATION 33 (0x21)*.

Operational events:			
Type	Id	Date	Description
>Error	41	4/1/2022 20:38:05	Operation: FSLogxLogon_ODFC, SessionId: 2, ErrorCode: 33, Detail: Logon failed. Please check logs and tracelogging and verify that the users disk was detached.
>Error	26	4/1/2022 20:38:05	Error (The process cannot access the file because another process has locked a portion of the file.)
>Error	26	4/1/2022 20:38:05	Frx.Service.ODFContainer.cpp(483): [WCODE: 0x00000021] Base disk failed to acquire an exclusive lock. (The process cannot access the file because another process has locked a portion of the file.)
>Error	41	4/1/2022 20:38:02	Operation: FSLogxLogon_PROFILE, SessionId: 2, ErrorCode: 33, Detail: Logon failed. Please check logs and tracelogging and verify that the users disk was detached.
>Error	26	4/1/2022 20:37:57	LoadProfile failed. Version: 2.9.7979.62170 User: AlexWilber. SID: S-1-12-1-3109892189-1245450925-3666730938-4200816277. SessionId: 2 (The process cannot access the file because another process has locked a portion of the file.)

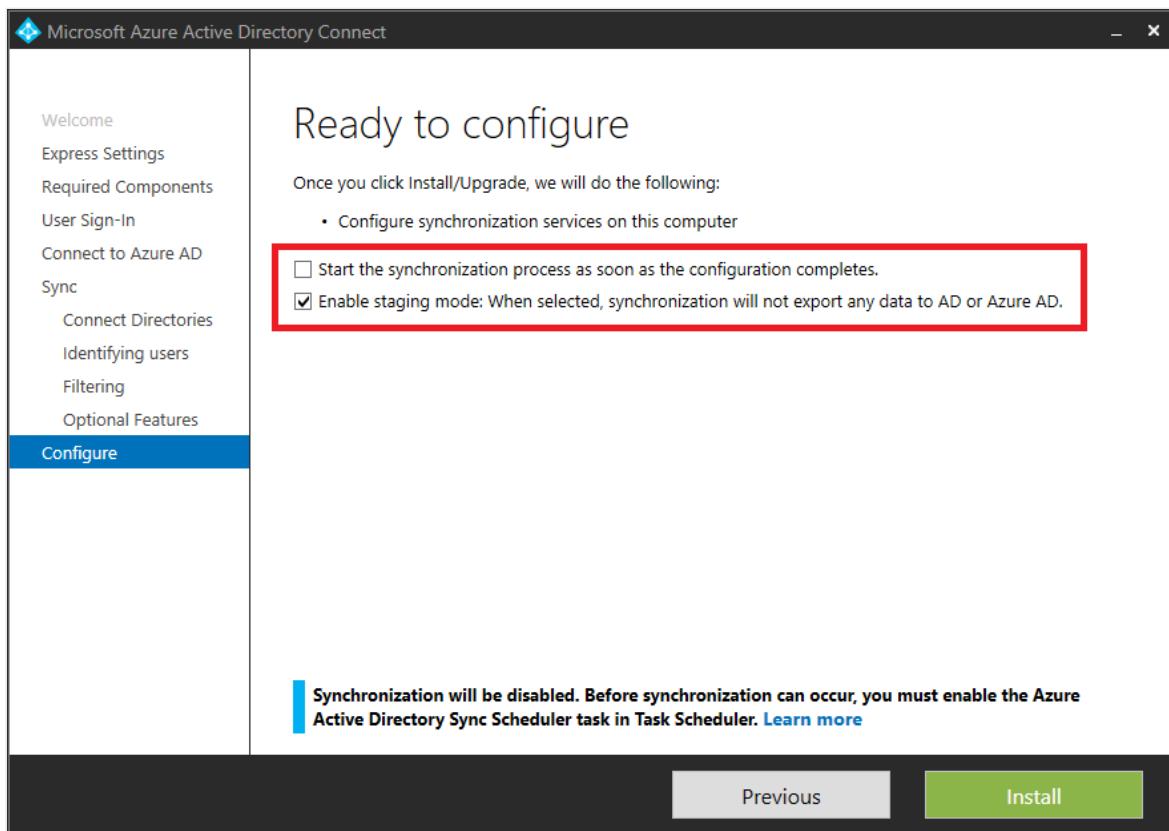


Identity

One of the most important dependencies for Virtual Desktop is the availability of user identity. To access virtual desktops and remote apps from your session hosts, your users need to be able to authenticate.

[Microsoft Entra ID](#) is Microsoft's centralized cloud identity service that enables this capability. Microsoft Entra ID is always used to authenticate users for Azure Virtual Desktop. Session hosts can be joined to the same Microsoft Entra tenant, or to an Active Directory domain using [Active Directory Domain Services](#) or Microsoft Entra Domain Services (Microsoft Entra Domain Services), providing you with a choice of flexible configuration options.

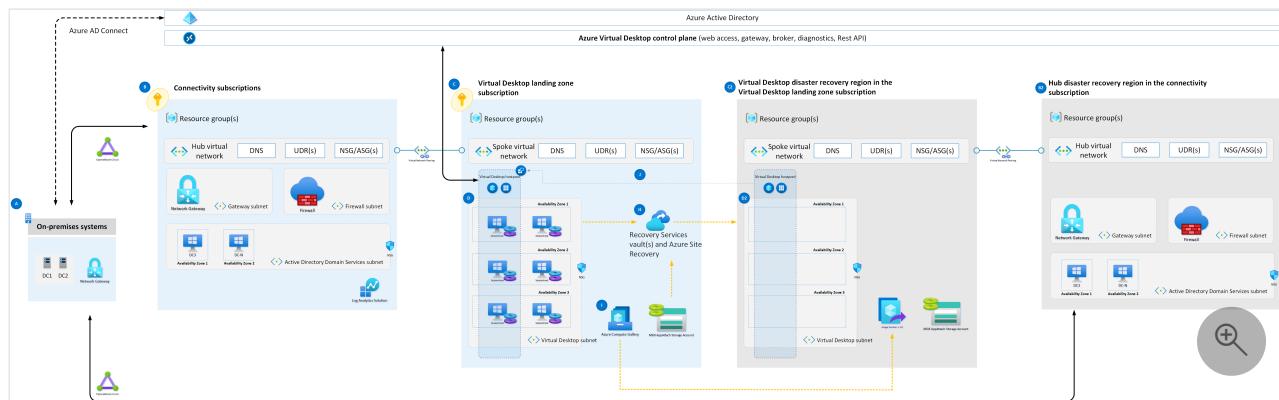
- **Microsoft Entra ID**
 - It's a global multi-region and resilient service with [high-availability](#). No other action is required in this context as part of a Virtual Desktop BCDR plan.
- **Active Directory Domain Services**
 - For Active Directory Domain Services to be resilient and highly available, even if there's a region-wide disaster, you should deploy at least two domain controllers in the primary Azure region. These domain controllers should be in different availability zones if possible, and you should ensure proper replication with the infrastructure in the secondary region and eventually on-premises. You should create at least one more domain controller in the secondary region with global catalog and DNS roles. For more information, see [Deploy AD DS in an Azure virtual network](#).
- **Microsoft Entra Connect**
 - If you're using Microsoft Entra ID with Active Directory Domain Services, and then [Microsoft Entra Connect](#) to synchronize user identity data between Active Directory Domain Services and Microsoft Entra ID, you should consider the resiliency and recovery of this service for protection from a permanent disaster.
 - You can provide high availability and disaster recovery by installing a second instance of the service in the secondary region and enable [staging mode](#).
 - If there's a recovery, the administrator is required to promote the secondary instance by taking it out of staging mode. They must follow the same procedure as placing a server into staging mode. Microsoft Entra Global Administrator credentials are required to perform this configuration.



- Microsoft Entra Domain Services
 - You can use Microsoft Entra Domain Services in some scenarios as an alternative to Active Directory Domain Services.
 - It offers [high-availability](#).
 - If geo-disaster recovery is in scope for your scenario, you should deploy another replica in the secondary Azure region by using a [replica set](#). You can also use this feature to increase high availability in the primary region.

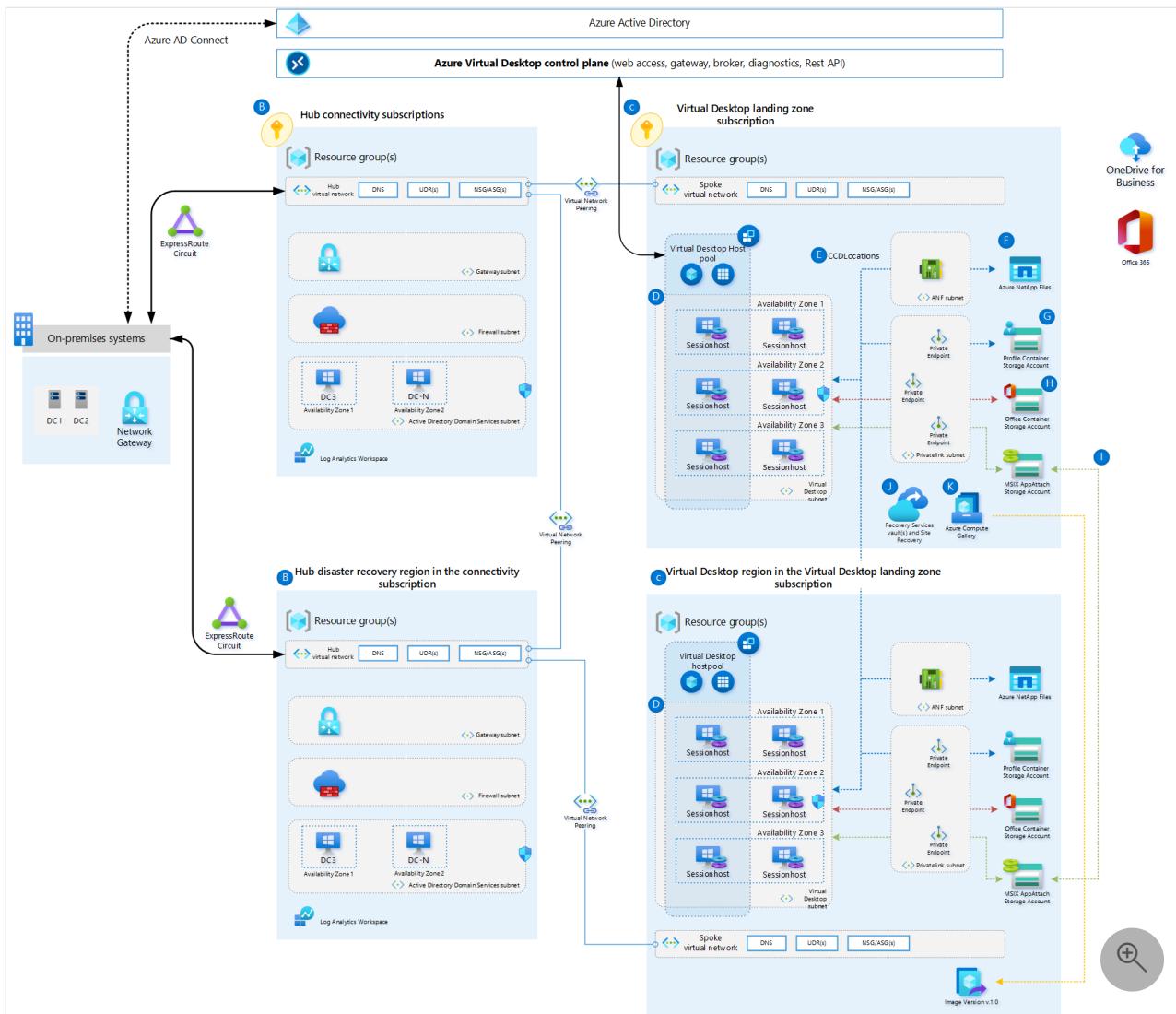
Architecture diagrams

Personal host pool



Download a [Visio file](#) of this architecture.

Pooled host pool



Download a [Visio file](#) of this architecture.

Failover and fallback

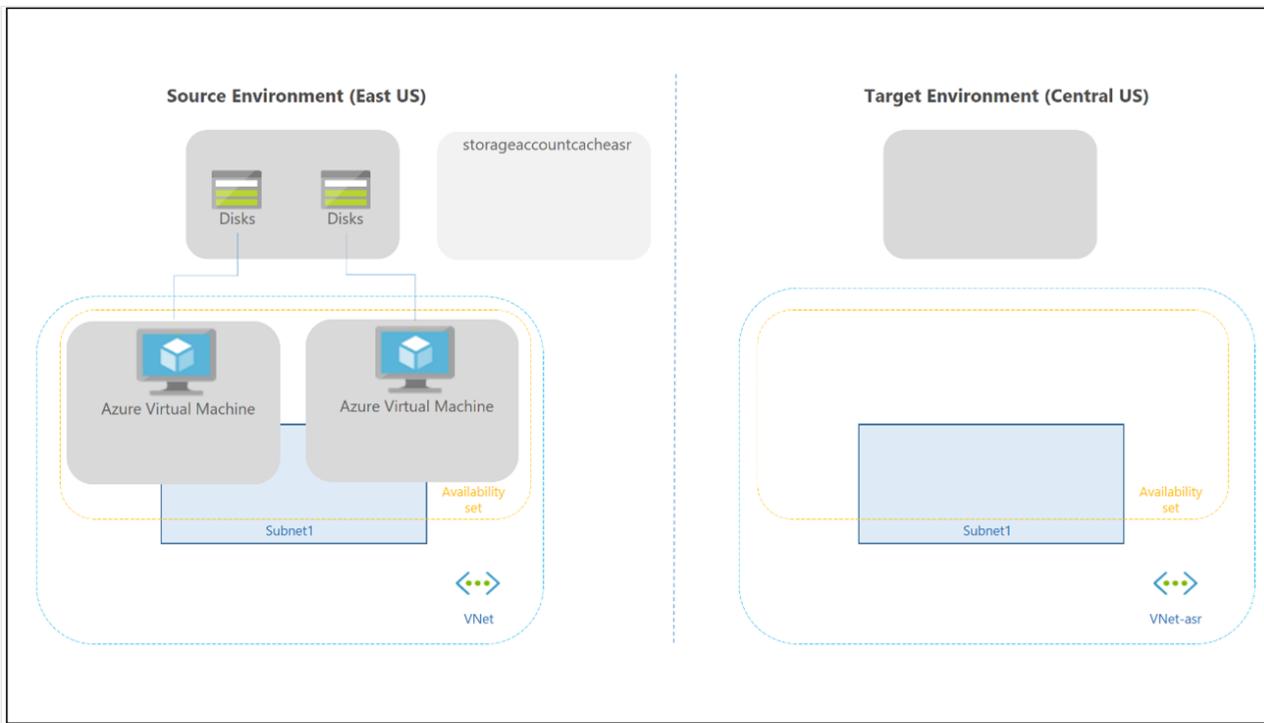
Personal host pool scenario

! Note

Only the active-passive model is covered in this section—an active-active doesn't require any failover or administrator intervention.

Failover and fallback for a personal host pool is different, as there's no Cloud Cache and external storage used for Profile and Office containers. You can still use FSLogix technology to save the data in a container from the session host. There's no secondary host pool in the disaster recovery region, so there's no need to create more workspaces and Virtual Desktop resources to replicate and align. You can use Site Recovery to replicate session host VMs.

You can use Site Recovery in several different scenarios. For Virtual Desktop, use the [Azure to Azure disaster recovery architecture in Azure Site Recovery](#).



The following considerations and recommendations apply:

- Site Recovery failover isn't automatic—an administrator must trigger it by using the Azure portal or [Powershell/API](#).
- You can script and automate the entire Site Recovery configuration and operations by using [PowerShell](#).
- Site Recovery has a declared RTO inside its [Service Level Agreement](#) (SLA). Most of the time, Site Recovery can fail over VMs within minutes.
- You can use Site Recovery with Azure Backup. For more information, see [Support for using Site Recovery with Azure Backup](#).
- You must enable Site Recovery at the VM level, as there's no direct integration in the Virtual Desktop portal experience. You must also trigger failover and fallback at the single VM level.
- Site Recovery provides test failover capability in a separate subnet for general Azure VMs. Don't use this feature for Virtual Desktop VMs, since you would have two identical Virtual Desktop session hosts calling the Virtual Desktop control plane at the same time.
- Site Recovery doesn't maintain Virtual Machine extensions during replication. If you enable any custom extensions for Virtual Desktop session host VMs, you must re-enable the extensions after failover or fallback. The Virtual Desktop built-in extensions **joindomain** and **Microsoft.PowerShell.DSC** are only used when a session host VM is created. It's safe to lose them after a first failover.
- Be sure to review [Support matrix for Azure VM disaster recovery between Azure regions](#) and check requirements, limitations, and the compatibility matrix for the Site Recovery Azure-to-Azure disaster recovery scenario, especially the supported OS versions.
- When you fail over a VM from one region to another, the VM starts up in the target disaster recovery region in an unprotected state. Failback is possible, but the user must [reprotect](#) VMs in the secondary region, and then enable replication back to the primary region.
- Execute periodic testing of failover and failback procedures. Then document an exact list of steps and recovery actions based on your specific Virtual Desktop environment.

(!) Note

Site Recovery is now integrated with [On-Demand Capacity Reservation](#). With this integration, you can use the power of capacity reservations with Site Recovery to reserve compute capacity in the disaster recovery region and guarantee your failovers. When you assign a capacity reservation group for your protected VMs, Site Recovery will fail over the VMs to that group.

Pooled host pool scenario

One of the desired characteristics of an active-active disaster recovery model is that administrator intervention isn't required to recover the service if there's an outage. Failover procedures should only be necessary in an active-passive architecture.

In an active-passive model, the secondary disaster recovery region should be idle, with minimal resources configured and active. Configuration should be kept aligned with the primary region. If there's a failover, reassessments for all users to all desktop and application groups for remote apps in the secondary disaster recovery host pool happen at the same time.

It's possible to have an active-active model and partial failover. If the host pool is only used to provide desktop and application groups, then you can partition the users in multiple non-overlapping Active Directory groups and reassign the group to desktop and application groups in the primary or secondary disaster recovery host pools. A user shouldn't have access to both host pools at the same time. If there's multiple application groups and applications, the user groups you use to assign users might overlap. In this case, it's difficult to implement an active-active strategy. Whenever a user starts a remote app in the primary host pool, the user profile is loaded by FSLogix on a session host VM. Trying to do the same on the secondary host pool might cause a conflict on the underlying profile disk.

Warning

By default, FSLogix [registry settings](#) prohibit concurrent access to the same user profile from multiple sessions. In this BCDR scenario, you shouldn't change this behavior and leave a value of 0 for registry key **ProfileType**.

Here's the initial situation and configuration assumptions:

- The host pools in the primary region and secondary disaster recovery regions are aligned during configuration, including Cloud Cache.
- In the host pools, both DAG1 desktop and APPG2 and APPG3 remote app application groups are offered to users.
- In the host pool in the primary region, Active Directory user groups GRP1, GRP2, and GRP3 are used to assign users to DAG1, APPG2, and APPG3. These groups might have overlapping user memberships, but since the model here uses active-passive with full failover, it's not a problem.

The following steps describe when a failover happens, after either a planned or unplanned disaster recovery.

1. In the primary host pool, remove user assignments by the groups GRP1, GRP2, and GRP3 for application groups DAG1, APPG2, and APPG3.
2. There's a forced disconnection for all connected users from the primary host pool.

3. In the secondary host pool, where the same application groups are configured, you must grant user access to DAG1, APPG2, and APPG3 using groups GRP1, GRP2, and GRP3.
4. Review and adjust the capacity of the host pool in the secondary region. Here, you might want to rely on an auto-scale plan to automatically power on session hosts. You can also manually start the necessary resources.

The **Failback** steps and flow are similar, and you can execute the entire process multiple times. Cloud Cache and configuring the storage accounts ensures that Profile and Office container data is replicated. Before failback, ensure that the host pool configuration and compute resources will be recovered. For the storage part, if there's data loss in the primary region, Cloud Cache will replicate Profile and Office container data from the secondary region storage.

It's also possible to implement a test failover plan with a few configuration changes, without affecting the production environment.

- Create a few new user accounts in Active Directory for production.
- Create a new Active Directory group named **GRP-TEST** and assign users.
- Assign access to DAG1, APPG2, and APPG3 by using the GRP-TEST group.
- Give instructions to users in the GRP-TEST group to test applications.
- Test the failover procedure by using the GRP-TEST group to remove access from the primary host pool and grant access to the secondary disaster recovery pool.

Important recommendations:

- Automate the failover process by using PowerShell, Azure CLI, or another available API or tool.
- Periodically test the entire failover and failback procedure.
- Conduct a regular configuration alignment check to ensure host pools in the primary and secondary disaster region are in sync.

Backup

An assumption in this guide is that there's profile split and data separation between Profile containers and Office containers. FSLogix permits this configuration and the usage of separate storage accounts. Once in separate storage accounts, you can use different backup policies.

- For Office Container, if the content represents only cached data that can be rebuilt from on-line data store like Office 365, it isn't necessary to back up data.
- If it's necessary to back up Office container data, you can use a less expensive storage or a different backup frequency and retention period.
- For a personal host pool type, you should execute the backup at the session host VM level. This method only applies if the data is stored locally.
- If you use OneDrive and known folder redirection, the requirement to save data inside the container might disappear.

 **Note**

OneDrive backup is not considered in this article and scenario.

- Unless there's another requirement, backup for the storage in the primary region should be enough. Backup of the disaster recovery environment isn't normally used.
- For Azure Files share, use [Azure Backup](#).
 - For the vault [resiliency type](#), use zone-redundant storage if off-site or region backup storage isn't required. If those backups are required, use geo-redundant storage.
- Azure NetApp Files provides its own backup solution. This solution is currently in preview and can provide zone-redundant storage resiliency.
 - Make sure you check the region [feature availability](#), along with requirements and limitations.
- The separate storage accounts used for MSIX should also be covered by a backup if the application packages repositories can't be easily rebuilt.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Ben Martin Baur](#) | Cloud Solution Architect
- [Igor Pagliai](#) | FastTrack for Azure (FTA) Principal Engineer

Other contributors:

- [Nelson Del Villar](#) | Cloud Solution Architect, Azure Core Infrastructure
- [Jason Martinez](#) | Technical Writer

Next steps

- [Azure Virtual Desktop disaster recovery plan](#)
- [BCDR for Azure Virtual Desktop - Cloud Adoption Framework](#)
- [Cloud Cache to create resiliency and availability](#)

Related resources

- [Design reliable Azure applications](#)
- [Azure files accessed on-premises and secured by AD DS](#)
- [Azure Virtual Desktop for the enterprise](#)
- [Enterprise file shares with disaster recovery](#)
- [FSLogix configuration examples](#)

Azure Virtual Desktop for Azure Stack HCI (preview)

Article • 01/19/2024

ⓘ Important

Azure Virtual Desktop for Azure Stack HCI is currently in PREVIEW. See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

With Azure Virtual Desktop for Azure Stack HCI (preview), you can deploy session hosts for Azure Virtual Desktop where you need them. If you already have an existing on-premises virtual desktop infrastructure (VDI) deployment, Azure Virtual Desktop for Azure Stack HCI can improve your experience. If you're already using Azure Virtual Desktop on Azure, you can extend your deployment to your on-premises infrastructure to better meet your performance or data locality needs.

Azure Virtual Desktop for Azure Stack HCI isn't an Azure Arc-enabled service. As such, it's not supported as a standalone service outside of Azure, in a multicloud environment, or on Azure Arc-enabled servers besides Azure Stack HCI virtual machines as described in this article.

Benefits

With Azure Virtual Desktop for Azure Stack HCI, you can:

- Improve performance for Azure Virtual Desktop users in areas with poor connectivity to the Azure public cloud by giving them session hosts closer to their location.
- Meet data locality requirements by keeping app and user data on-premises. For more information, see [Data locations for Azure Virtual Desktop](#).
- Improve access to legacy on-premises apps and data sources by keeping desktops and apps in the same location.
- Reduce cost and improve user experience with Windows 10 and Windows 11 Enterprise multi-session, which allows multiple concurrent interactive sessions.

- Simplify your VDI deployment and management compared to traditional on-premises VDI solutions by using the Azure portal.
- Achieve the best performance by using [RDP Shortpath](#) for low-latency user access.
- Deploy the latest fully patched images quickly and easily using [Azure Marketplace images](#).

Supported deployment configurations

Your Azure Stack HCI clusters need to be running a minimum of [version 23H2](#) and [registered with Azure](#).

Once your cluster is ready, you can use the following 64-bit operating system images for your session hosts that are in support:

- Windows 11 Enterprise multi-session
- Windows 11 Enterprise
- Windows 10 Enterprise multi-session
- Windows 10 Enterprise
- Windows Server 2022
- Windows Server 2019

To use session hosts on Azure Stack HCI with Azure Virtual Desktop, you also need to:

- License and activate the virtual machines. For activating Windows 10 and Windows 11 Enterprise multi-session, and Windows Server 2022 Datacenter: Azure Edition, use [Azure verification for VMs](#). For all other OS images (such as Windows 10 and Windows 11 Enterprise, and other editions of Windows Server), you should continue to use existing activation methods. For more information, see [Activate Windows Server VMs on Azure Stack HCI](#).
- Install the [Azure Connected Machine agent](#) on the virtual machines so they can communicate with [Azure Instance Metadata Service](#), which is a [required endpoint for Azure Virtual Desktop](#). The Azure Connected Machine agent is automatically installed when you add session hosts using the Azure portal as part of the process to [Deploy Azure Virtual Desktop](#) or [Add session hosts to a host pool](#).

Finally, users can connect using the same [Remote Desktop clients](#) as Azure Virtual Desktop.

Licensing and pricing

To run Azure Virtual Desktop on Azure Stack HCI, you need to make sure you're licensed correctly and be aware of the pricing model. There are three components that affect how much it costs to run Azure Virtual Desktop for Azure Stack HCI:

- **User access rights.** The same licenses that grant access to Azure Virtual Desktop on Azure also apply to Azure Virtual Desktop for Azure Stack HCI. Learn more at [Azure Virtual Desktop pricing](#).
- **Infrastructure costs.** Learn more at [Azure Stack HCI pricing](#).
- **Hybrid service fee.** This fee requires you to pay for each active virtual CPU (vCPU) for your Azure Virtual Desktop session hosts running on Azure Stack HCI. This fee becomes active once the preview period ends.

Data storage

There are different classifications of data for Azure Virtual Desktop, such as customer input, customer data, diagnostic data, and service-generated data. With Azure Stack HCI, you can choose to store user data on-premises when you deploy session host virtual machines (VMs) and associated services such as file servers. However, some customer data, diagnostic data, and service-generated data is still stored in Azure. For more information on how Azure Virtual Desktop stores different kinds of data, see [Data locations for Azure Virtual Desktop](#).

Limitations

Azure Virtual Desktop for Azure Stack HCI has the following limitations:

- Session hosts running on Azure Stack HCI don't support some Azure Virtual Desktop features, such as:
 - [Azure Virtual Desktop Insights](#)
 - [Autoscale](#)
 - [Session host scaling with Azure Automation](#)
 - [Start VM On Connect](#)
 - [Multimedia redirection](#)
 - [Per-user access pricing](#)
- Each host pool must only contain session hosts on Azure or on Azure Stack HCI. You can't mix session hosts on Azure and on Azure Stack HCI in the same host pool.
- Session hosts on Azure Stack HCI don't support certain cloud-only Azure services.

- Azure Stack HCI supports many types of hardware and on-premises networking capabilities, so performance and user density might vary compared to session hosts running on Azure. Azure Virtual Desktop's [virtual machine sizing guidelines](#) are broad, so you should use them for initial performance estimates and monitor after deployment.
- Templates may show failures in certain cases at the domain-joining step. To proceed, you can manually join the session hosts to the domain. For more information, see [VM provisioning through Azure portal on Azure Stack HCI](#).

Next steps

To learn how to deploy Azure Virtual Desktop for Azure Stack HCI, see [Deploy Azure Virtual Desktop](#).

Configuration examples

Article • 12/05/2023

The example configurations outlined in this article are a progression of complexity based on configuration choices. Each example has an associated configuration focused on redundancy or disaster recovery. We recommend customers select the simplest configuration for their environment. Adding unnecessary complexity leads to incorrect configurations and support cases.

ⓘ Note

Use these examples as a **starting** point of your FSLogix configuration. The ideas and concepts in these examples should inform your unique organizational requirements.

EXAMPLE 1: Standard

The **Standard** configuration example is the simplest configuration in which most customers should consider.

Prerequisites (Standard)

- ✓ [FSLogix prerequisites including antivirus exclusions](#)
- ✓ Azure Virtual Desktop or equivalent Virtual Desktop infrastructure.
- ✓ SMB File Share.
- ✓ Validated [share and NTFS permissions](#).

Configuration Items (Standard)

ⓘ [Expand table](#)

Items	Description
Single VHD location	The VHDLocations setting contains a single UNC path to an SMB file share.
Single container	A single Profile container is created for the user. The ODFC container isn't configured.
No concurrent connections	The ProfileType setting is set to 0 or <i>not configured</i> . A user's profile can only be mounted within a single connection.
No custom profile redirections	No use of <code>redirections.xml</code> file.

Registry Settings (Standard)

 Expand table

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ¹	DWORD	1	Recommended
FlipFlopProfileDirectoryName ²	DWORD	1	Recommended
LockedRetryCount ³	DWORD	3	Recommended
LockedRetryInterval ³	DWORD	15	Recommended
ProfileType ⁴	DWORD	0	Default
ReAttachIntervalSeconds ³	DWORD	15	Recommended
ReAttachRetryCount ³	DWORD	3	Recommended
SizeInMBs	DWORD	30000	Default
VHDLocations	MULTI_SZ or REG_SZ	\\ <storage-account-name>.file.core.windows.net\ <share-name></storage-account-name>	Example
VolumeType ⁵	REG_SZ	VHDX	Recommended

1 Recommended to ensure user's don't use local profiles and lose data unexpectedly.

2 Provides an easier way to browse the container directories.

3 Decreases the retry timing to enable a faster fail scenario.

4 Single connections reduce complexity and increase performance.

5 VHDX is preferred over VHD due to its supported size and reduced corruption scenarios.

EXAMPLE 2: Standard + High Availability (Cloud Cache)

The **Standard + High Availability** configuration enhances the basic **Standard** example by incorporating Cloud Cache to ensure regional availability for the profile container. This configuration is designed to provide robustness and redundancy, allowing the profile container to be accessible even in the event of failures or outages in a specific region. Cloud Cache acts as a resiliency and availability layer, periodically writing profile data upstream to multiple storage providers. By replicating data across unique storage providers, it ensures that the profile container remains available even if one storage

provider is unavailable. This approach enhances reliability and minimizes downtime for end-users.

Key Points

- **Redundant and robust:** Allows the profile container to be accessible even in the event of failures or outages, minimizing downtime for end-users
- **Resiliency:** Cloud Cache acts as an availability layer, periodically writing profile data upstream to multiple storage providers.
- **Storage design expertise:** Cloud Cache functionality is dependent on the performance of your storage providers.

Summary

The Standard + High Availability configuration combines the benefits of the Standard setup with additional measures to maintain availability across regions, making it suitable for critical applications that require continuous access to profile data.

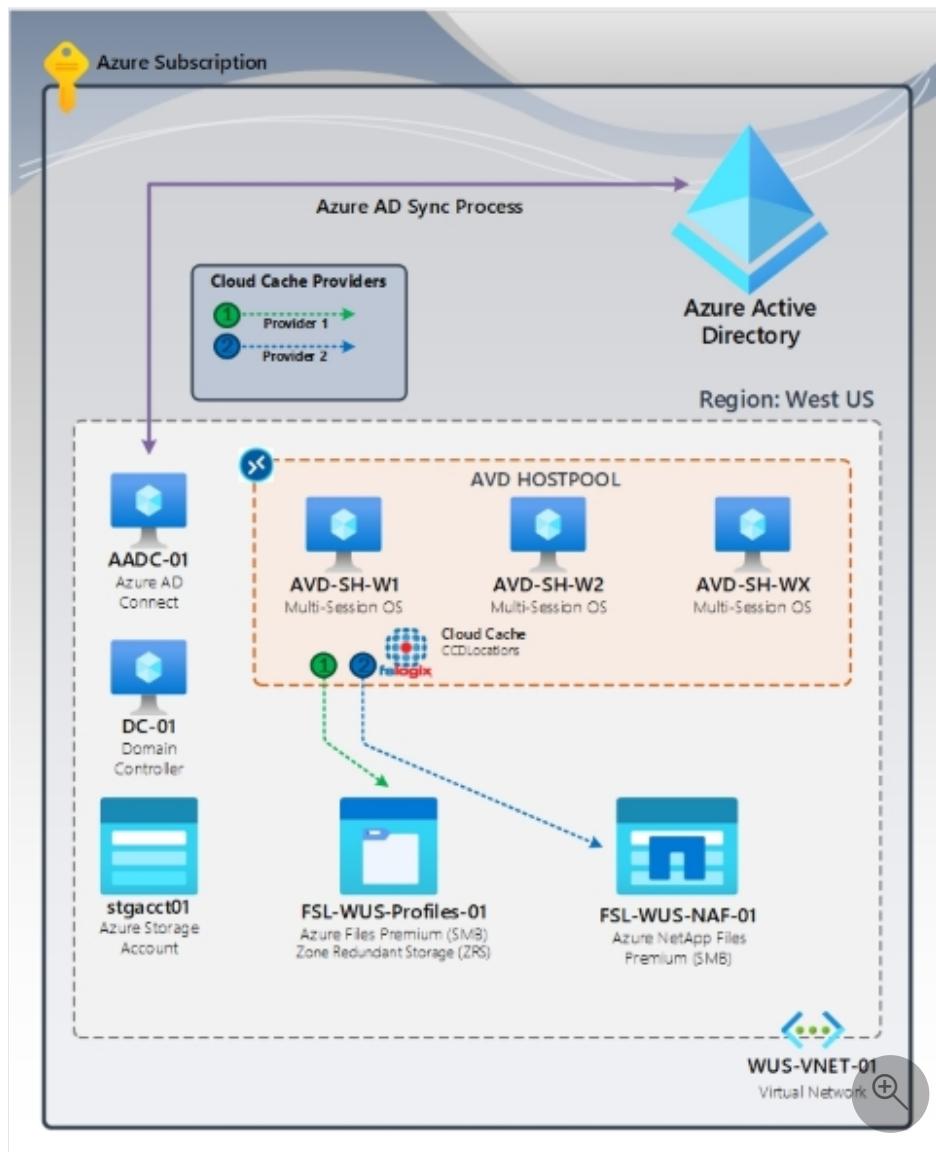


Figure 1: FSLogix High Availability using Cloud Cache

Prerequisites (Standard + High Availability)

- ✓ FSLogix prerequisites including antivirus exclusions
- ✓ Azure Virtual Desktop or equivalent Virtual Desktop infrastructure.
- ✓ Multiple storage providers in the same region or zone-redundant storage.
- ✓ Validated [share and NTFS permissions](#) (SMB only).

Configuration Items (Standard + High Availability)

[\[+\] Expand table](#)

Items	Description
Container redundancy	The CCDLocations contains at least 2 storage providers of varying kinds. The storage providers are in the SAME region as the virtual machines.
Single container	A single Profile container is created for the user. The ODFC container isn't configured.
No concurrent connections	The ProfileType setting is set to 0 or <i>not configured</i> . A user's profile can only be mounted within a single connection.
No custom profile redirections	No use of redirections.xml file.

Registry Settings (Standard + High Availability)

[\[+\] Expand table](#)

Key Name	Data Type	Value	Description
CCDLocations	MULTI_SZ or REG_SZ	<code>type=smb,name="FILES SMB PROVIDER",connectionString=\\<storage-account-name->.file.core.windows.net\\<share->;type=smb,name="ANF SMB PROVIDER",connectionString=\\<azure-netapp-files-fqdn>\\<volume-name></code>	Example
ClearCacheOnLogoff ¹	DWORD	1	Recommended

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ²	DWORD	1	Recommended
FlipFlopProfileDirectoryName ³	DWORD	1	Recommended
HealthyProvidersRequiredForRegister ⁴	DWORD	1	Recommended
LockedRetryCount ⁵	DWORD	3	Recommended
LockedRetryInterval ⁵	DWORD	15	Recommended
ProfileType ⁶	DWORD	0	Default
ReAttachIntervalSeconds ⁵	DWORD	15	Recommended
ReAttachRetryCount ⁵	DWORD	3	Recommended
SizeInMBs	DWORD	30000	Default
VolumeType ⁷	REG_SZ	VHDX	Recommended

1 Recommended to save disk space on the local disk and risk of data loss when using pooled desktops.

2 Recommended to ensure user's don't use local profiles and lose data unexpectedly.

3 Provides and easier way to browse the container directories.

4 Prevents users from creating a local cache if at least 1 provider isn't healthy.

5 Decreases the retry timing to enable a faster fail scenario.

6 Single connections reduce complexity and increase performance.

7 VHDX is preferred over VHD due to its supported size and reduced corruption scenarios.

EXAMPLE 3: Standard + Disaster Recovery (no profile recovery)

The **Standard + Disaster Recovery** is an extension of the basic **Standard**. In this setup, duplicate infrastructure exists in another region, but it remains **powered down** until needed. Unlike other recovery scenarios, there is **no profile recovery** in this approach. Instead, users create new profiles in the alternate location. While this is the **least complex recovery scenario**, it comes with a significant drawback: **end-user experience and training** become critical components for success.

Key Points

- **Duplicate Infrastructure:** The disaster recovery region mirrors the primary infrastructure but remains inactive until required.
- **No Profile Recovery:** Instead of restoring existing profiles, users create new ones in the alternate location.
- **Simplicity:** This approach minimizes complexity but relies heavily on user familiarity and training.
- **End-User Experience:** Ensuring a smooth transition and user understanding is crucial.

Summary

The **Standard + Disaster Recovery configuration** balances simplicity with the need for user education and adaptation.

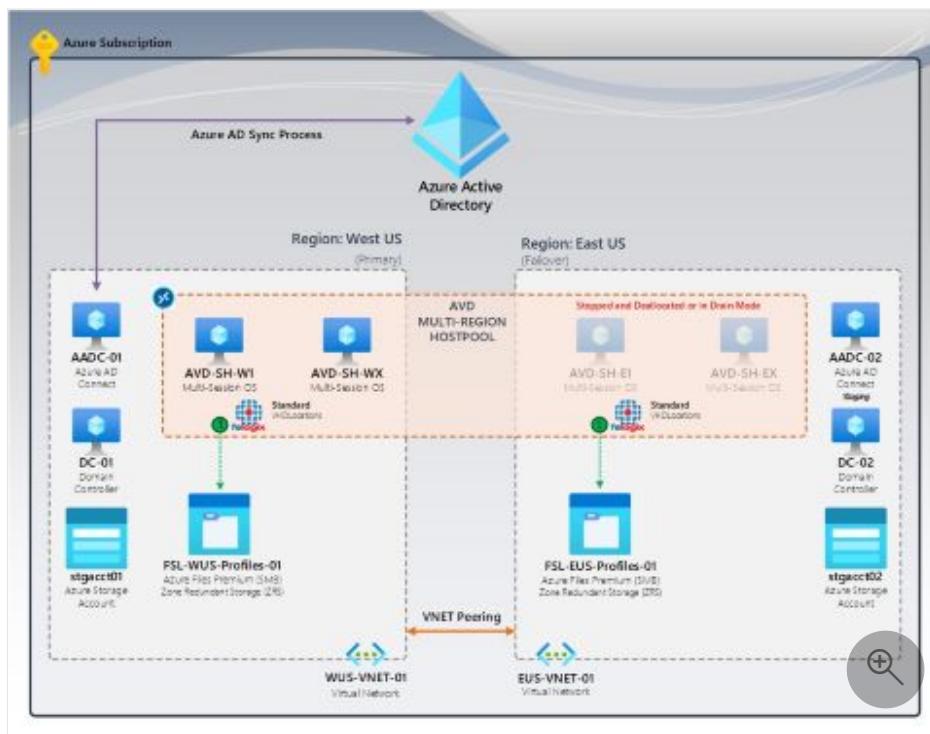


Figure 2: No Profile Recovery | FSLogix standard containers (VHDLocations)

Prerequisites (Standard + Disaster Recovery)

- ✓ **FSLogix prerequisites including antivirus exclusions**
- ✓ Azure Virtual Desktop or equivalent Virtual Desktop infrastructure.
- ✓ Duplicate storage and compute infrastructure in another region.
- ✓ Validated NTFS and share-level permissions (SMB only).

Configuration Items (Standard + Disaster Recovery)

[] [Expand table](#)

Items	Description
Single VHD location	The VHDLocations setting contains a single UNC path to an SMB file share.
Single container	A single Profile container is created for the user. The ODFC container isn't configured.
No concurrent connections	The ProfileType setting is set to 0 or <i>not configured</i> . A user's profile can only be mounted within a single connection.
No custom profile redirections	No use of <code>redirections.xml</code> file.

Registry Settings (Standard + Disaster Recovery)

[Expand table](#)

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ¹	DWORD	1	Recommended
FlipFlopProfileDirectoryName ²	DWORD	1	Recommended
LockedRetryCount ³	DWORD	3	Recommended
LockedRetryInterval ³	DWORD	15	Recommended
ProfileType ⁴	DWORD	0	Default
ReAttachIntervalSeconds ³	DWORD	15	Recommended
ReAttachRetryCount ³	DWORD	3	Recommended
SizeInMBs	DWORD	30000	Default
VHDLocations	MULTI_SZ or REG_SZ	\\\<storage-account-name>.file.core.windows.net\<share-name>	Example
VolumeType ⁵	REG_SZ	VHDX	Recommended

¹ Recommended to ensure user's don't use local profiles and lose data unexpectedly.

² Provides an easier way to browse the container directories.

³ Decreases the retry timing to enable a faster fail scenario.

⁴ Single connections reduce complexity and increase performance.

⁵ VHDX is preferred over VHD due to its supported size and reduced corruption scenarios.

EXAMPLE 4: Advanced

The Advanced configuration example builds upon the [Standard](#) example by introducing additional features to enhance flexibility and customization.

Key Points

- **Multiple VHDLocations or object-specific settings:** You can specify multiple locations for storing user profiles (VHDLocations). Alternatively, you can define object-specific settings to tailor profile behavior for specific users or groups. This flexibility allows you to optimize profile management based on your organization's needs.
- **Minimal entries in custom profile redirections:** Unlike the Standard setup, where the redirections.xml file isn't used, the Advanced configuration minimizes the number of redirections.xml entries. Each entry in the redirections.xml configuration adds complexity and can cause unknown application behaviors. Minimizing these entries may provide an overall better user experience.

Summary

The Advanced configuration provides granular control over profile storage and redirection, making it suitable for organizations with diverse requirements.

Prerequisites (Advanced)

- ✓ [FSLogix prerequisites including antivirus exclusions](#)
- ✓ Azure Virtual Desktop or equivalent Virtual Desktop infrastructure.
- ✓ Multiple SMB File Share(s).
- ✓ Validated [share and NTFS permissions](#) (SMB only).

Configuration Items (Advanced)

Expand table

Items	Description
Multiple VHD locations	The VHDLocations setting contains a single or multiple UNC paths (<i>separated by semi-colon</i>) to SMB file shares.
Object-specific settings	Allows unique settings based on a user or group SID.
Single container	A single profile container is created for the user. The ODFC container isn't configured.

Items	Description
No concurrent connections	The ProfileType setting is set to 0 or <i>not configured</i> . A user's profile can only be mounted within a single connection.
Minimal redirections.xml	XML file contains minimal entries with minor complexity.

💡 Tip

- Review the [Custom profile redirections.xml](#) page for additional information.
- The [Tutorial: Create and implement redirections.xml](#) page demonstrates how to implement this feature for Microsoft Teams.

Registry Settings (Advanced)

Multiple VHDLocations

[\[+\] Expand table](#)

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ¹	DWORD	1	Recommended
FlipFlopProfileDirectoryName ²	DWORD	1	Recommended
LockedRetryCount ³	DWORD	3	Recommended
LockedRetryInterval ³	DWORD	15	Recommended
ProfileType ⁴	DWORD	0	Default
ReAttachIntervalSeconds ³	DWORD	15	Recommended
ReAttachRetryCount ³	DWORD	3	Recommended
RedirXMLSourceFolder	REG_SZ	<code>\\"<server-name>\<share-name></code>	Example
SizeInMBs	DWORD	30000	Default
VHDLocations	MULTI_SZ or REG_SZ	<code>\\"<storage-account-name-1>.file.core.windows.net\<share-name>\\<storage-</code>	Example

Key Name	Data Type	Value	Description
		account-name- 2>.file.core.windows.net\ <share-name>	
VolumeType ⁵	REG_SZ	VHDX	Recommended

Object-Specific VHDLocations

The default VHDLocations is used for any user or group *not* matched by the object-specific configuration.

Registry Path: `HKLM:\SOFTWARE\FSLogix\Profiles\`

 Expand table

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ¹	DWORD	1	Recommended
FlipFlopProfileDirectoryName ²	DWORD	1	Recommended
LockedRetryCount ³	DWORD	3	Recommended
LockedRetryInterval ³	DWORD	15	Recommended
ProfileType ⁴	DWORD	0	Default
ReAttachIntervalSeconds ³	DWORD	15	Recommended
ReAttachRetryCount ³	DWORD	3	Recommended
RedirXMLSourceFolder	REG_SZ	<code>\<server-name>\<share-name></code>	Example
SizeInMBs	DWORD	30000	Default
VHDLocations	MULTI_SZ or REG_SZ	<code>\<storage-account-name>.file.core.windows.net\<share-name></code>	Example
VolumeType ⁵	REG_SZ	VHDX	Recommended

Registry Path: `HKLM:\SOFTWARE\FSLogix\Profiles\ObjectSpecific\S-0-0-00-000000000-000000000-000000000-1234\`

[Expand table](#)

Key Name	Data Type	Value	Description
VHDLocations	MULTI_SZ or REG_SZ	\<server-name>\<share-name>	Example

Registry Path: `HKLM:\SOFTWARE\FSLogix\Profiles\ObjectSpecific\S-0-0-00-000000000-000000000-000000000-4321\`

[Expand table](#)

Key Name	Data Type	Value	Description
VHDLocations	MULTI_SZ or REG_SZ	\<azure-netapp-files-computer-account>.contoso.com\<share-name>	Example

⚠ Warning

Multiple entries in **VHDLocations** doesn't provide container resiliency. When multiple entries exist, a user will try to create or locate their container from the list of locations in order. The first location which the user has access to or is available will be where the container is created or attached from. If using multiple entries, users should only have access to a single location. *Consider using the object-specific configuration settings in lieu of multiple VHDLocations.*

1 Recommended to ensure user's don't use local profiles and lose data unexpectedly.

2 Provides and easier way to browse the container directories.

3 Decreases the retry timing to enable a faster fail scenario.

4 Single connections reduce complexity and increase performance.

5 VHDX is preferred over VHD due to its supported size and reduced corruption scenarios.

EXAMPLE 5: Advanced + Disaster Recovery (primary / failover)

The Advanced + [Disaster Recovery](#) configuration example adds complexity through a failover design. This is a common strategy to ensure the availability and reliability of your infrastructure in case of a disaster or a failure. With Cloud Cache, you can configure your devices to use two (2) storage providers that store your profile data in different locations. Cloud Cache synchronizes your profile data to each of the two storage providers asynchronously, so you always have the latest version of your data. Some of your devices are in the primary location and the other devices are in the failover location. Cloud Cache prioritizes the first storage provider (*closest to your device*), and uses the other storage

provider as a backup. For example, if your primary device is in West US and your failover device is in East US, you can configure Cloud Cache as follows:

- The primary device uses a storage provider in West US as the first option and a storage provider in East US as the second option.
- The failover device uses a storage provider in East US as the first option and a storage provider in West US as the second option.
- If the primary device or the closest storage provider fails, you can switch to the failover device or the backup storage provider and continue your work without losing your profile data.

Key Points

- **Failover design:** This design ensures the availability and reliability of your infrastructure in case of a disaster or a failure.
- **Profile storage:** Cloud Cache enables you to store your profile data in different locations.
- **Additional storage cost:** Multiple storage locations will increase the overall cost.
- **Operational excellence:** Manual failover process, which may require the approval of the business stakeholders and process validation.
- **End-user experience:** You may experience some latency or inconsistency in your profile data due to the asynchronous synchronization to the two storage providers.

Summary

The Advanced + Disaster Recovery configuration shows how a failover design with Cloud Cache can enhance the reliability and availability of your infrastructure by using two storage providers in different locations. It also highlights the drawbacks of this approach, including additional costs, the need for manual failover initiation, and potential latency or inconsistency in profile data.

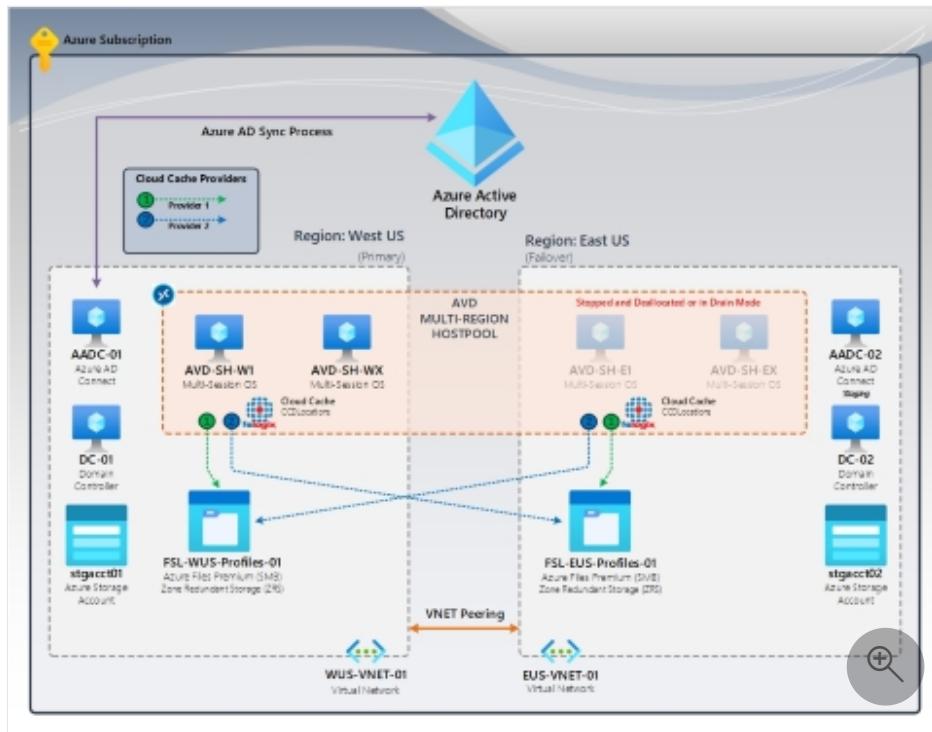


Figure 3: Cloud Cache (primary / failover) | FSLogix Cloud Cache (CCDLocations)

Prerequisites (Advanced + Disaster Recovery)

- ✓ FSLogix prerequisites including antivirus exclusions
- ✓ Azure Virtual Desktop or equivalent Virtual Desktop infrastructure.
- ✓ Two storage providers in at least two regions.
- ✓ Validated [share and NTFS permissions](#) (SMB only).

Configuration Items (Advanced + Disaster Recovery)

[\[+\] Expand table](#)

Items	Description
Container redundancy	The CCDLocations contains two (2) storage providers ¹ . The primary site is configured with the closest storage provider listed first and failover storage provider provided listed last. The failover site has the reverse configuration. The closest storage provider first, followed by the primary site storage providers listed last.
Single container	A single profile container is created for the user. The ODFC container isn't configured.
No concurrent connections	The ProfileType setting is set to 0 or <i>not configured</i> . A user's profile can only be mounted within a single connection.
No custom profile	No use of <code>redirections.xml</code> file.

Items	Description
redirections	

1 The storage providers must be in different regions or locations.

💡 Tip

Review the [Custom profile redirections](#) page for our recommended exclusions.

Registry Settings (Advanced + Disaster Recovery)

Primary site

[\[+\] Expand table](#)

Key Name	Data Type	Value	Description
CCDLocations	MULTI_SZ or REG_SZ	<pre>type=smb, name="FILES SMB" PRIMARY", connectionString=\\ <storage-account-name- primary>.file.core.windows.net\ <share- name>;type=smb, name="FILES SMB PROVIDER FAILOVER", connectionString=\\ <storage-account-name- failover>.file.core.windows.net\ <share-name></pre>	Example
ClearCacheOnLogoff ¹	DWORD	1	Recommended
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ²	DWORD	1	Recommended
FlipFlopProfileDirectoryName ³	DWORD	1	Recommended
HealthyProvidersRequiredForRegister ⁴	DWORD	1	Recommended
LockedRetryCount ⁵	DWORD	3	Recommended
LockedRetryInterval ⁵	DWORD	15	Recommended
ProfileType ⁶	DWORD	0	Default
ReAttachIntervalSeconds ⁵	DWORD	15	Recommended

Key Name	Data Type	Value	Description
ReAttachRetryCount ⁵	DWORD	3	Recommended
SizeInMBs	DWORD	30000	Default
VolumeType ⁷	REG_SZ	VHDX	Recommended

Failover site

[Expand table](#)

Key Name	Data Type	Value	Description
CCDLocations	MULTI_SZ or REG_SZ	<pre>type=smb,name="FILES SMB or REG_SZ <storage-account-name- failover>.file.core.windows.net\ <share- name>;type=smb,name="FILES SMB PROVIDER PRIMARY",connectionString=\\ <storage-account-name- primary>.file.core.windows.net\ <share-name></pre>	Example
ClearCacheOnLogoff ¹	DWORD	1	Recommended
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ²	DWORD	1	Recommended
FlipFlopProfileDirectoryName ³	DWORD	1	Recommended
HealthyProvidersRequiredForRegister ⁴	DWORD	1	Recommended
LockedRetryCount ⁵	DWORD	3	Recommended
LockedRetryInterval ⁵	DWORD	15	Recommended
ProfileType ⁶	DWORD	0	Default
ReAttachIntervalSeconds ⁵	DWORD	15	Recommended
ReAttachRetryCount ⁵	DWORD	3	Recommended
SizeInMBs	DWORD	30000	Default
VolumeType ⁷	REG_SZ	VHDX	Recommended

- 1 Recommended to save disk space on the local disk and risk of data loss when using pooled desktops.
- 2 Recommended to ensure user's don't use local profiles and lose data unexpectedly.
- 3 Provides an easier way to browse the container directories.
- 4 Prevents users from creating a local cache if at least 1 provider isn't healthy.
- 5 Decreases the retry timing to enable a faster fail scenario.
- 6 Single connections reduce complexity and increase performance.
- 7 VHDX is preferred over VHD due to its supported size and reduced corruption scenarios.

EXAMPLE 6: Complex

The **Complex** configuration example builds upon the [Advanced](#) example by introducing [multiple connections](#). In this setup, user profiles can handle multiple connections, allowing a single user to have active sessions across different devices simultaneously. Despite the increased complexity, the goal remains to provide a seamless experience for end-users. Properly configured multiple connections enhance productivity and flexibility, making this configuration suitable for organizations with diverse needs and high demands.

ⓘ Note

Azure Virtual Desktop does not support multiple connections within the **same** Host Pool.

Summary

The Complex configuration balances sophistication with user-centric design, making it ideal for large organizations requiring scalability and robust profile management.

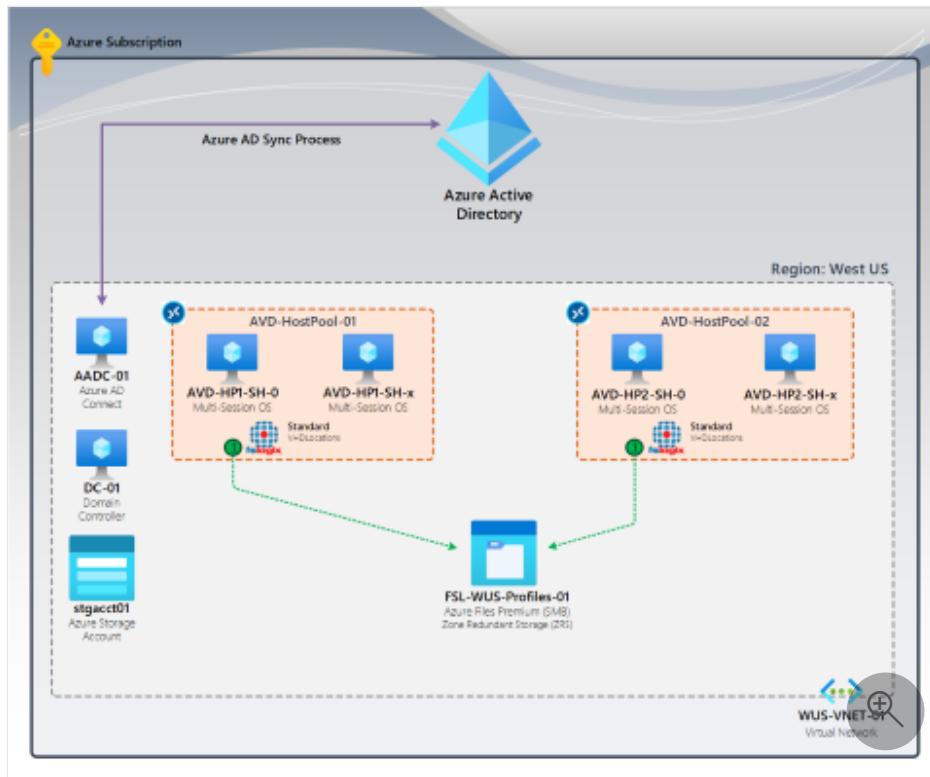


Figure 4: Complex example with multiple connections | FSLogix standard containers (VHDLocations)

Prerequisites (Complex)

- ✓ [FSLogix prerequisites including antivirus exclusions](#)
- ✓ Azure Virtual Desktop or equivalent Virtual Desktop infrastructure.
- ✓ Separate pools of virtual machines.
- ✓ Multiple SMB file share(s) (*not for high availability*).
- ✓ Validated [share and NTFS permissions](#) (SMB only).

Configuration Items (Complex)

[Expand table](#)

Items	Description
Multiple VHD location	The VHDLocations setting contains a single or multiple UNC paths (<i>separated by semi-colon</i>) to SMB file shares.
Object-specific settings	Allows unique settings based on a user or group SID.
Single container	A single profile container is created for the user. The ODFC container isn't configured.
Concurrent connection(s)	ProfileType is set to 3. Users can have multiple sign-ins, but only one (1) session allows writes to the base VHD disk.

Items	Description
[OPTIONAL] redirections.xml	XML file contains various entries with added complexity.

Tip

- Review the [Custom profile redirections.xml](#) page for additional information.
- The [Tutorial: Create and implement redirections.xml](#) page demonstrates how to implement this feature for Microsoft Teams.

Registry Settings (Complex)

Multiple VHDLocations

[\[+\] Expand table](#)

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ¹	DWORD	1	Recommended
FlipFlopProfileDirectoryName ²	DWORD	1	Recommended
LockedRetryCount ³	DWORD	3	Recommended
LockedRetryInterval ³	DWORD	15	Recommended
ProfileType ⁴	DWORD	0	Default
ReAttachIntervalSeconds ³	DWORD	15	Recommended
ReAttachRetryCount ³	DWORD	3	Recommended
RedirXMLSourceFolder	REG_SZ	<code>\\"<server-name>\<share-name></code>	Example
SizeInMBs	DWORD	30000	Default
VHDLocations	MULTI_SZ or REG_SZ	<code>\\"<storage-account-name-1>.file.core.windows.net\<share-name>;\\"<storage-account-name-2>.file.core.windows.net\<share-name></code>	Example

Key Name	Data Type	Value	Description
VolumeType ⁵	REG_SZ	VHDX	Recommended

Object-Specific VHDLocations

The default VHDLocations is used for any user or group *not* matched by the object-specific configuration.

Registry Path: `HKLM:\SOFTWARE\FSLogix\Profiles\`

[Expand table](#)

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ¹	DWORD	1	Recommended
FlipFlopProfileDirectoryName ²	DWORD	1	Recommended
LockedRetryCount ³	DWORD	3	Recommended
LockedRetryInterval ³	DWORD	15	Recommended
ProfileType ⁴	DWORD	0	Default
ReAttachIntervalSeconds ³	DWORD	15	Recommended
ReAttachRetryCount ³	DWORD	3	Recommended
RedirXMLSourceFolder	REG_SZ	<code>\<server-name>\<share-name></code>	Example
SizeInMBs	DWORD	30000	Default
VHDLocations	MULTI_SZ or REG_SZ	<code>\<storage-account-name>.file.core.windows.net\<share-name></code>	Example
VolumeType ⁵	REG_SZ	VHDX	Recommended

Registry Path: `HKLM:\SOFTWARE\FSLogix\Profiles\ObjectSpecific\S-0-0-00-00000000-00000000-00000000-1234\`

[Expand table](#)

Key Name	Data Type	Value	Description
VHDLocations	MULTI_SZ or REG_SZ	\\\<server-name>\<share-name>	Example

Registry Path: `HKLM:\SOFTWARE\FSLogix\Profiles\ObjectSpecific\{S-0-0-00-0000000000-0000000000-0000000000-4321\}`

[\[+\] Expand table](#)

Key Name	Data Type	Value	Description
VHDLocations	MULTI_SZ or REG_SZ	\\\<azure-netapp-files-computer-account>.contoso.com\<share-name>	Example

⚠ Warning

Multiple entries in **VHDLocations** doesn't provide container resiliency. When multiple entries exist, a user will try to create or locate their container from the list of locations in order. The first location which the user has access to or is available will be where the container is created or attached from. If using multiple entries, users should only have access to a single location. *Consider using the object-specific configuration settings in lieu of multiple VHDLocations.*

1 Recommended to ensure user's don't use local profiles and lose data unexpectedly.

2 Provides and easier way to browse the container directories.

3 Decreases the retry timing to enable a faster fail scenario.

4 Single connections reduce complexity and increase performance.

5 VHDX is preferred over VHD due to its supported size and reduced corruption scenarios.

EXAMPLE 7: Complex + Disaster Recovery (active / active)

The **Complex + Disaster Recovery** configuration builds upon the **Advanced + Disaster Recovery** configuration by implementing an active/active design. Instead of load balancing between the two sites, this configuration relies on users having access to only one location. In the event of a drill or BCDR, users from a failed region are granted access to virtual machines in the functioning region.

Key Points

- **Failover capability:** In the event of a disaster, the surviving regions must have capacity to support all users.
- **Profile storage:** Cloud Cache enables you to store your profile data in different locations.
- **Additional storage cost:** Multiple storage locations will increase the overall cost.
- **Operational excellence:** Manual failover process, which may require the approval of the business stakeholders, process validation and proper user assignments.
- **End-user experience:** You may experience some latency or inconsistency in your profile data due to the asynchronous synchronization to the two storage providers.

Summary

The Complex + Disaster Recovery configuration with Cloud Cache provides redundancy and flexibility, but business decisions play a crucial role in initiating failover.

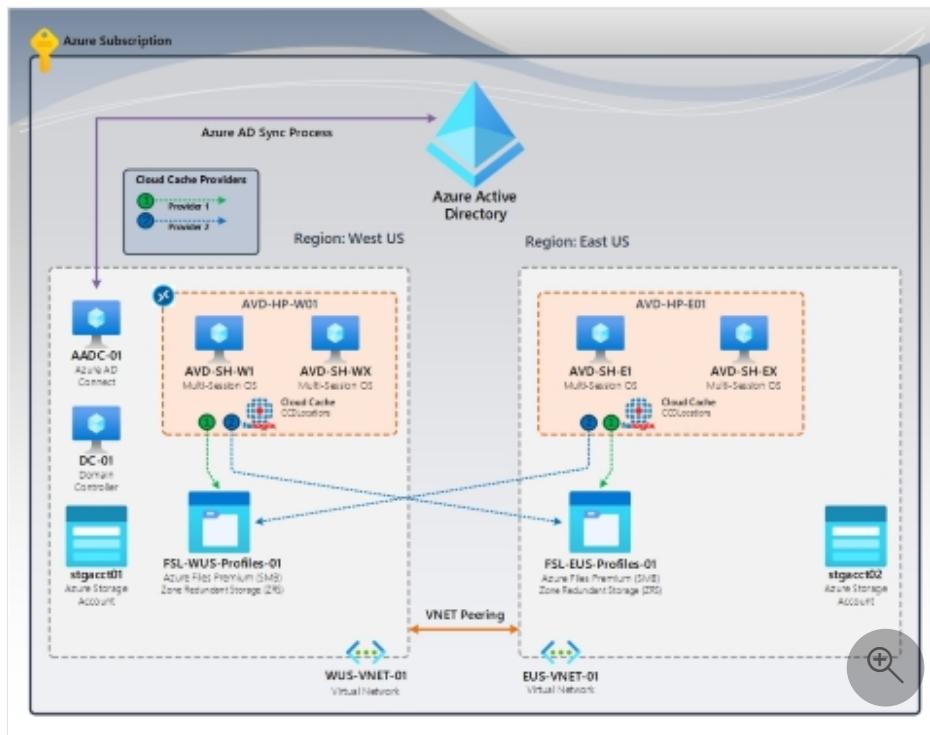


Figure 5: Cloud Cache (active / active) | FSLogix Cloud Cache (CCDLocations)

Prerequisites (Complex + Disaster Recovery)

- ✓ FSLogix prerequisites including antivirus exclusions
- ✓ Azure Virtual Desktop or equivalent Virtual Desktop infrastructure.
- ✓ Separate pools of virtual machines in each location.
- ✓ Users only have access to a single region at a time.
- ✓ Validated [share and NTFS permissions](#) (SMB only).

Configuration Items (Complex + Disaster Recovery)

[Expand table](#)

Items	Description
Container redundancy	The CCDLocations contains two (2) storage providers ¹ . The primary site is configured with the closest storage provider listed first and failover storage provider provided listed last. The failover site has the reverse configuration. The closest storage provider first, followed by the primary site storage providers listed last.
Single container	A Profile and ODFC container exists or is created for each user.
No concurrent connections	The ProfileType setting is set to 0 or <i>not configured</i> . A user's profile can only be mounted within a single connection.
[OPTIONAL] redirections.xml	XML file contains various entries with added complexity.

💡 Tip

- Review the [Custom profile redirections.xml](#) page for additional information.
- The [Tutorial: Create and implement redirections.xml](#) page demonstrates how to implement this feature for Microsoft Teams.

Registry Settings (Complex + Disaster Recovery)

Site A (West US)

[Expand table](#)

Key Name	Data Type	Value	Description
CCDLocations	MULTI_SZ or REG_SZ	<code>type=smb,name="FILES SMB WEST US",connectionString=\<storage- account-name- primary>.file.core.windows.net\<share- name>;type=smb,name="FILES SMB PROVIDER EAST US",connectionString=\<storage- account-name- failover>.file.core.windows.net\<share-name></code>	Example
ClearCacheOnLogoff ¹	DWORD	1	Recommended

Key Name	Data Type	Value	Description
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ²	DWORD	1	Recommended
FlipFlopProfileDirectoryName ³	DWORD	1	Recommended
HealthyProvidersRequiredForRegister ⁴	DWORD	1	Recommended
LockedRetryCount ⁵	DWORD	3	Recommended
LockedRetryInterval ⁵	DWORD	15	Recommended
ProfileType ⁶	DWORD	0	Default
ReAttachIntervalSeconds ⁵	DWORD	15	Recommended
ReAttachRetryCount ⁵	DWORD	3	Recommended
SizeInMBs	DWORD	30000	Default
VolumeType ⁷	REG_SZ	VHDX	Recommended

Site B (East US)

[Expand table](#)

Key Name	Data Type	Value	Description
CCDLocations	MULTI_SZ or REG_SZ	<pre>type=smb,name="FILES SMB EAST US",connectionString=\<storage- account-name- failover>.file.core.windows.net\ <share- name>;type=smb,name="FILES SMB PROVIDER WEST US",connectionString=\<storage- account-name- primary>.file.core.windows.net\ <share-name></pre>	Example
ClearCacheOnLogoff ¹	DWORD	1	Recommended
Enabled	DWORD	1	REQUIRED
DeleteLocalProfileWhenVHDShouldApply ²	DWORD	1	Recommended
FlipFlopProfileDirectoryName ³	DWORD	1	Recommended

Key Name	Data Type	Value	Description
HealthyProvidersRequiredForRegister ⁴	DWORD	1	Recommended
LockedRetryCount ⁵	DWORD	3	Recommended
LockedRetryInterval ⁵	DWORD	15	Recommended
ProfileType ⁶	DWORD	0	Default
ReAttachIntervalSeconds ⁵	DWORD	15	Recommended
ReAttachRetryCount ⁵	DWORD	3	Recommended
SizeInMBs	DWORD	30000	Default
VolumeType ⁷	REG_SZ	VHDX	Recommended

1 Recommended to save disk space on the local disk and risk of data loss when using pooled desktops.

2 Recommended to ensure user's don't use local profiles and lose data unexpectedly.

3 Provides and easier way to browse the container directories.

4 Prevents users from creating a local cache if at least 1 provider isn't healthy.

5 Decreases the retry timing to enable a faster fail scenario.

6 Single connections reduce complexity and increase performance.

7 VHDX is preferred over VHD due to its supported size and reduced corruption scenarios.

Appendix: Multiple `VHDLocations` logic diagram

When using multiple values in the `VHDLocations` setting, it's important to understand how FSLogix determines the location to use.

⚠ Warning

Users who have access to multiple locations may create a **new** profile in another location if the location for their actual profile is not available.

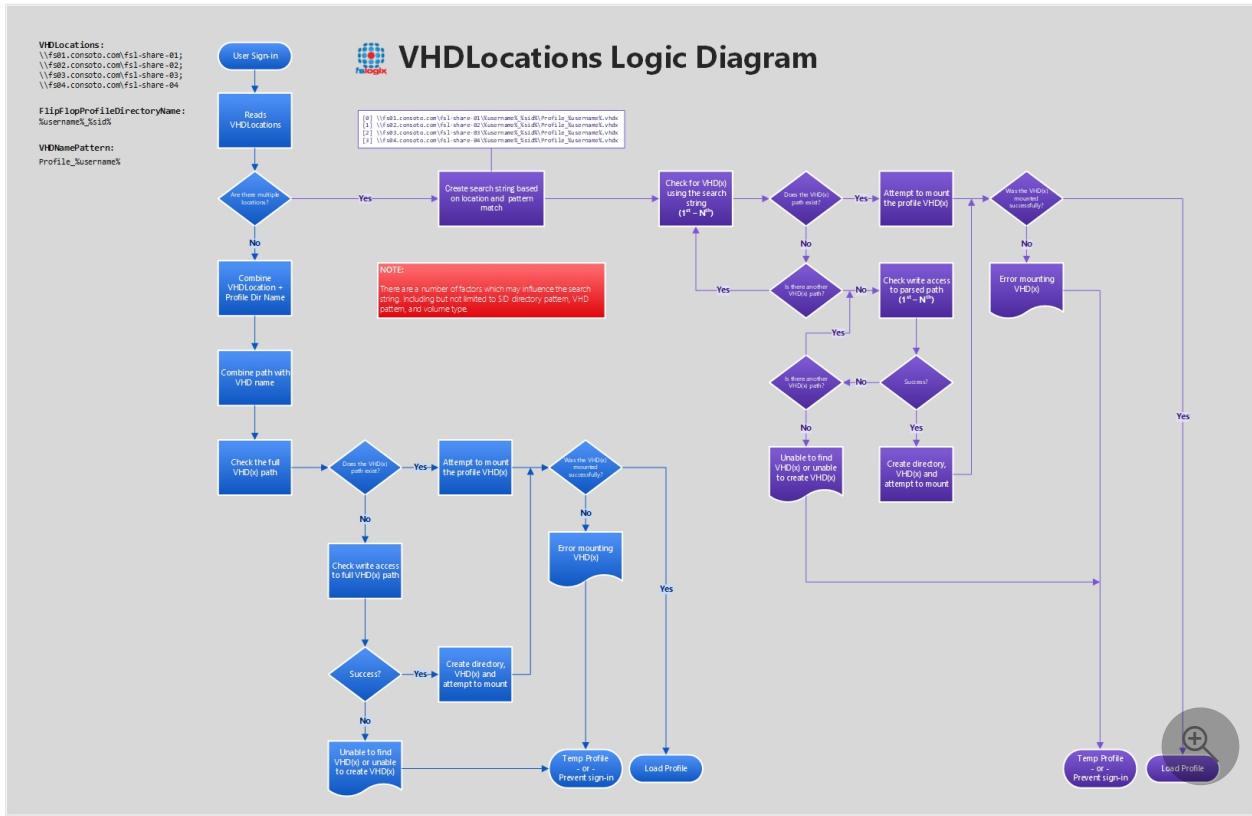


Figure 5: VHDLocations Logic Diagram

Next steps

How to use Group Policy Templates

FSLogix profile containers and Azure files

Article • 04/25/2023

The Azure Virtual Desktop service recommends FSLogix profile containers as a user profile solution. FSLogix is designed to roam profiles in remote computing environments, such as Azure Virtual Desktop. It stores a complete user profile in a single container. At sign in, this container is dynamically attached to the computing environment using natively supported Virtual Hard Disk (VHD) and Hyper-V Virtual Hard disk (VHDX). The user profile is immediately available and appears in the system exactly like a native user profile. This article describes how FSLogix profile containers used with Azure Files function in Azure Virtual Desktop.

ⓘ Note

If you're looking for comparison material about the different FSLogix Profile Container storage options on Azure, see [Storage options for FSLogix profile containers](#).

User profiles

A user profile contains data elements about an individual, including configuration information like desktop settings, persistent network connections, and application settings. By default, Windows creates a local user profile that is tightly integrated with the operating system.

A remote user profile provides a partition between user data and the operating system. It allows the operating system to be replaced or changed without affecting the user data. In Remote Desktop Session Host (RDSH) and Virtual Desktop Infrastructures (VDI), the operating system may be replaced for the following reasons:

- An upgrade of the operating system
- A replacement of an existing Virtual Machine (VM)
- A user being part of a pooled (non-persistent) RDSH or VDI environment

Microsoft products operate with several technologies for remote user profiles, including these technologies:

- Roaming user profiles (RUP)
- User profile disks (UPD)
- Enterprise state roaming (ESR)

UPD and RUP are the most widely used technologies for user profiles in Remote Desktop Session Host (RDSH) and Virtual Hard Disk (VHD) environments.

Challenges with previous user profile technologies

Existing and legacy Microsoft solutions for user profiles came with various challenges. No previous solution handled all the user profile needs that come with an RDSH or VDI environment. For example, UPD cannot handle large OST files and RUP does not persist modern settings.

Functionality

The following table shows benefits and limitations of previous user profile technologies.

Technology	Modern settings	Win32 settings	OS settings	User data	Supported on server	SKU	Back-end storage on Azure	Back-end storage on-premises	Version support	Subsequent sign in time	Notes
User Profile Disks (UPD)	Yes	Yes	Yes	Yes	Yes		No	Yes	Win 7+	Yes	
Roaming User Profile (RUP), maintenance mode	No	Yes	Yes	Yes	Yes		No	Yes	Win 7+	No	
Enterprise State Roaming (ESR)	Yes	No	Yes	No	See notes		Yes	No	Win 10	No	Functions on server SKU but no supporting user interface
User Experience Virtualization (UE-V)	Yes	Yes	Yes	No	Yes		No	Yes	Win 7+	No	
OneDrive cloud files	No	No	No	Yes	See notes		See notes	See Notes	Win 10 RS3	No	Not tested on server SKU. Back-end storage on Azure depends on sync client. Back-end storage on-premises needs a sync client.

Performance

UPD requires [Storage Spaces Direct \(S2D\)](#) to address performance requirements. UPD uses Server Message Block (SMB) protocol. It copies the profile to the VM in which the user is being logged.

Cost

While S2D clusters achieve the necessary performance, the cost is expensive for enterprise customers, but especially expensive for small and medium business (SMB) customers. For this solution, businesses pay for storage disks, along with the cost of the VMs that use the disks for a share.

Administrative overhead

S2D clusters require an operating system that is patched, updated, and maintained in a secure state. These processes and the complexity of setting up S2D disaster recovery make S2D feasible only for enterprises with a dedicated IT staff.

FSLogix profile containers

On November 19, 2018, [Microsoft acquired FSLogix](#). FSLogix addresses many profile container challenges. Key among them are:

- **Performance:** The [FSLogix profile containers](#) are high performance and resolve performance issues that have historically blocked cached exchange mode.
- **OneDrive:** Without FSLogix profile containers, OneDrive for Business is not supported in non-persistent RDSH or VDI environments. The [OneDrive VDI support page](#) will tell you how they interact. For more information, see [Use the sync client on virtual desktops](#).
- **Additional folders:** FSLogix provides the ability to extend user profiles to include additional folders.

Since the acquisition, Microsoft started replacing existing user profile solutions, like UPD, with FSLogix profile containers.

Best practices for Azure Virtual Desktop

Azure Virtual Desktop offers full control over size, type, and count of VMs that are being used by customers. For more information, see [What is Azure Virtual Desktop?](#).

To ensure your Azure Virtual Desktop environment follows best practices:

- Azure Files storage account must be in the same region as the session host VMs.
- Azure Files permissions should match permissions described in [Requirements - Profile Containers](#).
- Each host pool VM must be built of the same type and size VM based on the same master image.
- Each host pool VM must be in the same resource group to aid management, scaling and updating.
- For optimal performance, the storage solution and the FSLogix profile container should be in the same data center location.
- The storage account containing the master image must be in the same region and subscription where the VMs are being provisioned.

Next steps

- Learn more about storage options for FSLogix profile containers, see [Storage options for FSLogix profile containers in Azure Virtual Desktop](#).
- [Set up FSLogix Profile Container with Azure Files and Active Directory](#)
- [Set up FSLogix Profile Container with Azure Files and Azure Active Directory](#)
- [Set up FSLogix Profile Container with Azure NetApp Files](#)

Storage options for FSLogix profile containers in Azure Virtual Desktop

Article • 03/12/2023

Azure offers multiple storage solutions that you can use to store your FSLogix profile container. This article compares storage solutions that Azure offers for Azure Virtual Desktop FSLogix user profile containers. We recommend storing FSLogix profile containers on Azure Files for most of our customers.

Azure Virtual Desktop offers FSLogix profile containers as the recommended user profile solution. FSLogix is designed to roam profiles in remote computing environments, such as Azure Virtual Desktop. At sign-in, this container is dynamically attached to the computing environment using a natively supported Virtual Hard Disk (VHD) and a Hyper-V Virtual Hard Disk (VHDX). The user profile is immediately available and appears in the system exactly like a native user profile.

The following tables compare the storage solutions Azure Storage offers for Azure Virtual Desktop FSLogix profile container user profiles.

Azure platform details

Features	Azure Files	Azure NetApp Files	Storage Spaces Direct
Use case	General purpose	General purpose to enterprise scale	Cross-platform
Platform service	Yes, Azure-native solution	Yes, Azure-native solution	No, self-managed
Regional availability	All regions	Select regions 	All regions
Redundancy	Locally redundant/zone-redundant/geo-redundant/geo-zone-redundant	Locally redundant/zone-redundant with cross-zone replication /geo-redundant with cross-region replication	Locally redundant/zone-redundant/geo-redundant

Features	Azure Files	Azure NetApp Files	Storage Spaces Direct
Tiers and performance	Standard (Transaction optimized) Premium Up to max 100K IOPS per share with 10 GBps per share at about 3-ms latency	Standard Premium Ultra Up to max 460K IOPS per volume with 4.5 GBps per volume at about 1 ms latency. For IOPS and performance details, see Azure NetApp Files performance considerations and the FAQ .	Standard HDD: up to 500 IOPS per-disk limits Standard SSD: up to 4k IOPS per-disk limits Premium SSD: up to 20k IOPS per-disk limits We recommend Premium disks for Storage Spaces Direct
Capacity	100 TiB per share, Up to 5 PiB per general purpose account	100 TiB per volume, up to 12.5 PiB per NetApp account	Maximum 32 TiB per disk
Required infrastructure	Minimum share size 1 GiB	Minimum capacity pool 2 TiB, min volume size 100 GiB	Two VMs on Azure IaaS (+ Cloud Witness) or at least three VMs without and costs for disks
Protocols	SMB 3.0/2.1, NFSv4.1 (preview), REST	NFSv3, NFSv4.1, SMB 3.x/2.x, dual-protocol	NFSv3, NFSv4.1, SMB 3.1

Azure management details

Features	Azure Files	Azure NetApp Files	Storage Spaces Direct
Access	Cloud, on-premises and hybrid (Azure file sync)	Cloud, on-premises	Cloud, on-premises
Backup	Azure backup snapshot integration	Azure NetApp Files snapshots Azure NetApp Files backup	Azure backup snapshot integration
Security and compliance	All Azure supported certificates	Azure supported certificates	All Azure supported certificates

Features	Azure Files	Azure NetApp Files	Storage Spaces Direct
Azure Active Directory integration	Native Active Directory and Azure Active Directory Domain Services	Azure Active Directory Domain Services and Native Active Directory	Native Active Directory or Azure Active Directory Domain Services support only

Once you've chosen your storage method, check out [Azure Virtual Desktop pricing](#) for information about our pricing plans.

Azure Files tiers

Azure Files offers two different tiers of storage: premium and standard. These tiers let you tailor the performance and cost of your file shares to meet your scenario's requirements.

- Premium file shares are backed by solid-state drives (SSDs) and are deployed in the FileStorage storage account type. Premium file shares provide consistent high performance and low latency for input and output (IO) intensive workloads. Premium file shares use a provisioned billing model, where you pay for the amount of storage you would like your file share to have, regardless of how much you use.
- Standard file shares are backed by hard disk drives (HDDs) and are deployed in the general purpose version 2 (GPv2) storage account type. Standard file shares provide reliable performance for IO workloads that are less sensitive to performance variability, such as general-purpose file shares and dev/test environments. Standard file shares use a pay-as-you-go billing model, where you pay based on storage usage, including data stored and transactions.

To learn more about how billing works in Azure Files, see [Understand Azure Files billing](#).

The following table lists our recommendations for which performance tier to use based on your workload. These recommendations will help you select the performance tier that meets your performance targets, budget, and regional considerations. We've based these recommendations on the example scenarios from [Remote Desktop workload types](#).

Workload type	Recommended file tier
Light (fewer than 200 users)	Standard file shares
Light (more than 200 users)	Premium file shares or standard with multiple file shares
Medium	Premium file shares

Workload type	Recommended file tier
Heavy	Premium file shares
Power	Premium file shares

For more information about Azure Files performance, see [File share and file scale targets](#). For more information about pricing, see [Azure Files pricing](#).

Azure NetApp Files tiers

Azure NetApp Files volumes are organized in capacity pools. Volume performance is defined by the service level of the hosting capacity pool. Three performance levels are offered, ultra, premium and standard. For more information, see [Storage hierarchy of Azure NetApp Files](#). Azure NetApp Files performance is [a function of tier times capacity](#). More provisioned capacity leads to higher performance budget, which likely results in a lower tier requirement, providing a more optimal TCO.

The following table lists our recommendations for which performance tier to use based on workload defaults.

Workload	Example Users	Azure NetApp Files
Light	Users doing basic data entry tasks	Standard tier
Medium	Consultants and market researchers	Premium tier: small-medium user count Standard tier: large user count
Heavy	Software engineers, content creators	Premium tier: small-medium user count Standard tier: large user count
Power	Graphic designers, 3D model makers, machines learning researchers	Ultra tier: small user count Premium tier: medium user count Standard tier: large user count

In order to provision the optimal tier and volume size, consider using [this calculator](#) for guidance.

Next steps

To learn more about FSLogix profile containers, user profile disks, and other user profile technologies, see the table in [FSLogix profile containers and Azure Files](#).

If you're ready to create your own FSLogix profile containers, get started with one of these tutorials:

- [Set up FSLogix Profile Container with Azure Files and Active Directory](#)
- [Set up FSLogix Profile Container with Azure NetApp Files](#)

Azure Virtual Desktop for the enterprise

Microsoft Entra ID

Microsoft Entra

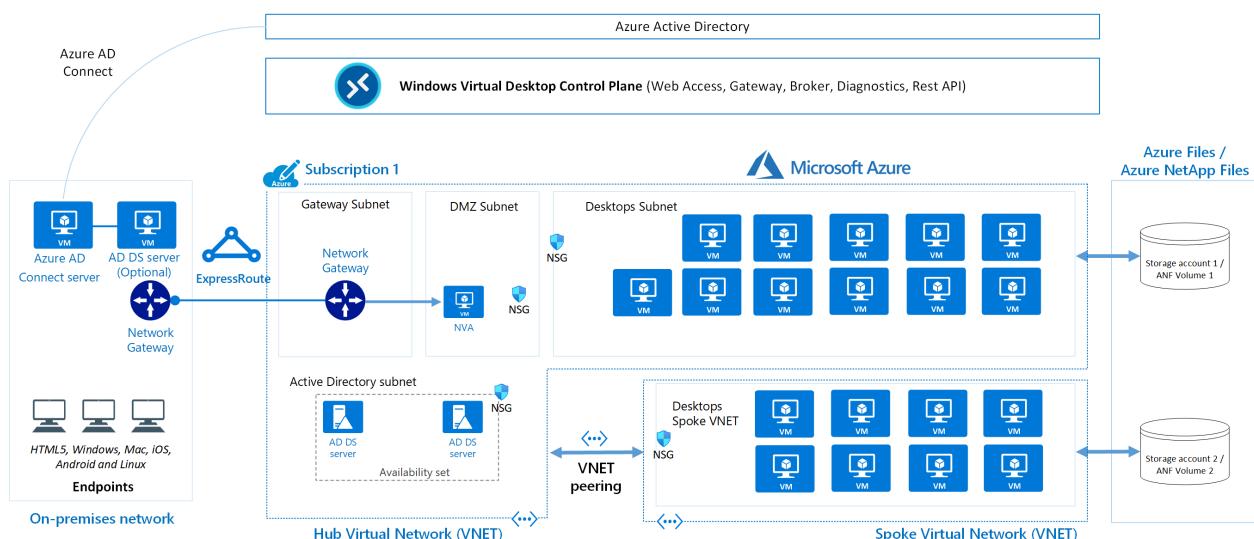
Azure Virtual Network

Azure Virtual Desktop

Azure Virtual Desktop [↗](#) is a desktop and application virtualization service that runs in Azure. This article is intended to help desktop infrastructure architects, cloud architects, desktop administrators, and system administrators explore Azure Virtual Desktop and build virtualized desktop infrastructure (VDI) solutions at enterprise scale. Enterprise-scale solutions generally cover 1,000 or more virtual desktops.

Architecture

A typical architectural setup for Azure Virtual Desktop is illustrated in the following diagram:



Download a [Visio file ↗](#) of this architecture.

Dataflow

The diagram's dataflow elements are described here:

- The application endpoints are in a customer's on-premises network. Azure ExpressRoute extends the on-premises network into Azure, and Microsoft Entra Connect integrates the customer's Active Directory Domain Services (AD DS) with Microsoft Entra ID.

- The Azure Virtual Desktop control plane handles web access, gateway, broker, diagnostics, and extensibility components such as REST APIs.
- The customer manages AD DS and Microsoft Entra ID, Azure subscriptions, virtual networks, [Azure Files or Azure NetApp Files](#), and the Azure Virtual Desktop host pools and workspaces.
- To increase capacity, the customer uses two Azure subscriptions in a hub-spoke architecture and connects them via virtual network peering.

For more information about FSLogix Profile Container - Azure Files and Azure NetApp Files best practices, see [FSLogix configuration examples](#).

Components

Azure Virtual Desktop service architecture is similar to [Windows Server Remote Desktop Services](#). Although Microsoft manages the infrastructure and brokering components, enterprise customers manage their own desktop host virtual machines (VMs), data, and clients.

Components that Microsoft manages

Microsoft manages the following Azure Virtual Desktop services, as part of Azure:

- **Web Access:** By using the [Web Access](#) service within Azure Virtual Desktop you can access virtual desktops and remote apps through an HTML5-compatible web browser just as you would with a local PC, from anywhere and on any device. You can secure web access by using multifactor authentication in Microsoft Entra ID.
- **Gateway:** The Remote Connection Gateway service connects remote users to Azure Virtual Desktop apps and desktops from any internet-connected device that can run an Azure Virtual Desktop client. The client connects to a gateway, which then orchestrates a connection from a VM back to the same gateway.
- **Connection Broker:** The Connection Broker service manages user connections to virtual desktops and remote apps. Connection Broker provides load balancing and reconnection to existing sessions.
- **Diagnostics:** Remote Desktop Diagnostics is an event-based aggregator that marks each user or administrator action on the Azure Virtual Desktop deployment as a success or failure. Administrators can query the event aggregation to identify failing components.

- **Extensibility components:** Azure Virtual Desktop includes several extensibility components. You can manage Azure Virtual Desktop by using Windows PowerShell or with the provided REST APIs, which also enable support from third-party tools.

Components that you manage

You manage the following components of Azure Virtual Desktop solutions:

- **Azure Virtual Network:** With [Azure Virtual Network](#), Azure resources such as VMs can communicate privately with each other and with the internet. By connecting Azure Virtual Desktop host pools to an Active Directory domain, you can define network topology to access virtual desktops and virtual apps from the intranet or internet, based on organizational policy. You can connect an Azure Virtual Desktop instance to an on-premises network by using a virtual private network (VPN), or you can use [Azure ExpressRoute](#) to extend the on-premises network into Azure over a private connection.
- **Microsoft Entra ID:** Azure Virtual Desktop uses [Microsoft Entra ID](#) for identity and access management. Microsoft Entra integration applies Microsoft Entra security features, such as conditional access, multifactor authentication, and [Intelligent Security Graph](#), and it helps maintain app compatibility in domain-joined VMs.
- **Active Directory Domain Services (Optional):** Azure Virtual Desktop VMs can either be domain joined to an [AD DS](#) service or use [Deploy Microsoft Entra joined virtual machines in Azure Virtual Desktop](#)
 - When using an AD DS domain, the domain must be in sync with Microsoft Entra ID to associate users between the two services. You can use [Microsoft Entra Connect](#) to associate AD DS with Microsoft Entra ID.
 - When using Microsoft Entra join, review the [supported configurations](#) to ensure your scenario is supported.
- **Azure Virtual Desktop session hosts:** Session hosts are VMs that users connect to for their desktops and applications. Several versions of Windows are supported and you can create images with your applications and customizations. You can choose VM sizes, including GPU-enabled VMs. Each session host has an Azure Virtual Desktop host agent, which registers the VM as part of the Azure Virtual Desktop workspace or tenant. Each host pool can have one or more app groups, which are collections of remote applications or desktop sessions that you can access. To see which versions of Windows are supported, see [Operating systems and licenses](#).

- **Azure Virtual Desktop workspace:** The Azure Virtual Desktop workspace or tenant is a management construct for managing and publishing host pool resources.

Scenario details

Potential use cases

The greatest demand for enterprise virtual desktop solutions comes from:

- Security and regulation applications, such as financial services, healthcare, and government.
- Elastic workforce needs, such as remote work, mergers and acquisitions, short-term employees, contractors, and partner access.
- Specific employees, such as bring your own device (BYOD) and mobile users, call centers, and branch workers.
- Specialized workloads, such as design and engineering, legacy apps, and software development testing.

Personal and pooled desktops

By using personal desktop solutions, sometimes called *persistent desktops*, users can always connect to the same specific session host. Users can ordinarily modify their desktop experience to meet personal preferences, and they can save files in the desktop environment. Personal desktop solutions:

- Let users customize their desktop environment, including user-installed applications, and users can save files within the desktop environment.
- Allow assigning dedicated resources to specific users, which can be helpful for some manufacturing or development use cases.

Pooled desktop solutions, also called *non-persistent desktops*, assign users to whichever session host is currently available, depending on the load-balancing algorithm. Because users don't always return to the same session host each time they connect, they have limited ability to customize the desktop environment and don't usually have administrator access.

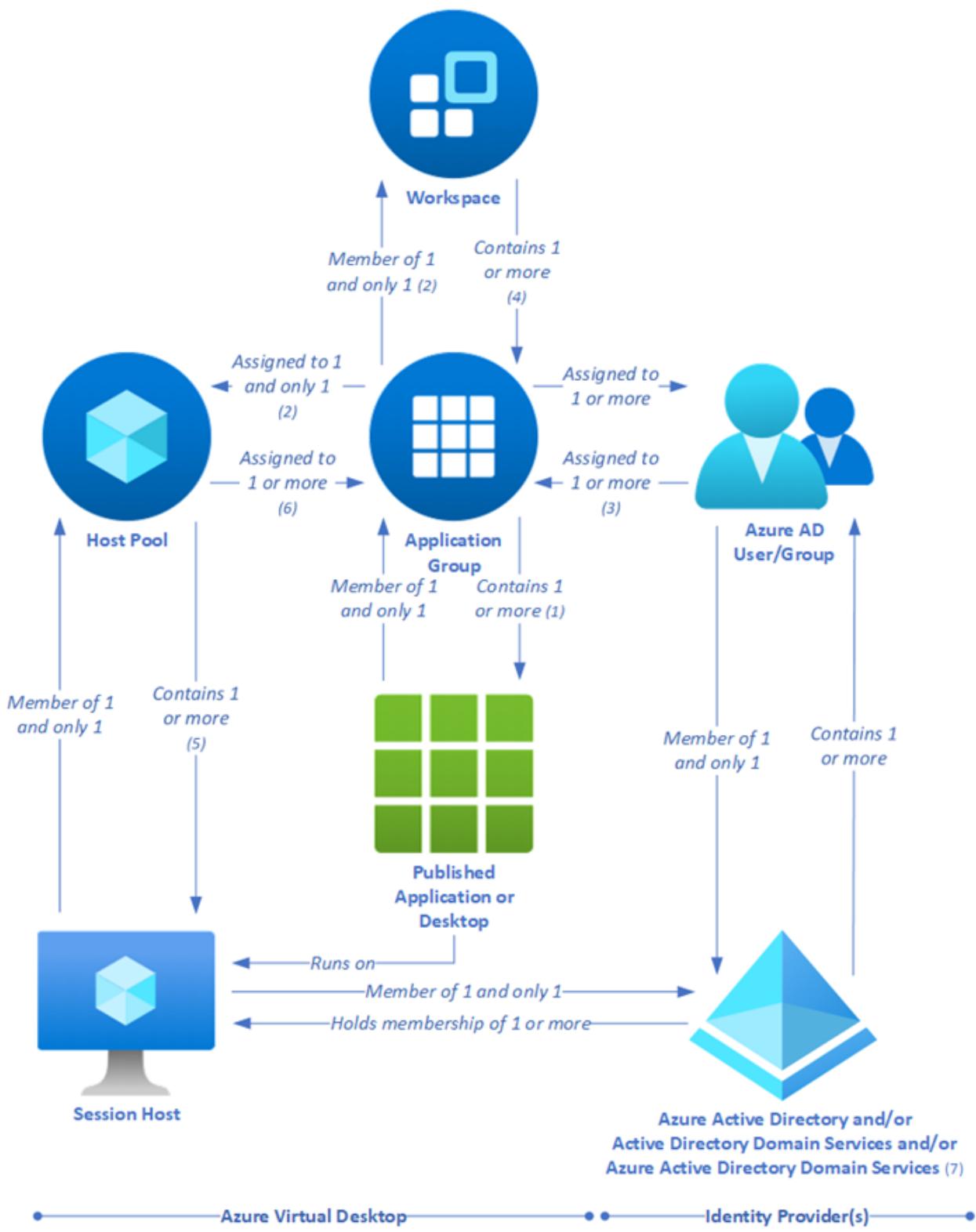
Windows servicing

There are several options for updating Azure Virtual Desktop instances. Deploying an updated image every month guarantees compliance and state.

- [Microsoft Endpoint Configuration Manager \(MECM\)](#) updates server and desktop operating systems.
- [Windows Updates for Business](#) updates desktop operating systems such as Windows 10 multi-session.
- [Azure Update Management](#) updates server operating systems.
- [Azure Log Analytics](#) checks compliance.
- Deploy a new (custom) image to session hosts every month for the latest Windows and applications updates. You can use an image from Azure Marketplace or a [custom Azure-managed image](#).

Relationships between key logical components

The relationships between host pools, workspaces, and other key logical components vary. They're summarized in the following diagram:



The numbers in the following descriptions correspond to those in the preceding diagram.

- (1) An application group that contains a published desktop can only contain MSIX packages mounted to the host pool (the packages will be available in the *Start* menu of the session host), it can't contain any other published resources and is called a desktop application group.
- (2) Application groups assigned to the same host pool must be members of the same workspace.

- (3) A user account can be assigned to an application group either directly or via a Microsoft Entra group. It's possible to assign no users to an application group, but then it can't service any.
- (4) It's possible to have an empty workspace, but it can't service users.
- (5) It's possible to have an empty host pool, but it can't service users.
- (6) It's possible for a host pool not to have any application groups assigned to it but it can't service users.
- (7) Microsoft Entra ID is required for Azure Virtual Desktop. This is because Microsoft Entra user accounts and groups must always be used to assign users to Azure Virtual Desktop application groups. Microsoft Entra ID is also used to authenticate users into the Azure Virtual Desktop service. Azure Virtual Desktop session hosts can also be members of a Microsoft Entra domain, and in this situation the Azure Virtual Desktop-published applications and desktop sessions will also be launched and run (not just assigned) by using Microsoft Entra accounts.
 - (7) Alternatively, Azure Virtual Desktop session hosts can be members of an AD DS domain, and in this situation the Azure Virtual Desktop-published applications and desktop sessions will be launched and run (but not assigned) by using AD DS accounts. To reduce user and administrative overhead, AD DS can be synchronized with Microsoft Entra ID through Microsoft Entra Connect.
 - (7) Finally, Azure Virtual Desktop session hosts can, instead, be members of a Microsoft Entra Domain Services domain, and in this situation the Azure Virtual Desktop-published applications and desktop sessions will be launched and run (but not assigned) by using Microsoft Entra Domain Services accounts. Microsoft Entra ID is automatically synchronized with Microsoft Entra Domain Services, one way, from Microsoft Entra ID to Microsoft Entra Domain Services only.

[] [Expand table](#)

Resource	Purpose	Logical relationships
Published desktop	A Windows desktop environment that runs on Azure Virtual Desktop session hosts and is delivered to users over the network	Member of one and only one application group (1)
Published application	A Windows application that runs on Azure Virtual	Member of one and only one application group

Resource	Purpose	Logical relationships
Application group	A logical grouping of published applications or a published desktop	<ul style="list-style-type: none"> - Contains a published desktop (1) or one or more published applications - Assigned to one and only one host pool (2) - Member of one and only one workspace (2) - One or more Microsoft Entra user accounts or groups are assigned to it (3)
Microsoft Entra user account/group	Identifies the users who are permitted to launch published desktops or applications	<ul style="list-style-type: none"> - Member of one and only one Microsoft Entra ID - Assigned to one or more application groups (3)
Microsoft Entra ID (7)	Identity provider	<ul style="list-style-type: none"> - Contains one or more user accounts or groups, which must be used to assign users to application groups, and can also be used to log in to the session hosts - Can hold the memberships of the session hosts - Can be synchronized with AD DS or Microsoft Entra Domain Services
AD DS (7)	Identity and directory services provider	<ul style="list-style-type: none"> - Contains one or more user accounts or groups, which can be used to log in to the session hosts - Can hold the memberships of the session hosts - Can be synchronized with Microsoft Entra ID
Microsoft Entra Domain Services (7)	Platform as a service (PaaS)-based identity and directory services provider	<ul style="list-style-type: none"> - Contains one or more user accounts or groups, which can be used to log in to the session hosts - Can hold the memberships of the session hosts

Resource	Purpose	Logical relationships
		- Synchronized with Microsoft Entra ID
Workspace	A logical grouping of application groups	Contains one or more application groups (4)
Host pool	A group of identical session hosts that serve a common purpose	- Contains one or more session hosts (5) - One or more application groups are assigned to it (6)
Session host	A virtual machine that hosts published desktops or applications	Member of one and only one host pool

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

The numbers in the following sections are approximate. They're based on a variety of large customer deployments and are subject to change over time.

Also, note that:

- You can't create more than 500 application groups per single Microsoft Entra tenant*.
- We recommend that you do *not* publish more than 50 applications per application group.

Azure Virtual Desktop limitations

Azure Virtual Desktop, much like Azure, has certain service limitations that you need to be aware of. To avoid having to make changes in the scaling phase, it's a good idea to address some of these limitations during the design phase.

Expand table

Azure Virtual Desktop object	Per Parent container object	Service limit
Workspace	Microsoft Entra tenant	1300
HostPool	Workspace	400
Application group	Microsoft Entra tenant	500*
RemoteApp	Application group	500
Role assignment	Any Azure Virtual Desktop object	200
Session host	HostPool	10,000

*If you require more than 500 application groups, submit a support ticket via the Azure portal.

- We recommend that you deploy no more than 5,000 VMs per Azure subscription per region. This recommendation applies to both personal and pooled host pools, based on Windows Enterprise single and multi-session. Most customers use Windows Enterprise multi-session, which allows multiple users to log in to each VM. You can increase the resources of individual session-host VMs to accommodate more user sessions.
- For automated session-host scaling tools, the limits are around 2,500 VMs per Azure subscription per region, because VM status interaction consumes more resources.
- To manage enterprise environments with more than 5,000 VMs per Azure subscription in the same region, you can create multiple Azure subscriptions in a hub-spoke architecture and connect them via virtual network peering (using one subscription per spoke). You could also deploy VMs in a different region in the same subscription to increase the number of VMs.
- Azure Resource Manager (ARM) subscription API throttling limits don't allow more than 600 Azure VM reboots per hour via the Azure portal. You can reboot all your machines at once via the operating system, which doesn't consume any Azure Resource Manager subscription API calls. For more information about counting and troubleshooting throttling limits based on your Azure subscription, see [Troubleshoot API throttling errors](#).
- You can currently deploy up to 132 VMs in a single ARM template deployment in the Azure Virtual Desktop portal. To create more than 132 VMs, run the ARM

template deployment in the Azure Virtual Desktop portal multiple times.

- Azure VM session-host name prefixes can't exceed 11 characters, due to auto-assigning of instance names and the NetBIOS limit of 15 characters per computer account.
- By default, you can deploy up to 800 instances of most resource types in a resource group. Azure Compute doesn't have this limit.

For more information about Azure subscription limitations, see [Azure subscription and service limits, quotas, and constraints](#).

VM sizing

[Virtual machine sizing guidelines](#) lists the maximum suggested number of users per virtual central processing unit (vCPU) and minimum VM configurations for different workloads. This data helps estimate the VMs you need in your host pool.

Use simulation tools to test deployments with both stress tests and real-life usage simulations. Make sure that the system is responsive and resilient enough to meet user needs, and remember to vary the load sizes when testing.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

You can architect your Azure Virtual Desktop solution to realize cost savings. Here are five different options to help manage costs for enterprises:

- **Windows 10 multi-session:** By delivering a multi-session desktop experience for users with identical compute requirements, you can let more users log in to a single VM at once, an approach that can result in considerable cost savings.
- **Azure Hybrid Benefit:** If you have Software Assurance, you can use [Azure Hybrid Benefit for Windows Server](#) to save on the cost of your Azure infrastructure.
- **Azure Reserved VM Instances:** You can prepay for your VM usage and save money. Combine [Azure Reserved VM Instances](#) with Azure Hybrid Benefit for up to 80 percent savings over list prices.
- **Session-host load-balancing:** When you're setting up session hosts, *breadth-first* mode, which spreads users randomly across the session hosts, is the standard default mode. Alternatively, you can use *depth-first* mode to fill up a session-host server with the maximum number of users before it moves on to the next session host. You can adjust this setting for maximum cost benefits.

Deploy this scenario

Use the [ARM templates](#) to automate the deployment of your Azure Virtual Desktop environment. These ARM templates support only Azure Resource Manager's Azure Virtual Desktop objects. These ARM templates don't support Azure Virtual Desktop (classic).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Tom Hickling](#) | Senior Product Manager, Azure Virtual Desktop Engineering

Other contributor:

- [Nelson Del Villar](#) | Cloud Solution Architect, Azure Core Infrastructure

Next steps

- [Azure Virtual Desktop partner integrations](#) lists approved Azure Virtual Desktop partner providers and independent software vendors.
- Use the [Virtual Desktop Optimization Tool](#) to help optimize performance in a Windows 10 Enterprise VDI (virtual desktop infrastructure) environment.
- See [Deploy Microsoft Entra joined virtual machines in Azure Virtual Desktop](#).
- Learn more about [Active Directory Domain Services](#).
- [What is Microsoft Entra Connect?](#)

Related resources

- For more information about multiple Active Directory forests architecture, see [Multiple Active Directory forests architecture in Azure Virtual Desktop](#).

Deploy Esri ArcGIS Pro in Azure Virtual Desktop

Azure Virtual Desktop

Azure NetApp Files

Azure Monitor

Azure Policy

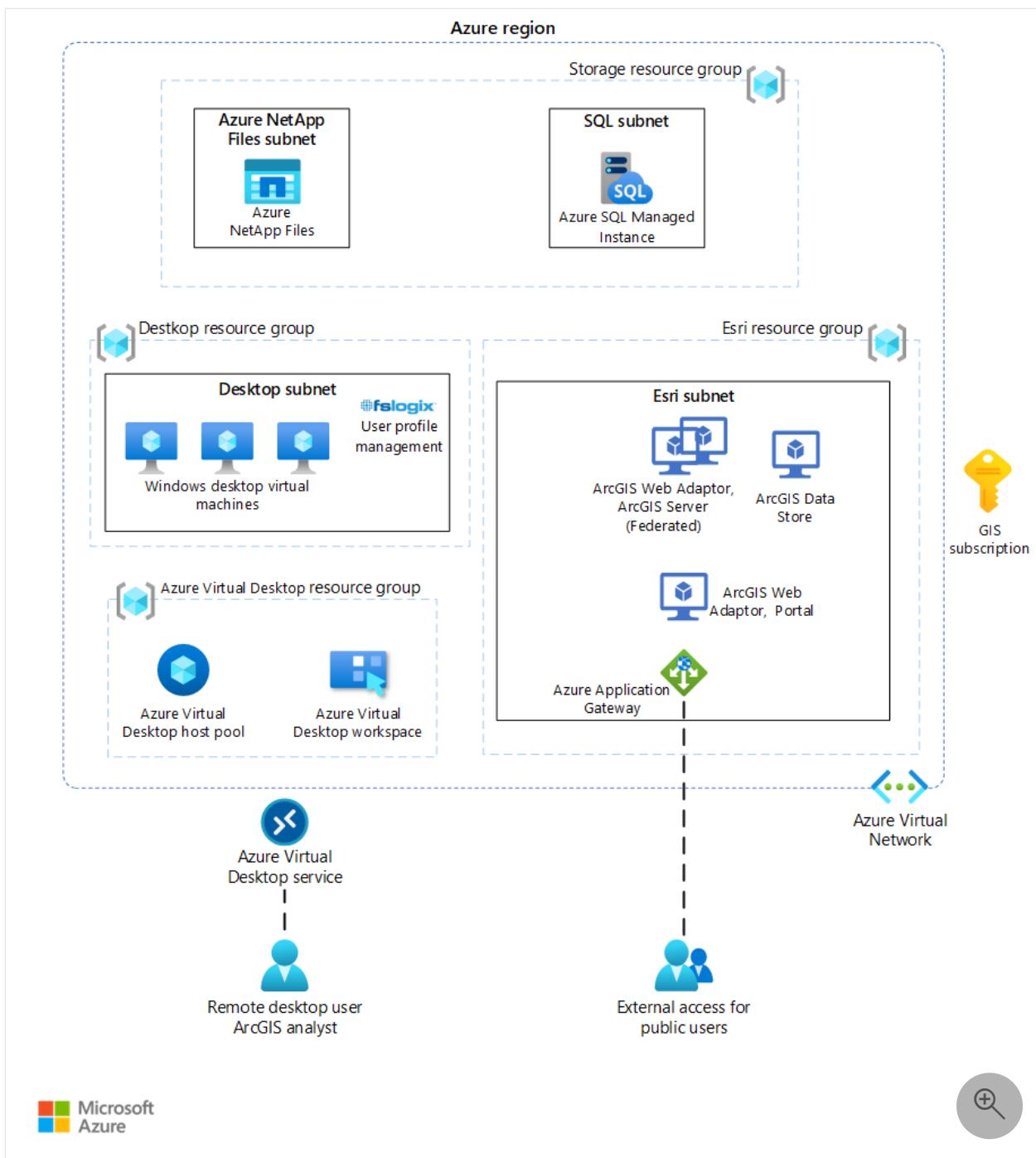
Microsoft Entra ID

This architecture shows how you can deploy Esri ArcGIS Pro in Azure Virtual Desktop to support the hyperscale of Azure. The architecture also includes back-end components like ArcGIS Enterprise to build a complete system on Azure.

ArcGIS® is a trademark of its company. No endorsement is implied by the use of this mark.

Architecture

The following diagram presents a high-level architecture for deploying ArcGIS components on Azure.



Download a [Visio file](#) of this architecture.

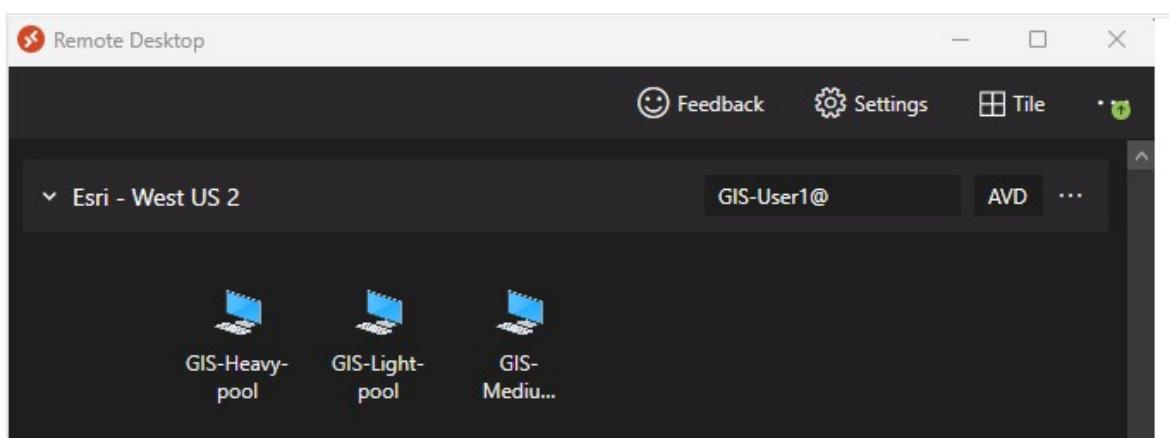
Workflow

- This solution is deployed to a single region with storage, GIS desktop, GIS back end, and Azure Virtual Desktop resource groups. Each resource group contains one subnet, and all subnets are in one virtual network. All components are in a single Azure subscription. This architecture is a three-tier deployment.
- The application endpoints are in the on-premises network.
- The Azure Virtual Desktop control plane handles web access, gateway, broker, diagnostics, and extensibility components like REST APIs.

- You manage Microsoft Entra Domain Services and Microsoft Entra ID, Azure subscriptions, virtual networks, [Azure Files or Azure NetApp Files](#), and the Azure Virtual Desktop host pools and workspaces.
- GIS analysts, administrators, and editors connect to Azure Virtual Desktop via a Remote Desktop Protocol (RDP) session. From there, ArcGIS Pro is accessed and takes advantage of the GPUs for light, medium, and heavy workflows. *Light* refers to a 2D workflow, *medium* refers to a more demanding 2D workflow, and *heavy* refers to a 2D or 3D workflow that requires GPUs. GIS administrators can also use ArcGIS Pro to publish services and administer the enterprise geodatabase. Finally, GIS editors can maintain the vector and raster layers.
- The desktop VMs are based on [N-Series VMs](#). Example VM SKUs for ArcGIS Pro:
 - Heavy: Standard_NV16as_v4. 16 CPU, 56 GB.
 - Medium: Standard_NV8as_v4. 8 CPU, 28 GB.
 - Light: Standard_NV4as_v4. 4 CPU, 14 GB.

For details on the VMs, see [NV-Series](#). The preceding groupings allow administrators to size the VMs based on workflows as opposed to VM capabilities. For example, end users see **GIS-Heavy-pool**, which indicates that the VM is for power users that need 3D-intensive workflows.

Administrators can also make it possible to publish new versions of ArcGIS Pro by using semantic versioning. For example, as new versions of ArcGIS Pro are available, like ArcGIS Pro 3.0, the new version can be published in the Remote Desktop tool. As a result, users can pick that new version when they're ready to upgrade without having to perform the upgrade themselves. The GPU drivers can be included in the creation of the images that the deployments are based on.



- Web GIS users can also take advantage of this solution by accessing ArcGIS Enterprise administrative interfaces either in the browser in the Azure Virtual Desktop RDP session or via their local browser (if ArcGIS is published as public facing). The Azure application gateway routes the traffic to the correct endpoint for

the ArcGIS server roles. As with ArcGIS Pro, the latency between the browsers and the back end are minimized.

- You can deploy the enterprise geodatabase in Azure SQL Managed Instance. ArcGIS Pro users can then create, manage, and edit the geodatabase from an RDP session. During the creation of the Azure Virtual Desktop image, administrators can include the ODBC drivers so users don't have to install them on the Azure Virtual Desktop VMs.
- Azure NetApp Files supports fast access to the ArcGIS Server configuration store and directories. You can use Azure Files and Azure Storage, but Azure NetApp Files costs less for large deployments. Additionally, you can use Azure NetApp Files to store Portal for ArcGIS items and raster images, lidar data, and so on.

Components

- [Azure NetApp Files](#) is an enterprise-class, high-performance, metered file Network Attached Storage (NAS) service.
- [Azure Monitor](#) is a collection of tools that provides visibility into the state of your system. It helps you understand how your cloud-native services are performing and proactively identifies problems that affect them.
- [Azure Policy](#) helps you enforce organizational standards and assess compliance at scale. Through its compliance dashboard, it provides an aggregated view of the overall state of the environment and the ability to drill down to per-resource, per-policy granularity. It also helps you bring your resources to compliance via bulk remediation for existing resources and automatic remediation for new resources.
- [Microsoft Entra ID](#) enterprise identity service provides single sign-on, multifactor authentication, and conditional access to guard against 99.9 percent of cybersecurity attacks.
- [Active Directory Domain Services \(AD DS\)](#) enables you to store directory data and make that data available to network users and administrators. AD DS stores information about user accounts, like names, passwords, and phone numbers, and enables other authorized users on the same network to access that information. This data store, also known as the *directory*, contains information about Active Directory objects. These objects typically include shared resources like servers, volumes, printers, and the network user and computer accounts.

Security is integrated with Active Directory through sign-in authentication and controlled access to objects in the directory. With a single network sign-in,

administrators can manage directory data and organization throughout their network, and authorized network users can access resources anywhere on the network.

- [Azure Virtual Desktop](#) is a desktop and application virtualization service that runs on Azure. This service is free and managed by Microsoft as a platform as a service (PaaS) offering, saving you money on licensing and infrastructure costs. It's a flexible cloud virtual desktop infrastructure (VDI) platform that delivers virtual desktops and remote apps with maximum control and improved security.
- [Azure SQL Managed Instance](#) is a PaaS version of SQL Server. It's an intelligent and scalable relational database service.
- [Azure Application Gateway](#) is an application delivery controller-as-a-service offering that provides layer-7 load balancing, security, and web application firewall functionality.
- [FSLogix](#) enhances and enables user profile management for Windows remote computing environments. It allows users to roam between remote computing session hosts, minimize sign-in times for virtual desktop environments, and optimize file I/O between the host/client and the remote profile store.

For information about FSLogix Profile Container, Azure Files, and Azure NetApp Files best practices, see [FSLogix configuration examples](#).

- [Azure Virtual Network](#) enables you to create your own private network infrastructure in the cloud.
- [ArcGIS Pro](#) is Esri's professional desktop GIS application. It enables power users to explore, geovisualize, and analyze data. It includes 2D and 3D capabilities and runs best on Azure high performance computing VMs, like those in the NV-Series. You can scale the use of ArcGIS by using Azure Virtual Desktop.
- [ArcGIS Enterprise](#) is a platform for mapping and geovisualization, analytics, and data management that hosts data, applications, and custom low-code or no-code applications. It works with ArcGIS Pro or ArcGIS Desktop (not included here because it has been replaced by ArcGIS Pro). ArcGIS Enterprise isn't part of this reference architecture, but you can extend the architecture to include it.
- [Portal for ArcGIS](#) is part of the base deployment. It provides the ability to share maps, scenes, apps, and other geospatial information within an organization. With this front-end interface, anyone in the organization can make a map, find layers, and perform queries with very little training.

- [ArcGIS Server](#) is back-end server software that's deployed with ArcGIS Enterprise or in a standalone deployment with ArcGIS Enterprise. ArcGIS Server receives requests from clients to draw maps, run tools, query data, and so on. It also has a management plane that enables administrators to start, stop, and delete services.
- [ArcGIS Server configuration store](#) contains system configuration information so that, as ArcGIS Server scales to other machines, it can share that information.
- [Enterprise geodatabase](#) is a geospatial database designed to host vector and raster data. It can be deployed in many database management systems. In this architecture, the enterprise geodatabase is stored in Azure SQL Managed Instance.

Alternatives

- You can use [ArcGIS Enterprise Builder](#) to set up a base ArcGIS Enterprise deployment on a single machine or multiple machines.
- Although Azure Files and Azure Blob Storage are fine for many enterprises, Azure NetApp Files might be better suited for GIS because of large raster image files, Portal for ArcGIS items, shapefiles, lidar datasets, file geodatabases, and other geospatial data types that require fast access.
- You can add other ArcGIS Enterprise server roles, like Raster Analytics Server, GeoAnalytics Server, GeoEvent Server, Knowledge Server, and Mission Server, to this base deployment as needed. You can also use newer technologies, like ArcGIS Enterprise on Kubernetes, as a replacement for or supplement to ArcGIS Enterprise. GPU-based VMs for Drone2Map, CityEngine, and SURE for ArcGIS can also take advantage of these VMs. For more information, see [ArcGIS Enterprise server roles](#).
- To increase capacity, you can use multiple Azure subscriptions in a hub-and-spoke architecture and connect them via virtual network peering. Also, you can use Azure landing zones to lay down the initial services. For more information, see [What is an Azure landing zone?](#).

Scenario details

Esri's technology is a geographic information system (GIS) that contains capabilities for the visualization, analysis, and data management of geospatial data. Esri's core technology is called *the ArcGIS platform*. It includes capabilities for mapping, spatial analysis, 3D GIS, imagery and remote sensing, data collection and management, and field operations. For more information, see the [ArcGIS page](#) on the Esri website.

A desktop app called *ArcGIS Pro* is a key part of the technology. It's a 64-bit professional desktop GIS. GIS analysts can use it to perform spatial analysis and edit spatial data. GIS administrators can use it to create and publish geospatial services.

Potential use cases

Esri's ArcGIS and virtual desktop solutions are frequently used for:

- Security and regulation applications like utilities (energy), healthcare, and government.
- Elastic workforce needs like remote work, mergers and acquisition, short-term employees, contractors, and partner access.
- Employees like bring your own device (BYOD) users, mobile users, and branch workers.
- Specialized workloads like land management (facilities and real estate), design and engineering, legacy apps, and software testing.

Although GIS has been implemented in Azure for many years, it has typically included only the back-end components. That implementation introduces latency between the client and server components. Organizations have been able to deploy desktop GIS on VMs from Azure Marketplace, but that deployment is for thick clients and isn't very scalable. This architecture addresses both challenges.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, a set of guiding tenets that you can use to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

Ideally, the latency between the end user and the RDP session needs to be around 200 ms or less. This latency helps to ensure that, when ArcGIS Pro users interact with maps and perform measurements or edits, the interactive edits and the tooltips appear quickly enough. The [Azure Virtual Desktop Experience Estimator](#) can provide a quick assessment of connection round-trip time (RTT) from your location, through the Azure

Virtual Desktop service, and to each Azure region in which you can deploy virtual machines.

When you use a remote Windows session, your network's available bandwidth greatly affects the quality of your experience. The following table lists the minimum recommended bandwidths for a smooth user experience. These recommendations are based on the guidelines in [Remote Desktop workloads](#).

Expand table

Workload type	Recommended bandwidth
Light	1.5 Mbps
Medium	3 Mbps
Heavy	5 Mbps
Power	15 Mbps

Keep in mind that the stress put on your network depends on both your app workload's output frame rate and your display resolution. If either the frame rate or display resolution increases, the bandwidth requirement also rises. For example, a light workload with a high-resolution display requires more available bandwidth than a light workload with regular or low resolution.

Ideally, all components in the preceding architecture diagram are deployed in a single region to minimize latency between components. However, for large organizations, a multi-region deployment is necessary and supported. Another component to consider is [Azure Front Door](#), which routes users to the closest region.

Another significant benefit of this architecture is that the latency between it and Esri's SaaS offerings, like ArcGIS Velocity and ArcGIS Image, is also reduced for ArcGIS Pro users and web browser users. All components of the ArcGIS platform are in the cloud.

Scalability

You can scale this architecture in many ways. You can scale the VMs for the back end or the desktops (both CPU and GPUs) in, out, up, or down. You can also deploy Azure Virtual Desktop on individual VMs or multi-session VMs. Azure Virtual Desktop can scale

hundreds or thousands of VMs. For more information, see [Windows 10 or Windows 11 Enterprise multi-session remote desktops](#).

Testing

You can test your system's latency by using the [Connection Experience Indicator](#). You can use [Esri's ArcGIS Pro Performance Assessment Tool](#) to test the performance. Esri also recommends [tools for testing ArcGIS Enterprise](#). [Azure Load Testing](#) can also be helpful.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Matt Hallenborg](#) | (Senior Cloud Solution Architect)
- [Ron Vincent](#) | (Senior Program Manager)

Other contributor:

- [Mick Alberts](#) | (Technical Writer)

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Create a managed image of a generalized VM in Azure](#)
- Prepare an Azure Virtual Desktop image with the [Virtual Desktop Optimization Tool \(VDOT\)](#)
- Download and install FSLogix
- Create a golden image in Azure
- Create an Azure Virtual Desktop host pool
- Create an Azure SQL Managed Instance
- Install ArcGIS Server
- Install Portal for ArcGIS
- Install NVIDIA GPU drivers on N-Series VMs running Windows
- Assess Azure SQL Managed Instance via SSMS
- Configure public endpoint in Azure SQL Managed Instance
- Connect to Microsoft SQL Server from ArcGIS
- Create Enterprise Geodatabase

- Best practices for tuning ArcGIS Enterprise ↗
- Configure highly available ArcGIS Enterprise ↗
- Esri GIS mapping software, location intelligence, and spatial analytics ↗

Related resources

- [Azure Virtual Desktop for the enterprise](#)
- [FSLogix configuration examples](#)
- [Multiple forests with AD DS and Microsoft Entra ID](#)

Multiple forests with AD DS and Microsoft Entra ID

Azure Virtual Desktop

Microsoft Entra ID

Microsoft Entra

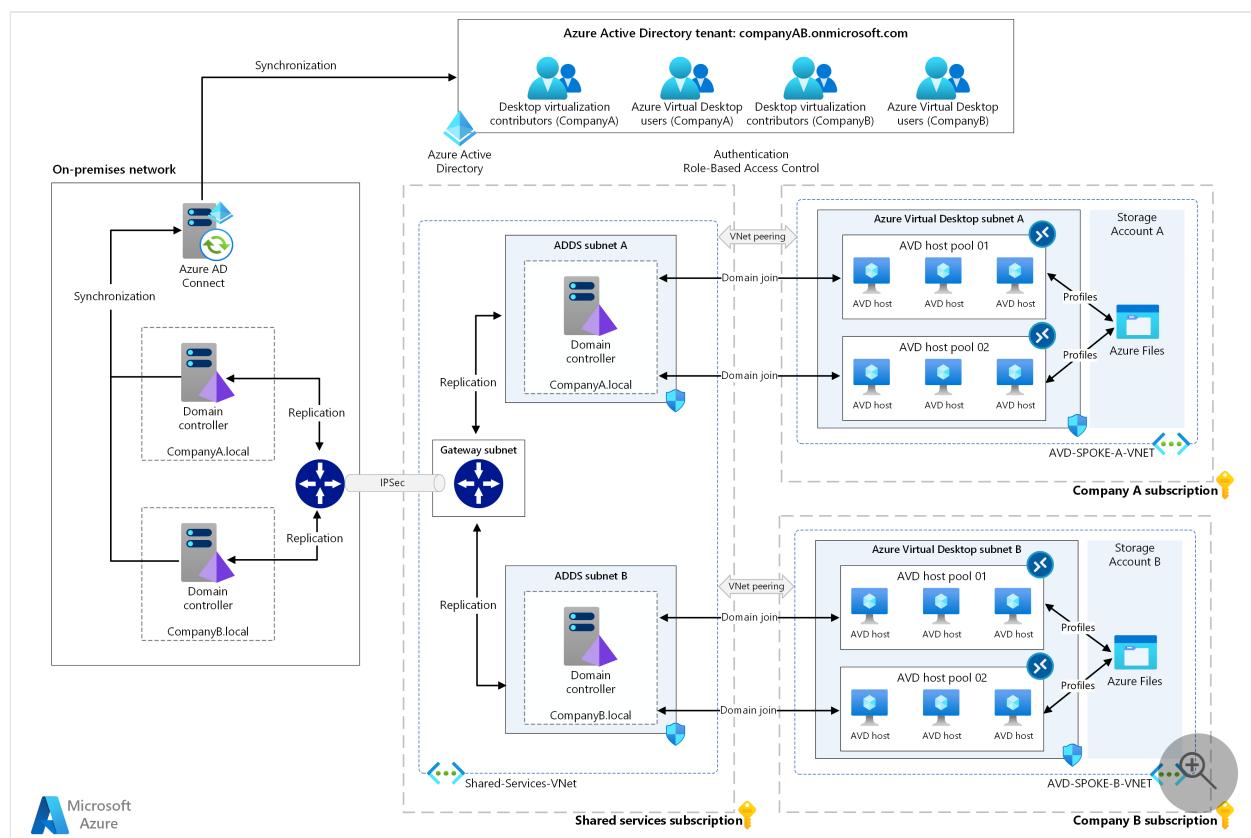
Azure ExpressRoute

Azure Storage

Many organizations want to take advantage of Azure Virtual Desktop to create environments that have multiple on-premises Active Directory forests.

This article expands on the architecture that's described in the [Azure Virtual Desktop at enterprise scale](#) article. It's intended to help you understand how to integrate multiple domains and Azure Virtual Desktop by using [Microsoft Entra Connect](#) to sync users from on-premises [Active Directory Domain Services \(AD DS\)](#) to Microsoft Entra ID.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

In this architecture, the identity flow works as follows:

1. Microsoft Entra Connect syncs users from both CompanyA.com and CompanyB.com to a Microsoft Entra tenant (NewCompanyAB.onmicrosoft.com).
2. Host pools, workspaces, and app groups are created in separate subscriptions and spoke virtual networks.
3. Users are assigned to the app groups.
4. Azure Virtual Desktop session hosts in the host pools join the domains CompanyA.com and CompanyB.com by using the domain controllers in Azure.
5. Users sign in by using either the [Azure Virtual Desktop application](#) or the [web client](#) with a User Principal Name (UPN) in the following format: user@NewCompanyA.com, user@CompanyB.com, or user@NewCompanyAB.com, depending on their configured UPN suffix.
6. Users are presented with their respective virtual desktops or applications. For example, users in CompanyA are presented with a virtual desktop or application in Workspace A, host pool 1 or 2.
7. FSLogix user profiles are created in Azure Files shares on the corresponding storage accounts.
8. Group Policy Objects (GPOs) that are synced from on-premises are applied to users and Azure Virtual Desktop session hosts.

Components

This architecture uses the same [components](#) as those listed in [Azure Virtual Desktop at enterprise scale architecture](#).

Additionally, this architecture uses the following components:

- **Microsoft Entra Connect in staging mode:** The [Staging server for Microsoft Entra Connect topologies](#) provides additional redundancy for the Microsoft Entra Connect instance.
- **Azure subscriptions, Azure Virtual Desktop workspaces, and host pools:** You can use multiple subscriptions, Azure Virtual Desktop workspaces, and host pools for administration boundaries and business requirements.

Scenario details

This architecture diagram represents a typical scenario that contains the following elements:

- The Microsoft Entra tenant is available for a new company named *NewCompanyAB.onmicrosoft.com*.

- Microsoft Entra Connect syncs users from on-premises AD DS to Microsoft Entra ID.
- Company A and Company B have separate Azure subscriptions. They also have a [shared services subscription](#), referred to as the *Subscription 1* in the diagram.
- An [Azure hub-spoke architecture](#) is implemented with a shared services hub virtual network.
- Complex hybrid on-premises Active Directory environments are present with two or more Active Directory forests. Domains live in separate forests, each with a different [UPN suffix](#). For example, *CompanyA.local* with the UPN suffix *CompanyA.com*, *CompanyB.local* with the UPN suffix *CompanyB.com*, and an additional UPN suffix, *NewCompanyAB.com*.
- Domain controllers for both forests are located on-premises and in Azure.
- Verified domains are present in Azure for *CompanyA.com*, *CompanyB.com*, and *NewCompanyAB.com*.
- GPO and legacy authentication, such as [Kerberos](#), [NTLM \(Windows New Technology LAN Manager\)](#), and [LDAP \(Lightweight Directory Access Protocol\)](#) , is used.
- For Azure environments that still have dependency on-premises infrastructure, private connectivity ([Site-to-site VPN or Azure ExpressRoute](#)) is set up between on-premises and Azure.
- The [Azure Virtual Desktop environment](#) consists of an Azure Virtual Desktop workspace for each business unit and two host pools per workspace.
- The Azure Virtual Desktop session hosts are joined to domain controllers in Azure. That is, CompanyA session hosts join the *CompanyA.local* domain, and CompanyB session hosts join the *CompanyB.local* domain.
- Azure storage accounts can use [Azure Files for FSLogix profiles](#). One account is created per company domain (that is, *CompanyA.local* and *CompanyB.local*), and the account is joined to the corresponding domain.

Note

Active Directory Domain Services is a self-managed, on-premises component in many hybrid environments, and Microsoft Entra Domain Services (Microsoft Entra Domain Services) provides managed domain services with a subset of fully compatible, traditional AD DS features such as domain join, group policy, LDAP, and Kerberos/NTLM authentication. For a detailed comparison of these components, see [Compare self-managed AD DS, Microsoft Entra ID, and managed Microsoft Entra Domain Services](#).

The solution idea [Multiple Azure Virtual Desktop forests using Microsoft Entra Domain Services](#) discusses architecture that uses cloud-managed Microsoft Entra

Potential use cases

Here are a few relevant use cases for this architecture:

- Mergers and acquisitions, organization rebranding, and multiple on-premises identities
- [Complex on-premises active directory environments \(multi-forest, multi-domains, group policy \(or GPO\) requirements, and legacy authentication\)](#)
- On-premises GPO infrastructure with Azure Virtual Desktop

Considerations

When you're designing your workload based on this architecture, keep the following ideas in mind.

Group Policy Objects

- To extend GPO infrastructure for Azure Virtual Desktop, the on-premises domain controllers should sync to the Azure infrastructure as a service (IaaS) domain controllers.
- Extending GPO infrastructure to Azure IaaS domain controllers requires private connectivity.

Network and connectivity

- The domain controllers are shared components, so they need to be deployed in a shared services hub virtual network in this [hub-spoke architecture](#).
- Azure Virtual Desktop session hosts join the domain controller in Azure over their respective hub-spoke virtual network peering.

Azure Storage

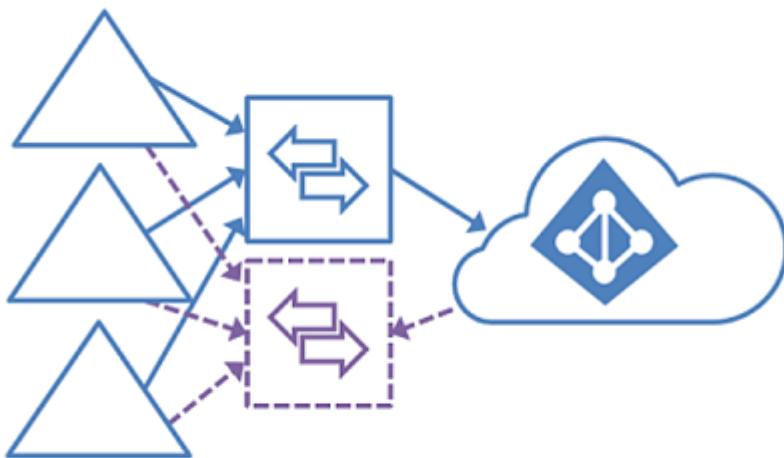
The following design considerations apply to user profile containers, cloud cache containers, and [MSIX](#) packages:

- You can use both [Azure Files](#) and [Azure NetApp Files](#) in this scenario. You choose the right solution based on factors such as expected performance, cost, and so on.

- Both Azure storage accounts and Azure NetApp Files are limited to joining to one single AD DS at a time. In these cases, multiple Azure storage accounts or Azure NetApp Files instances are required.

Microsoft Entra ID

In scenarios with users in multiple on-premises Active Directory forests, only one Microsoft Entra Connect Sync server is connected to the Microsoft Entra tenant. An exception to this is a Microsoft Entra Connect server that's used in staging mode.



The following identity topologies are supported:

- Multiple on-premises Active Directory forests.
- One or more resource forests trust all account forests.
- A full mesh topology allows users and resources to be in any forest. Commonly, there are two-way trusts between the forests.

For more details, see the [Staging server section of Microsoft Entra Connect topologies](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Tom Maher](#) | Senior Security and Identity Engineer

Next steps

For more information, see the following articles:

- [Microsoft Entra Connect topology](#)
- [Compare different identity options: Self-managed Active Directory Domain Services \(AD DS\), Microsoft Entra ID, and Microsoft Entra Domain Services \(Microsoft Entra Domain Services\)](#)
- [Azure Virtual Desktop documentation](#)

Related resources

- [Azure Virtual Desktop for the enterprise](#)
- [Solution idea: Multiple forests with Microsoft Entra Domain Services](#)

Multiple forests with AD DS, Microsoft Entra ID, and Microsoft Entra Domain Services

Microsoft Entra ID

Microsoft Entra

Azure Files

Azure Virtual Desktop

💡 Solution ideas

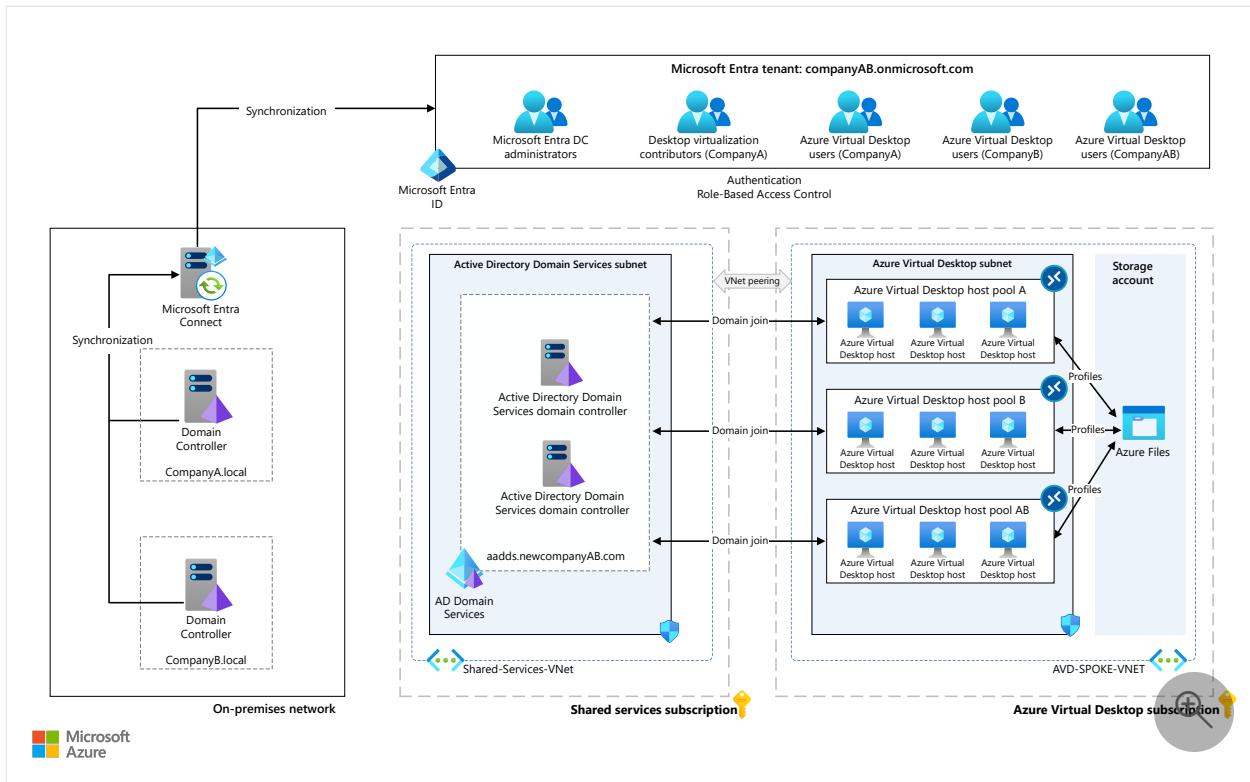
This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea illustrates how to deploy Azure Virtual Desktop rapidly in a *minimum viable product* (MVP) or a *proof of concept* (PoC) environment with the use of [Microsoft Entra Domain Services \(Microsoft Entra Domain Services\)](#). Use this idea to both extend on-premises multi-forest AD DS identities to Azure without private connectivity and support [legacy authentication](#).

Potential use cases

This solution idea also applies to mergers and acquisitions, organization rebranding, and multiple on-premises identities requirements.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The following steps show how the data flows in this architecture in the form of identity.

1. Complex hybrid on-premises Active Directory environments are present, with two or more Active Directory forests. Domains live in separate forests, with distinct User Principal Name (UPN) suffixes. For example, *CompanyA.local* with UPN suffix *CompanyA.com*, *CompanyB.local* with UPN suffix *CompanyB.com*, and an additional UPN suffix, *newcompanyAB.com*.
2. Instead of using customer-managed domain controllers, either on-premises or on Azure (that is, Azure infrastructure as a service [IaaS] domain controllers), the environment uses [the two cloud-managed domain controllers provided by Microsoft Entra Domain Services](#).
3. Microsoft Entra Connect syncs users from both *CompanyA.com* and *CompanyB.com* to the Microsoft Entra tenant, *newcompanyAB.onmicrosoft.com*. The user account is represented only once in Microsoft Entra ID, and private connectivity isn't used.
4. Users then sync from Microsoft Entra ID to the managed Microsoft Entra Domain Services as a one-way sync.
5. A custom and *routeable* Microsoft Entra Domain Services domain name, *aadds.newcompanyAB.com*, is created. The *newcompanyAB.com* domain is a registered domain that supports LDAP certificates. We generally recommend that

you *not* use non-routable domain names, such as contoso.local, because it can cause issues with DNS resolution.

6. The Azure Virtual Desktop session hosts join the Microsoft Entra Domain Services domain controllers.
7. Host pools and app groups can be created in a separate subscription and spoke virtual network.
8. Users are assigned to the app groups.
9. Users sign in by using either the [Azure Virtual Desktop application](#) or the [web client](#), with a UPN in a format such as john@companyA.com, jane@companyB.com, or joe@newcompanyAB.com, depending on their configured UPN suffix.
10. Users are presented with their respective virtual desktops or apps. For example, john@companyA.com is presented with virtual desktops or apps in host pool A, jane@companyB is presented with virtual desktops or apps in host pool B, and joe@newcompanyAB is presented with virtual desktops or apps in host pool AB.
11. The storage account (Azure Files is used for FSLogix) is joined to the managed domain AD DS. The FSLogix user profiles are created in Azure Files shares.

Note

- For Group Policy requirements in Microsoft Entra Domain Services, you can install [Group Policy Management tools](#) on a Windows Server virtual machine that's joined to Microsoft Entra Domain Services.
- To extend Group Policy infrastructure for Azure Virtual Desktop from the on-premises domain controllers, you need to manually export and import it to Microsoft Entra Domain Services.

Components

You implement this architecture by using the following technologies:

- [Microsoft Entra ID](#)
- [Microsoft Entra Domain Services](#)
- [Azure Files](#)
- [Azure Virtual Desktop](#)
- [Azure Virtual Network](#)

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Tom Maher](#) | Senior Security and Identity Engineer

Next steps

- [Multiple Active Directory forests architecture with Azure Virtual Desktop](#)
- [Azure Virtual Desktop for enterprises](#)
- [Microsoft Entra Connect topologies](#)
- [Compare different identity options](#)
- [Azure Virtual Desktop documentation](#)

Related resources

- [Hybrid architecture design](#)
- [Multiple forests with AD DS and Microsoft Entra ID](#)

Web applications architecture design

Article • 11/01/2023

Today's web apps are expected to be available all day, every day from anywhere in the world, and usable from virtually any device or screen size. Web applications must be secure, flexible, and scalable to meet spikes in demand.

This article provides an overview of Azure web app technologies, guidance, solution ideas, and reference architectures.

Azure provides a wide range of tools and capabilities for creating, hosting, and monitoring web apps. These are just some of the key web app services available in Azure:

- [Azure App Service](#)  enables you to easily create enterprise-ready web and mobile apps for any platform or device and deploy them on a scalable cloud infrastructure.
- [Azure Web Application Firewall](#)  provides powerful protection for web apps.
- [Azure Monitor](#)  provides full observability into your applications, infrastructure, and network. Monitor includes [Application Insights](#), which provides application performance management and monitoring for live web apps.
- [Azure SignalR Service](#)  enables you to easily add real-time web functionalities.
- [Static Web Apps](#)  provides streamlined full-stack development, from source code to global high availability.
- [Web App for Containers](#)  enables you to run containerized web apps on Windows and Linux.
- [Azure Service Bus](#)  enables you to integrate with other web apps using loosely coupled event-driven patterns.

Introduction to web apps on Azure

If you're new to creating and hosting web apps on Azure, the best way to learn more is with [Microsoft Learn training](#). This free online platform provides interactive training for Microsoft products and more.

These are a few good starting points to consider:

- [Create Azure App Service web apps](#)
- [Deploy and run a containerized web app with Azure App Service](#)
- [Azure Static Web Apps](#)

Path to production

Consider these patterns, guidelines, and architectures as you plan and implement your deployment:

- [Basic web application](#)
- [Baseline zone-redundant web application](#)
- [Multi-region active-passive web application](#)
- [Common web application architectures](#)
- [Design principles for Azure applications](#)
- [Design and implementation patterns - Cloud Design Patterns](#)
- [Enterprise deployment using App Services Environment](#)
- [High availability enterprise deployment using App Services Environment](#)

Best practices

For a good overview, see [Characteristics of modern web applications](#).

For information specific to Azure App Service, see:

- [Azure App Service and operational excellence](#)
- [App Service deployment best practices](#)
- [Security recommendations for App Service](#)
- [Azure security baseline for App Service](#)

Web app architectures

The following sections, organized by category, provide links to sample web app architectures.

E-commerce

- [E-commerce front end](#)
- [Intelligent product search engine for e-commerce](#)
- [Scalable order processing](#)
- [E-commerce website running in secured App Service Environment](#)
- [Scalable e-commerce web app](#)
- [Scalable Episerver marketing website](#)
- [Scalable Sitecore marketing website](#)

Healthcare

- Clinical insights with Microsoft Cloud for Healthcare
- Consumer health portal on Azure
- Virtual health on Microsoft Cloud for Healthcare

Modernization

- Choose between traditional web apps and single-page apps
- ASP.NET architectural principles
- Common client-side web technologies
- Development process for Azure
- Azure hosting recommendations for ASP.NET Core web apps

Multi-tier apps

- Multi-tier web application built for HA/DR

Multi-region apps

- Highly available multi-region web application

Scalability

- Scalable and secure WordPress on Azure
- Baseline web application with zone redundancy
- Scalable Umbraco CMS web app

Security

- Improved-security access to multitenant web apps from an on-premises network
- Protect APIs with Application Gateway and API Management

SharePoint

- Highly available SharePoint farm
- Hybrid SharePoint farm with Microsoft 365

Stay current with web development

Get the latest [updates on Azure web app products and features](#).

Additional resources

Example solutions

Here are some additional implementations to consider:

- [Eventual consistency between multiple Power Apps instances](#)
- [App Service networking features](#)
- [Migrate a web app using Azure APIM](#)
- [Sharing location in real time using low-cost serverless Azure services](#)
- [Serverless web application](#)
- [Web application monitoring on Azure](#)

AWS or Google Cloud professionals

- [AWS to Azure services comparison - Web applications](#)
- [Google Cloud to Azure services comparison - Application services](#)

Industry solutions with Azure

Article • 12/16/2022

The digital landscape is being transformed by unprecedented amounts of data across industries. Start here to learn about industry-specific opportunities to take advantage of that transformation. Keep up with the pace of innovation, unlock the value of your data, and demonstrate to customers that you understand their challenges and how to address them.

Core industries

Industry	Opportunities
Retail	Retailers can enhance or reimagine customer journeys by using Azure services.
Financial services	By modernizing and digitally transforming financial systems to move to cloud platforms like Azure, financial institutions can mitigate common problems and provide more value to their customers.
Healthcare	With Azure and other Microsoft services, healthcare organizations can create highly efficient and resilient healthcare systems that improve patient/provider interactions and provide clinical and data insights.
Government	Azure provides a mission-critical cloud platform, Azure Government, that delivers breakthrough innovation to US government customers and their partners.
Manufacturing	Cloud computing is transforming manufacturing IT infrastructures and processes to the highly available, highly secure, and efficient cloud, in addition to providing cutting edge Internet of Things (IoT), AI / machine learning, and analytics solutions.
Energy and environment	Rising energy needs and sustainability targets are pushing companies to explore innovative solutions and architectures. Innovations like IoT, AI, and machine learning can help address these critical needs.
Telecommunications	Telecommunications organizations use AI, automation, and advanced analytics to realize efficiencies, avoid service disruptions, and reduce costs.
Automotive, mobility, and transportation	Powerful technologies like cloud computing, IoT, AI, and machine learning can help organizations in the automotive, mobility, and transportation industries move people and things safely, quickly, and efficiently.

Industry	Opportunities
Education	Azure provides tools to enable, elevate, and enhance remote learning, connect teachers to students, help organizations create models of smart campus buildings, deploy virtual labs, and more.
Nonprofit	Nonprofit organizations can apply for an Azure credit grant of \$3,500 per year to set up and run an instance of the Microsoft Community Training platform.

Additional verticals

Industry	Opportunities
Game development	Microsoft tools and services help game developers build, scale, and operate games on the global, reliable Azure cloud and incorporate features like multiplayer, leaderboards, translation, and bots.
Media and entertainment	Media and entertainment organizations use cloud computing to reach their customers in more personalized and innovative ways.
Travel and hospitality	Azure has the flexibility and scalability to handle the challenges that are faced in the travel and hospitality industries. These organizations can improve customer service and uncover new business opportunities by using built-in support for analyzing data and key insights.
Facilities and real estate	IoT, AI, machine learning, and other Azure technologies can help building owners, operators, and occupants save money and live better lives.
Aerospace	Intelligent cloud analytics, AI and machine learning, speech capabilities, and improved security are just some of the benefits that aerospace enterprises gain by using Azure.
Agriculture	Agriculture businesses can use Azure FarmBeats, a business-to-business technology, to build AI or machine learning models that are based on aggregated data sets.
Sports	There's been an explosion of data in recent years that sports teams can use to improve the performance of an individual athlete or an entire team.

Related resources

- [Industry-specific Azure IoT reference architectures](#)
- [Cloud adoption for the retail industry](#)

Solutions for the retail industry

Article • 11/01/2023

Retail is one of the fastest growing industries worldwide, generating some of the biggest revenues and accounting to almost a third of American jobs. The core of retail industry is selling products and services to consumers, through channels such as, storefront, catalog, television, and online. Retailers can enhance or reimagine their customer's journey using Microsoft Azure services by:

- keeping their supply chains agile and efficient,
- unlocking new opportunities with data and analytics,
- creating innovative customer experiences using mixed reality, AI, and IoT, and
- building a personalized and secure multi-channel retail experience for customers.

<https://www.youtube-nocookie.com/embed/Vn5x7VM7UwQ> ↗

ⓘ Note

Learn more about a retail company's journey to cloud adoption, in [Cloud adoption for the retail industry](#).

Using Azure services, retailers can easily achieve these goals. For use cases and customer stories, visit [Azure for retail](#) ↗. Microsoft is also revolutionizing the retail industry, by providing a comprehensive retail package, [Microsoft Cloud for Retail](#) ↗.

Architecture guides for retail

The following articles provide more details about retail architectural topics. Although they are mostly conceptual, they can also include implementation details.

Guide	Summary	Technology focus
Data Management in Retail	Primer for how to ingest, prepare, store, analyze, and take action on data, for the retail industry.	Databases
Migrate your e-commerce solution to Azure	Learn how to move an existing e-commerce solution to the cloud. The three stages are to rehost, refactor, and rebuild your solution.	Migration

Guide	Summary	Technology focus
Visual search in retail with Azure Cosmos DB	This document focuses on the AI concept of visual search and offers a few key considerations on its implementation. It provides a workflow example and maps its stages to the relevant Azure technologies.	Databases
SKU optimization for consumer brands	Topics include automating decision making, SKU assortment optimization, descriptive analytics, predictive analytics, parametric models, non-parametric models, implementation details, data output and reporting, and security considerations.	Analytics

Architectures for retail

The following articles provide detailed analysis of architectures developed and recommended for the retail industry.

Architecture	Summary	Technology focus
Batch scoring with R models to forecast sales	Perform batch scoring with R models using Azure Batch. Azure Batch works well with intrinsically parallel workloads and includes job scheduling and compute management.	IoT
Batch scoring with R models to forecast sales	Perform batch scoring with R models using Azure Batch. Azure Batch works well with intrinsically parallel workloads and includes job scheduling and compute management.	AI/ML
Build a content-based recommendation system	This example scenario shows how your business can use machine learning to automate content-based personalization for your customers.	AI/ML
Build a Real-time Recommendation API on Azure	Build a recommendation engine that can be generalized for products, movies, news, and other consumer services, using Azure Databricks, Azure Machine Learning, Azure Cosmos DB, and Azure Kubernetes Service.	AI/ML
Data warehousing and analytics	Build an insightful sales and marketing solution with a data pipeline that integrates large amounts of data from multiple sources into a unified analytics platform in Azure.	Analytics

Architecture	Summary	Technology focus
E-commerce front end	Implement a scalable and cost-effective e-commerce front end using Azure platform as a service (PaaS) tools.	Web
IBM z/OS online transaction processing on Azure	With a dynamically adaptable infrastructure, businesses can realize and launch their products quickly to delight their users. Learn how to migrate a z/OS mainframe OLTP application to a secure, scalable, and highly available system in the cloud.	Mainframe
Intelligent product search engine for e-commerce	Use Azure Cognitive Search, a dedicated search service, to dramatically increase the relevance of search results for your e-commerce customers.	Web
Magento e-commerce platform in Azure Kubernetes Service	Learn how to deploy and host Magento, an open-source e-commerce platform, on Azure.	Web
Retail - Buy online, pickup in store (BOPIS)	Develop an efficient and secure curbside pickup process on Azure.	Web
Scalable order processing	Build a highly scalable and resilient architecture for online order processing, using managed Azure services, such as Azure Cosmos DB and HDInsight.	Web
Stream processing with Azure Databricks	Use Azure Databricks to build an end-to-end stream processing pipeline for a taxi company, to collect, and analyze trip and fare data from multiple devices.	Analytics
Stream processing with Azure Stream Analytics	Use Azure Stream Analytics to build an end-to-end stream processing pipeline for a taxi company, to collect, and analyze trip and fare data from multiple devices.	Analytics

Solution ideas for retail

The following are other ideas that you can use as a starting point for your retail solution.

AI:

- [Customer Feedback and Analytics](#)
- [Optimize Marketing with Machine Learning](#)
- [Personalized marketing solutions](#)
- [Personalized Offers](#)
- [Retail Assistant with Visual Capabilities](#)

Analytics:

- Big data analytics with Azure Data Explorer
- Demand forecasting and price optimization
- Demand forecasting with Azure Machine Learning
- Demand forecasting for shipping and distribution
- Interactive price analytics
- Modern analytics architecture with Azure Databricks

Mixed Reality:

- Facilities management powered by mixed reality and IoT

Networking:

- Video capture and analytics for retail

Web:

- E-commerce website running in secured App Service Environment
- Architect a scalable e-commerce web app
- Scalable Episerver marketing website
- Scalable Sitecore marketing website

What is Microsoft Cloud for Retail?

Article • 01/23/2024

Microsoft Cloud for Retail accelerates business growth by providing trusted retail industry solutions that integrate with retailer's existing systems. Through this complete set of retail specific capabilities across the Microsoft Cloud portfolio, in addition to partner solutions, it becomes possible to seamlessly connect your customers, your people, and your data. Microsoft Cloud for Retail brings together different data sources across the retail value chain and connects experiences throughout the shopper journey using capabilities from Dynamics 365, Microsoft 365, and Azure.

Elevate the shopping experience

Transform the shopping experience through data analytics and store technology to create more engaged shoppers with stronger lifetime value.

Expand table

Capability	Description	Solutions included
Intelligent stores	Maximize sales by optimizing in-store customer and product signals.	Smart Store Analytics
Unified commerce	Increase engagement and drive conversations across channels.	Dynamics 365 Commerce Azure Cognitive Search
Real-time personalization	Enable personalized recommendations and search results to improve customer engagement and product discovery.	Copilot template for personalized shopping Microsoft Intelligent Recommendations Azure Cognitive Search Dynamics 365 Marketing
Digital advertising solutions	Enhance your advertising to drive growth and acquire new customers.	Microsoft Advertising
Seamless customer service	Utilize intelligent and automated customer service tools to improve the customer experience.	Dynamics 365 Commerce Omnichannel for Customer Service Power Virtual Agents

For more information: [Elevate the shopping experience](#).

Build a real-time retail supply chain

Create an agile, resilient retail supply chain by connecting data across your ecosystem to identify issues and optimize performance.

 Expand table

Capability	Description	Solutions included
Demand planning and optimization	Predict demand using AI to optimize inventory.	Dynamics 365 Supply Chain Management Microsoft 365 Teams for Frontline Workers
Supply chain visibility	Use demand and supply signals for future opportunities.	Dynamics 365 Supply Chain Management Dynamics 365 Supply Chain Insights Dynamics 365 Intelligent Order Management Microsoft 365 Teams for Frontline Workers
Flexible fulfillment	Optimize order management, giving customers a choice across delivery channels.	Dynamics 365 Commerce Dynamics 365 Intelligent Order Management Dynamics 365 Supply Chain Management Microsoft 365 Teams for Frontline Workers

For more information, see [Build a real-time, retail supply chain](#).

Empower the store associate

Equip your frontline workforce with solutions that increase customer satisfaction while reducing the burden on your frontline so you can invest in your team's growth.

 Expand table

Capability	Description	Solutions included
Real-time store communication and collaboration	Use modern tools for connecting your team.	Microsoft 365 Teams for Frontline Workers

Capability	Description	Solutions included
Retail workforce management	Automate managerial tasks such as store scheduling.	Store Operations Assist Copilot template for store operations Microsoft 365 Teams for Frontline Workers
Process automation and career development	Expand what your stores and people can do through automation.	Microsoft Viva Learning Microsoft Viva Insights Microsoft Viva Connections

For more information, see [Empower the store associate](#).

Maximize the value of your data

Realize the true value of your data by unifying disparate data and ecosystems across the shopper journey, uncovering insights and optimization throughout.

[Expand table](#)

Capability	Description	Solutions included
Unified customer profile	Gain insights across the complete view of a shopper's journey	Dynamics 365 Customer Insights Retail channel churn model
Shopper and operations analytics	Unlock omnichannel insights with advanced analytics.	Smart Store Analytics Dynamics 365 Customer Insights Azure Synapse Analytics Microsoft Clarity Sitecore OrderCloud Connector (Preview) Frequently bought together (Preview) Retail industry data model (Preview)
Retail media	Unlock ad revenue using your shopper data.	Microsoft PromotelQ

For more information, see [Maximize the value of your data](#).

Next steps

- How to buy Microsoft Cloud for Retail
- Set up and configure Microsoft Cloud for Retail

Elevate the shopping experience

Article • 01/23/2024

Microsoft Cloud for Retail helps you improve the shopping experience for customers both in-store and online. Create personalized e-commerce experiences, optimize your advertising, enhance customer service, increase multichannel engagement, and automate store processes. The following sections highlight key Microsoft technologies and solutions that support the use cases/scenarios of Microsoft Cloud for Retail.

Smart Store Analytics

Get analytics and insights to help retailers grow their smart stores business. Microsoft Smart Store Analytics preview brings together analytics such as store KPIs, data visualizations, AI/ML insights with recommendations on store layout, product catalog, shelf placement and frequently bought together.

For more information, see [Smart Store Analytics](#).

Copilot template for personalized shopping (Preview)

AI shopping assistant conversational commerce starter app enables retailers to provide rich and personalized chat-based commerce experience to their consumers using Generative AI alongside retailer's domain data.

For more information, see [Copilot template for personalized shopping](#).

Dynamics 365 Commerce

Enable customers and partners to buy whenever, wherever and through whatever channel they choose. Microsoft Dynamics 365 Commerce provides the agility, flexibility, and scalability needed to drive successful business outcomes across the buyer's journey.

For more information, see [Dynamics 365 Commerce](#).

Azure Cognitive Search

Improve the quality of search results by prioritizing context and relatedness to prioritize the best matches. Azure Cognitive Search gives developers the infrastructure, APIs, and

tools for building a rich search experience over private, heterogeneous content in web, mobile, and enterprise applications.

For more information, see [Azure Cognitive Search](#).

Microsoft Intelligent Recommendations

Drive better engagement, conversion, revenue, and customer satisfaction with personalized recommendations and insights. Microsoft Intelligent Recommendations help you:

- Significantly improve catalog navigation and item discovery.
- Create upsell and cross-sell opportunities.
- Improve shoppers' experiences and product usability.

You can provide these scenarios with Intelligent Recommendations:

- **Personalized recommendations for end users:** Includes a list of unique content for a specific user based on their consumer habits and interactions. Businesses can recommend products, articles, videos, and more.
- **Real-time and session-based recommendations for both known and unknown users:** Each customer journey that includes new customers, can now have unique recommendations.
- **Similar items:** Recommend products based on various signals (user interactions such as clicks, purchases, and views) or metadata (such as images or text). Intelligent Recommendations can recommend visually similar items in a catalog (for example, floral-patterned shirts) or show similar wine based on the description and taste notes.
- **Basket completion:** Shows complementary items for users based on what is already in their carts.
- **Trending lists:** With user interactions and catalog metadata, you can add algorithmically generated charts such as New, Trending, and Most Popular. These lists can highlight key items on general landing pages or home pages.

For more information, see [Intelligent Recommendations](#).

Dynamics 365 Marketing

Dynamics 365 Marketing elevates customer experiences, allowing you to orchestrate personalized journeys across all touchpoints to strengthen relationships and earn loyalty. The Dynamics 365 Marketing app works seamlessly with Dynamics 365 Sales,

Dynamics 365 Customer Insights, Microsoft Teams, and other products and allows you to make faster and better decisions using the power of data and AI.

For more information, see [Dynamics 365 Marketing](#).

Microsoft Advertising

Connect with your customers through meaningful, one-to-one brand experiences without compromising trust. With Microsoft Advertising, you can create and manage advertising campaigns, measure campaign performance, and track online advertising budgets and spending. And you can evaluate your ad keyword performance and develop insights for optimizing your campaigns.

For more information, see [Microsoft Advertising help](#).

Omnichannel for Customer Service

Omnichannel for Customer Service (Microsoft Dynamics 365 for Customer Service Digital Messaging and Voice add-on) is a robust application that extends the power of Dynamics 365 Customer Service to enable organizations to instantly connect and engage with customers via channels such as Live Chat and SMS.

Omnichannel for Customer Service also provides a modern, customizable, high-productivity app that allows agents to engage with customers across different channels. The application offers contextual customer identification, real-time notification, integrated communication, and agent productivity tools like KB integration, search, and case creation to ensure agents are effective.

For more information, see [Omnichannel for Customer Service](#).

Power Virtual Agents

Power Virtual Agents lets you create powerful chatbots that can answer questions from visitors to your website or service. These virtual agents can also contextually and seamlessly transfer the conversation to live agents.

For more information, see [Power Virtual Agents](#).

Maximize the value of your data

Article • 01/23/2024

Realize the true value of your data by unifying disparate data and ecosystems across the shopper journey, uncovering insights and optimization throughout. Microsoft Cloud for Retail includes a wide variety of tools to help you make the most of your data. The following sections highlight key Microsoft technologies and solutions that support the use cases/scenarios of Microsoft Cloud for Retail.

Smart Store Analytics

Improve in-store shopping experience and drive business performance of your fleet of smart stores using Microsoft's advanced analytics and data science.

For more information, see [Smart Store Analytics](#).

Sitecore OrderCloud Connector (Preview)

Bring your commerce data from Sitecore OrderCloud into a standardized format and draw actionable insights from it.

For more information, see [Sitecore OrderCloud Connector](#).

Frequently bought together (Preview)

The AI/ML Frequently Bought Together model empowers the store or the merchandising manager to make data-driven decisions on products' placement and promotions, based on insights on closely related products.

For more information, see [Frequently bought together](#).

Retail industry data model (Preview)

Plan, architect, and design data solutions for data governance, reporting, business intelligence, and advanced analytics

For more information, see [Retail industry data model](#).

Dynamics 365 Customer Insights

Combine real-time transactional, behavioral, and demographic data to create a complete view of your customers. Use prebuilt or custom AI models to predict customer needs. Combine transactional, behavioral, and demographic data in real time to create a complete view of your customers. Use prebuilt or custom AI models to predict customer needs.

For more information, see [Dynamics 365 Customer Insights](#).

Retail channel churn model

AI-based retail channel churn predictive model assesses the chance customers churn, or stop actively buying from you. After you configure and run it, the retail channel churn model assigns a score to each customer that reflects the chance they'll churn, and identifies the most influential factors on those scores. You can see which customers are likely to churn and which aspects of their customer experience weigh the heaviest on that risk, so you can plan ways to delight your customers and retain their business.

For more information, see [Unified customer profile](#).

Azure Synapse Analytics

Discover powerful insights across your data with Azure Synapse that brings together data integration, enterprise data warehousing, and Big Data analytics. It gives you the freedom to query data on your terms, using either serverless or dedicated resources—at scale.

For more information, see [Azure Synapse Analytics](#).

Microsoft Clarity

See what your users want. Replay user sessions and explore heat maps to make your website work better for your customers and your business.

For more information, see [Microsoft Clarity](#).

Microsoft PromoteIQ

Generate ad revenue by enabling trusted brand partners to promote products on your digital channels. Microsoft PromoteIQ helps retailers work directly with brand partners to promote products on-site.

For more information, see [PromoteIQ in Microsoft Advertising](#) .

Security in Microsoft Cloud for Retail

Article • 01/10/2023

Microsoft's approach to securing data relies on the understanding of the shared responsibility in the cloud model.

Microsoft

Microsoft cloud services are built on a foundation of trust and security. Microsoft enables the best-in breed security controls, monitoring, and protections to ensure that it's trustworthy when you come to the cloud.

The security of your Microsoft cloud service is an operational partnership between Microsoft and you.

You

You own your data and all user identities. You're responsible for protecting them, the security of your on-premises resources, and the security of cloud components you control (varies by service).

Responsibility	On-prem	IaaS	PaaS	SaaS
Customer data	■	■	■	■
Configurations and settings	■	■	■	■
Identities and users	■	■	■	■
Client devices	■	■	■	□
Applications	■	■	□	□
Network controls	■	■	□	□
Operating system	■	■	□	□
Physical hosts	■	□	□	□
Physical network	■	□	□	□
Physical data center	■	□	□	□

■ Customer □ Shared □ Microsoft

ⓘ Note

Microsoft Cloud for Retail uses PaaS and SaaS services only.

Microsoft commitment to secure solutions

Microsoft cloud services are built on a foundation of trust and security. Microsoft enables the best in breed security controls, monitoring, and protections to ensure that when you come to the cloud, it's trustworthy. Microsoft uses the best development and operation practices outlined in [Microsoft Security Development Lifecycle \(SDL\)](#) and [Microsoft Operational Security Assurance \(OSA\)](#). Microsoft developers must validate that source code, documentation, configurations, and dependencies don't cause unintended side effects. For more information, go to [Security development and operations overview](#).

The data security section in [Microsoft Products and Services Data Protection Addendum \(DPA\)](#) describes the security practices and policies adopted by Microsoft online services.

Shared responsibility and customer responsibilities

To address your privacy controls ensure your data is secure, we recommend that you follow a set of best practices when deploying into Azure:

- [Azure security best practices and patterns](#)
- [Microsoft Services in Cybersecurity](#)

Protecting your data also requires that all aspects of your security and compliance program include your cloud infrastructure and data. The following guidance can help you to secure your deployment.

Microsoft Purview for data governance and inventory discovery

Ensuring that your data stored in the cloud, hybrid, and on-premise is classified and cataloged reflects one of the most essential elements of a security model. [Microsoft Purview](#) can help you assess and inventory your network.

Microsoft Purview can connect to and classify the following services used in Microsoft Cloud for Retail:

- Microsoft Dataverse
- Microsoft Power BI

Microsoft Defender for Cloud to protect your deployment

You can use [Defender for Cloud](#) to protect Microsoft Cloud for Retail. Defender for Cloud provides Cloud Security Posture Management (CSPM) and Cloud Workload Protection Platform (CWPP) for all of your Azure, on-premises, and multicloud (Amazon AWS and Google GCP) resources. Defender for Cloud fills three vital needs as you manage the security of your resources and workloads in the cloud and on-premises:

- [Defender for Cloud secure score](#): Continually assesses your security posture so you can track new security opportunities and precisely report on the progress of your security efforts.
- [Defender for Cloud recommendations](#): Secures your workloads with step-by-step actions that protect your workloads from known security risks.
- [Defender for Cloud alerts](#): Defends your workloads in real-time so you can react immediately and prevent security events from developing.

Defender for Cloud can protect the following elements of Microsoft Cloud for Retail:

- [Teams and Office 365](#)
- [Microsoft Power BI](#)
- [Dynamics 365](#)
- [Identity and Microsoft Entra ID integration](#)
- [Microsoft Sentinel](#)

Microsoft Sentinel cloud-based security operations

[Microsoft Sentinel](#) brings together signals that include Microsoft Purview, Defender for Cloud, and data logs across your environment.

You can integrate the following services used in Microsoft Cloud for Retail into Microsoft Sentinel for a full view of your security information and event management (SIEM) and security orchestration, automation, and response (SOAR) solution.

- [Microsoft Purview](#)
- [Power Apps logging](#)
- [Dynamics 365 continuous threat monitoring](#) ↗

- Microsoft Entra ID
- Other data sources

For guidance on deploying, managing, and using Microsoft Sentinel, go to [Best practices for Microsoft Sentinel](#).

Configuring your auditable logs in Office 365 will give you a richer view of your data. Microsoft provides an extensive set of logging and audit capabilities that are included in [Office 365 Security and Compliance Center](#) and [Microsoft Defender for Cloud](#) . You can enable logging and monitoring for each service capability:

- [Power Apps activity logging](#)
- [Power Automate activity logging](#)
- [Data loss prevention activity logging](#)
- [Dynamics 365 auditing](#)
- [Microsoft Dataverse and model-driven apps activity logging](#)
- [Microsoft Teams logging](#)

See also

- [Security center](#)
- [Microsoft Power Platform security documentation](#)
- [Trust Center](#) 
- [Resources for Software Assurance](#) 
- [Azure Government security](#)

Compliance in Microsoft Cloud for Retail

Article • 01/11/2024

Microsoft Azure, Microsoft Dynamics 365, Microsoft 365, and Microsoft Power Platform services and its underlying infrastructure employ a security framework that encompasses industry best practices and spans multiple standards. These include the ISO 27000 family of standards and others. As part of our comprehensive [compliance offering](#), Microsoft regularly undergoes independent audits performed by qualified third-party accredited assessors.

To use Microsoft Cloud for Retail, you need to agree to the [Online Service Terms](#) and the [Microsoft Privacy Statement](#). These are the qualifying license terms for Microsoft 365/Office 365, Dynamics 365, Microsoft Power Platform, and Azure.

The following table lists the products available with Microsoft Cloud for Retail and their compliance offerings:

[\[+\] Expand table](#)

Product Family	Product	ISO 27001	ISO 27017	ISO 27018	ISO 22301	SOC2 Type 2	PCI DSS Level	GDPR
Dynamics 365	Marketing	✓	✓	✓	✓	✓	✓	✓
Dynamics 365	Customer Service	✓	✓	✓	✓	✓	✓	✓
Dynamics 365	Customer Insights	✓	✓	✓	✓	✓	✓	✓
Dynamics 365	Commerce	✓	✓	✓	✓	✓	✓	✓
Dynamics 365	Connected Store	✓	✓	✓	✓	✓	-	✓
Dynamics 365	Fraud Protection	✓	✓	-	-	-	✓	-
Dynamics 365	Intelligent Order Management	✓	✓	✓	✓	✓	-	✓

Product Family	Product	ISO 27001	ISO 27017	ISO 27018	ISO 22301	SOC2 Type 2	PCI DSS Level	GDPR Level 1
Dynamics 365	Intelligent Recommendations	-	-	-	-	✓	-	-
Dynamics 365	Supply Chain Insights	-	-	-	-	-	-	-
Dynamics 365	Supply Chain Management	✓	✓	✓	✓	✓	✓	✓
Dynamics 365	Chat for Dynamics	✓	✓	✓	✓	✓	✓	✓
Microsoft 365	Microsoft Teams	✓	-	-	-	-	-	-
Microsoft 365	Viva Connections	-	-	-	-	-	-	-
Microsoft 365	Viva Insights	-	-	-	-	-	-	-
Microsoft 365	Viva Learning	-	-	-	-	-	-	-
Microsoft Azure	Azure Search	✓	✓	✓	✓	✓	✓	✓
Microsoft Azure	Azure Synapse Analytics	✓	✓	✓	✓	✓	✓	✓
	Clarity	-	-	-	-	✓	-	✓
	Power Virtual Agents	✓	✓	✓	✓	✓	✓	✓
	PromotelQ	-	-	-	-	-	-	✓

Legend:  = available

You can find more [details](#) about these offerings on our compliance page.

Elevated access

Microsoft internal policy allows Microsoft employees who have the appropriate security group membership to request temporary **just-in-time** elevated access so that they can

perform servicing and support activities on production systems. The internal ticketing system tracks and reviews every **just-in-time** access request.

Disclaimer

It is important to understand that PCI DSS compliance status for Microsoft Cloud for Retail solutions doesn't automatically translate to PCI DSS certification for the services that customers build or host on these platforms. Additionally, Microsoft Cloud for Retail doesn't offer payment card processing as a service and thus doesn't use an acquirer. Customers are responsible for ensuring that they achieve compliance with PCI DSS requirements.

You can find the regulatory compliance standards that apply to certain features offered through the Microsoft Retail Add-On on the [compliance dashboard](#). You can also visit our [Trust Center](#) to learn more about Microsoft's commitments to data protection and privacy.

Resources

- [Trust Center](#)
- Microsoft 365 [data residency](#) and [Privacy](#)
- Azure [data residency](#) and [Privacy](#)
- Dynamics 365 and Power Platform [data residency](#) and [Privacy](#)
- [Security in Microsoft Cloud for Retail](#)

Microsoft Dynamics 365 Commerce documentation

Discover how to make the most of Dynamics 365 Commerce with training, documentation, and videos covering product capabilities. Learn how to use Commerce to deliver unified, personalized, and seamless buying experiences for customers and partners.



GET STARTED
[Get started](#)



WHAT'S NEW
[What's new](#)



TRAINING
[Training](#)



CONCEPT
[Troubleshoot](#)

Implement

Guidance

[Dynamics 365 implementation guidance](#)
[Implementation lifecycle management](#)
[Adoption frameworks](#)
[Security](#)

Try/Buy/Deploy

[Deploy a demo environment](#)
[Before you buy](#)
[Deployment options](#)

Set up

[Environment planning](#)
[Architecture](#)
[Onboard an implementation project](#)
[Prepare for go-live](#)
[Configure and install Commerce Scale Unit \(self-hosted\)](#)

Administer and develop

Platform development

Commerce development

Manage environments

Develop and customize your environment
Deploy and access development environments
Application stack and server architecture

Migrate to the Commerce SDK
Introduction to Commerce Scale Unit (CSU) core
Point of sale (POS) extension
E-commerce online extensibility development
Retail software development kit (SDK) (legacy)

Dynamics 365 Lifecycle Services resources
Continuous delivery
Upgrades, updates, and hotfixes
Set up new environments, Azure DevOps, and branches for projects
Update code and environments for Commerce projects

Manage data storage
Data management
Data import and export
Database movement operations
Data entities

System administration
System administration
Process automation
Batch processing

Organization administration
Organization administration
Organizations and organizational hierarchies
Workflow system
Number sequences

Manage user access
Role-based security
Create users
Assign users to security roles
Manage e-commerce users and roles

Set up channels
Set up a retail channel
Set up an online channel
Online and offline point of sale (POS) operations
Store Commerce app
Dynamics 365 Payment Connector for Adyen

Integration development
Integration between finance and operations apps and third party apps
Data management REST API
Service endpoint overview
Design principles and best practices for data entities
Consume Retail Server APIs in external applications

Microsoft Office integration
Office integration
Configure and send email
Customize the Open in Microsoft Office menu

Microsoft Power Platform integration
Power Platform integration
Dual-write
Virtual entities
Dual-write mapping reference

Azure Data Lake integration
Export to Azure Data Lake
Export to Azure Data Lake in finance and operations apps

Commerce and Microsoft Teams integration

Developer reference

- API references
- X++ language reference
- Service endpoint overview
- Open Data Protocol (OData)
- Commerce Scale Unit customer and consumer APIs

Reporting and analytics with Power BI

Developer tools

- Set up a development environment
- Development tools in Visual Studio
- Create deployable packages

Enable Azure Data Lake Storage in a Commerce

Extend and customize

- Extensibility overview
- Write extensible code
- Manage your e-commerce site
- POS extension basics
- POS user interface visual configurations

Use

Merchandise products and services

- Product information
- Manage attributes and attribute groups
- Retail sales price management
- Retail discounts
- Inventory availability for online and retail channels

Manage orders

- Customer orders in point of sale (POS)
- Create returns in point of sale (POS)
- Distributed order management (DOM)

Manage customers

- Loyalty
- Clienteling overview

Manage financials

- Statements
- Edit and audit online order and asynchronous customer order transactions

Manage worker tasks

- Task management overview
- Configure task management
- Create task lists and add tasks
- Assign task lists to stores or employees

Manage your e-commerce site

- E-commerce site overview
- Create an e-commerce site
- Ways to add content
- Digital asset management
- Make your site compliant

Related products

Dynamics 365 Fraud Protection

Dynamics 365 Supply Chain Management

Dynamics 365 Intelligent Order Management

Help protect your revenue and customers with adaptive AI that continuously learns to protect you against payment...

Dynamics 365 Sales

Dynamics 365 Sales enables salespeople to build strong relationships with their customers, take actions base...

Supply Chain Management allows you to provide the right services and products to your customers...

Dynamics 365 Finance

Reimagine your financial models to prepare, respond, and thrive in the face of disruption.

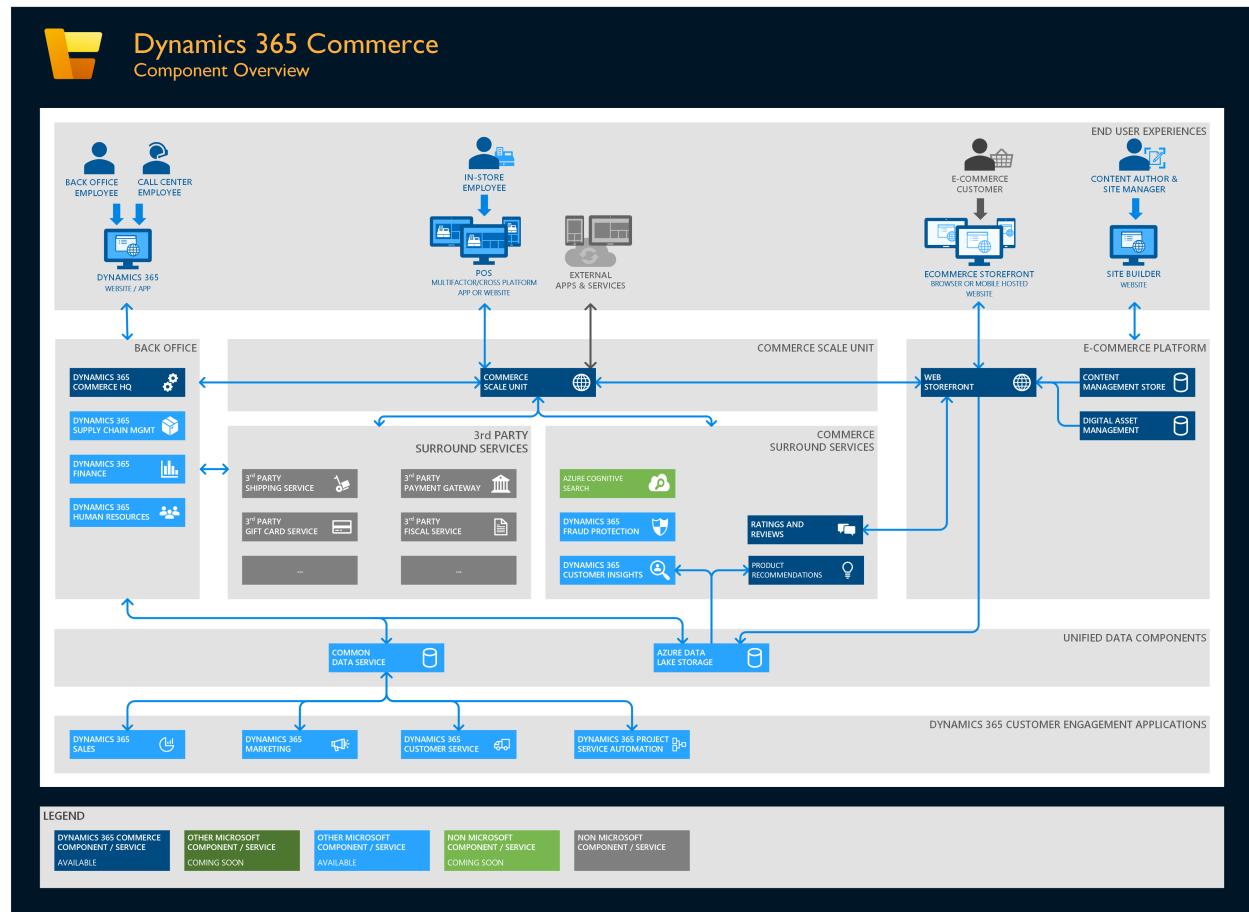
Centrally manage orders from capture to fulfillment using real-time inventory data tools. Meet growing digital...

Dynamics 365 Commerce architecture overview

Article • 09/29/2023

This article provides an overview of all components in the Microsoft Dynamics 365 Commerce ecosystem, including integration points to the suite of Dynamics 365 products.

The following illustration shows an overview of Dynamics 365 Commerce components.



Architecture benefits

Omni-enabled headless commerce engine

The Commerce Scale Unit hosts the headless commerce engine. It serves as the central integration point for all commerce business logic, and powers a complete omni-channel solution across physical and digital stores. The Commerce Scale Unit is built by using a portable architecture, and allows for flexible hosting options across cloud, edge, and hybrid topologies.

The headless commerce engine powers all native Dynamics 365 Commerce channels, including in-store and e-commerce channels. It also serves as the single integration point for third-party channel solutions. Therefore, those solutions can take advantage of the power of Dynamics 365 Commerce business logic and integration with other commerce-related services that are provided by Microsoft and independent software vendors (ISVs).

Interconnected business processes

The platform that is shared among the various Dynamics 365 business applications, such as Dynamics 365 Commerce, Dynamics 365 Supply Chain Management, and Dynamics 365 Finance, provides a set of interconnected business processes that users can immediately benefit from. All back-office capabilities across these applications are built on the same web experience and data stores. Therefore, there is a seamless flow of business processes across various functions in the organization, but custom integrations across applications and services aren't required. The out-of-the-box integration between the headless commerce engine and the back office further expands the coverage of these interconnected business processes across the back office and commerce channels.

Unified data

Dynamics 365 Commerce provides a unified data solution through out-of-the-box integrations with [Microsoft Dataverse](#) and [Azure Data Lake Storage](#). Integrations and data sharing across Dynamics 365 apps such as Dynamics 365 Sales and Dynamics 365 Marketing are supported through Microsoft Dataverse. Transactional data in Azure Data Lake Storage is used to power analytics and insight scenarios using various integrations.

Powered by AI and analytics

Because of the accessible, persistent, up-to-date, and unified organizational data that is available in Data Lake Storage, the whole organization has a "single source of truth" that analytics, artificial intelligence (AI), and machine learning (ML) can be applied on top of. In this way, the organization can derive insights and get key performance indicators (KPIs) that can be used to optimize and automate business processes across all channels.

Business and transactional analytical reports in POS and Commerce headquarters

The prepackaged set of business and transactional analytical reports in Commerce provides retailers with intelligent insights across all points of the Commerce ecosystem

by embedding high-charts and SSRS-based reports in Store commerce point of sale (POS) systems and Commerce headquarters. The commerce analytics solution provides a comprehensive set of out-of-the-box business and transactional reports, enabling retailers to take advantage of insights across all channels.

The highcharts-based reports on POS are real-time, enabling store associates to analyze their channel's transactional and other POS activities at the conclusion of a transaction. This contrasts with the out-of-the-box reports in Commerce headquarters, which enable retailers to view analytical reports for transactional activity across all channels, once the data is synced. Commerce headquarters reports are specifically designed for channel managers who focus on sales performance to predict trends and uncover insights, allowing them to drill down in reports about organization-wide sales performance across global geography by employee, category, product, terminal, channel, and more.

Component overview

User experiences

Dynamics 365

Dynamics 365 is a collection of applications that together provide comprehensive and flexible enterprise resource planning (ERP) solutions for medium to large businesses. It provides an extensible framework and ecosystem that can be tailored to customer-specific requirements via an extensive set of partners. Dynamics 365 applications provide capabilities for their target business segments. They also take advantage of each other, and other Microsoft services and offerings, to provide solutions that help run customers' complex businesses.

POS

POS simplifies the experience for the retailers by providing Omnichannel solutions in emerging channels. These channels perform a variety of commerce business operations like cash and carry transactions, cash/shift management, customer engagement, assisted selling, clienteling, endless aisle, order processing/fulfillment, inventory management, and reporting.

Store Commerce app

The Store Commerce app is a cross-platform (Windows, iOS, and Android), multi-form factor (desktop, tablet, and phone) solution for all in-store first-line workers, such as

cashiers, sales associates, stock clerks, and store managers. It can be deployed as an app that has offline capabilities. Alternatively, it can be deployed in the cloud and accessed through a web browser. The application is role-based and fully configurable from headquarters. It's also highly customizable, and can be extended or integrated into third-party services by using the Retail software development kit (SDK).

In addition to standard "cash and carry" transaction processing, Store Commerce includes features for assisted selling, clienteling, endless aisle, order processing/fulfillment, inventory management, cash/shift management, and reporting. For more information, see [Modern POS \(MPOS\) architecture](#) and [Choose between Store Commerce app and Store Commerce for web](#).

E-commerce storefront

The e-commerce storefront is the customer-facing website rendering system. It's built on the React.js framework, and uses a combination of server-side and client-side rendering to deliver responsive web experiences for one or more online channels. Although the storefront has a rich set of out-of-box capabilities, it's also highly customizable, and delivers an efficient and scalable solution for online business. For more information, see [Online store overview](#).

Site builder

Site builder is the web-based authoring interface for the content management and storefront website rendering systems. Visual page builder in site builder is a what-you-see-is-what-you-get (WYSIWYG) editor for site managers and content authors who perform the day-to-day workflow tasks of managing and producing the marketing content for the e-commerce experience. In site builder, a marketer can provide more marketing detail for specific products to enhance the shopping experience for consumers. In addition, site builder includes integrated accessibility reporting, URL management, site map generation, and image focal point management, among other features. For more information, see [Online store overview](#).

External services and apps

The headless commerce engine that is exposed via the Commerce Scale Unit lets partners and customers take advantage of all the same channel-side capabilities and business logic that are used by the out-of-box e-commerce and point of sale (POS) components. Therefore, by tapping into the same data and business process capabilities, it allows for seamless omni-channel capabilities across out-of-box channel components and partner-provided/customer-developed services and applications. It

also provides access to all out-of-box and ISV-developed surround services that are available through the Commerce Scale Unit.

Back office

Dynamics 365 Commerce headquarters

The Dynamics 365 Commerce application, which is often referred to as the Commerce headquarters component, provides back-office capabilities that enable the configuration of products, employees, business processes, and other functionality that is required for the business. It's also the application that call center workers use to provide assisted commerce-related workflows.

Dynamics 365 Supply Chain Management

[Dynamics 365 Supply Chain Management](#) provides functionality to help you manage your products throughout the supply chain lifecycle, from production, to inventory and warehouse, to transportation and distribution. For more information, see [Supply Chain Management documentation](#).

Dynamics 365 Finance

[Dynamics 365 Finance](#) provides functionality to automatically manage your global finances. For customers of the Dynamics 365 Commerce application, Dynamics 365 Finance offers an integrated experience for managing stores and e-commerce financial statements alongside the rest of their operations. For more information, see [Dynamics 365 Finance help resources](#).

Dynamics 365 Human Resources

[Dynamics 365 Human Resources](#) lets businesses get a comprehensive view of their employee resources and manage them in a unified way. It provides integrated experiences from the hiring process through workforce planning and employee time management. For more information, see [Dynamics 365 Human Resources help resources](#).

Commerce Scale Unit

Retailers are distributed organizations, where the business topography can be represented as a hub and spoke model. Dynamics 365 Commerce supports this model

by having head-office capabilities (the hub), and also many distributed channel components (the spokes) that can be deployed and self-managed in-store or in nearby Microsoft-managed Azure datacenters. The spokes are referred to as Scale Units, because they represent physical isolation (a function of scale) and an atomic unit of update.

To facilitate cloud and edge computing scenarios, a Commerce Scale Unit is available both as a software as a service (SaaS) component that is managed by Microsoft (Commerce Scale Unit in the Cloud) and as a self-managed component that can be deployed locally (self-hosted). A single environment can have a mixture of Commerce Scale Units (cloud and self-hosted). Therefore, organizations can tune their investments in operational overhead on a store-by-store basis by implementing network redundancy for poor connectivity. For more information, see [Select an in-store topology](#).

Commerce Scale Units (cloud)

Multiple Commerce Scale Units can be associated with each environment. Each Commerce Scale Unit can be independently serviced and updated, and each can serve one or more channels across one or more legal entities. Each Commerce Scale Unit can be deployed to any of the supported Azure regions, and multiple Commerce Scale Units can be deployed to the same region. The independent nature of each Commerce Scale Unit allows for phased rollout of updates across a collection of channels.

Commerce Scale Units (self-hosted)

The ability to bring the Commerce Scale Units to edge computing helps accommodate scenarios where internet connectivity is poor or unreliable. For retailers, this approach typically means having a physical footprint in their stores. By using a Commerce Scale Unit (self-hosted), retailers can bring the same business logic and capabilities that run in the Azure cloud into their stores. In these cases, although in-store connectivity is presumably more reliable, self-management of these components will involve additional overhead in terms of monitoring and updates. For more information, see [Select an in-store topology](#).

E-commerce platform

Content management system

A fully featured content management system (CMS) is integrated directly into the e-commerce platform. In addition to rich indexing features, the CMS provides lifecycle

management for marketing materials that supplement the product information that is managed by the headless commerce engine. It includes features for localization and multi-item publishing through releases. The system is built on top of a scalable, resilient Azure infrastructure that includes Azure Active Directory (Azure AD) and Azure Cosmos DB.

Digital asset management

Commerce digital asset management extends the content management store, and keeps track of images, videos, and file downloads that are served by the web storefront site. Its image resizer service optimizes downloaded images for different devices and contexts. In this way, it helps enhance performance while it also manages image quality. Digital asset management is also integrated with Azure Media Services for efficient playback of video streams.

Web storefront

The CMS stores its pages as a series of modules. The storefront web server assembles those modules into a rendered HTML page. The web storefront is composed of the rendering platform, the commerce data proxy, and the extensibility layer. Those components form a base that is supplemented by a set of modules that power a web-based commerce experience, the Dynamics 365 Commerce module library. The initial module library can be modified to meet each business's unique requirements.

Alternatively, it can be supplemented by extensions and modules that are developed by a partner.

Commerce surround services

Dynamics 365 Fraud Protection

[Dynamics 365 Fraud Protection](#) is integrated into the e-commerce checkout flows that are managed and processed through the Commerce Scale Unit. The connection to the service is automatically provisioned with the Commerce Scale Unit, and customers who sign up for Dynamics 365 Fraud Protection can enable and configure the integration in Commerce headquarters. The service can run either in "evaluate" mode, so that you can assess the effectiveness of the service, or in "protect" mode, so that you can catch fraudulent transactions by using configured business rules. For more information, see [Dynamics 365 Fraud Protection integration with Dynamics 365 Commerce](#).

Dynamics 365 Customer Insights

Dynamics 365 Customer Insights helps you gain a deeper understanding of your customers by connecting data from various transactional, behavioral, and observational sources to create a 360-degree customer view and generate insights. Dynamics 365 Commerce makes it easy for retailers to enable the integration with Dynamics 365 Customer Insights and show the generated insights at the POS. These insights include churn probability and next best action, and they are valuable because they help sales associates have effective conversations with customers and deliver personalized shopping experiences to them. For more information, see [Dynamics 365 Customer Insights integration with Dynamics 365 Commerce](#).

Azure cognitive search

[Azure Cognitive Search](#) is integrated into Commerce to provide consistent product discovery and search experiences across all of the Commerce channels that use the Commerce Scale Unit (CSU). With integrated Azure Cognitive Search, customers can quickly find products by browsing categories, searching, and filtering. Improved product discoverability capabilities help retailers increase customer retention and conversion rates. Azure Cognitive Search also provides the scalability and performance required for e-commerce traffic. For more information, see the [Cloud powered search overview](#) article.

Product recommendations

Dynamics 365 Commerce can be used to show product recommendations on the e-commerce website and POS devices. These product recommendations are items that a customer might be interested in, and they are based on the purchase trends of other customers in online and brick-and-mortar stores.

Product recommendations let customers easily and quickly find products that they might want to purchase, and cross-selling and upselling can be used to help customers find additional products that they didn't originally intend to buy. When recommendations are used to assist with product discovery, they can help create more conversion opportunities, increase sales revenue, and enhance customer satisfaction and retention. For more information, see [Product recommendations overview](#).

Ratings and reviews

The Commerce ratings and reviews solution lets online retail customers enter product reviews and ratings through the e-commerce storefront. Retailers can then show

averaged ratings and review information across their e-commerce websites. Azure Cognitive Services offers automatic moderation of profane words in 40 languages, and because human approval isn't required, moderation costs are reduced. The system also offers moderator tools that can be used to respond to customer concerns, feedback, and take-down requests, and to address data requests from users. For more information, see [Rating and reviews overview](#).

Unified data components

Azure Data Lake Storage

Customers who bring their own Azure Data Lake Storage accounts can take advantage of structured business data from back-office operations and clickstream data from the e-commerce storefront. This data flows back into intelligence services such as product recommendations, customer insights, and commerce analytics to power customer-centric business processes and user experiences. Those business processes and user experiences can then be embedded back into Dynamics 365 Commerce headquarters, the POS, and e-commerce storefronts. For more information, see [Make Entity store available as Data Lake](#).

Dataverse

Dataverse is the unified data store that integrates the data from all your business applications. Dynamics 365 applications such as Dynamics 365 Sales, Dynamics 365 Customer Service, and Dynamics 365 Commerce use Dataverse to store business data. Therefore, Dataverse enables cross-business application scenarios, and can power new scenarios through Power Apps and Power Automate. For more information, see [What is Microsoft Dataverse?](#).

Additional resources

[Dynamics 365 Commerce authentication flows](#)

[Azure Data Lake Storage](#)

[Microsoft Dataverse](#)

[Modern POS \(MPOS\) architecture](#)

[Dynamics 365 Supply Chain Management](#)

[Dynamics 365 Human Resources](#)

[Dynamics 365 Finance](#) ↗

[Dynamics 365 Fraud Protection](#)

[Dynamics 365 Fraud Protection integration with Dynamics 365 Commerce](#)

[Dynamics 365 Customer Insights](#)

[Azure Cognitive Search](#)

Dynamics 365 Commerce authentication flows

Article • 09/29/2023

This article provides an overview of the various authentication flows in Microsoft Dynamics 365 Commerce. Although the Dynamics 365 Commerce solution currently supports several authentication scenarios and flows, the core authentication infrastructure of the Commerce Scale Unit (also known as the headless commerce engine) is fully based on [OpenID Connect](#).

Authentication methods

Access to each of the application programming interfaces (APIs) on the Commerce Scale Unit is natively restricted by one or more of the following roles:

- **Employee** – Access to APIs associated with this role requires point of sale (POS) device activation (a device token) and an authenticated employee.
- **Customer** – Access to APIs associated with this role requires an authenticated customer. E-Commerce sites generally use these APIs for operations such as retrieving order history and changing customer details.
- **Application** – Access to APIs associated with this role requires application-level authentication, such as Azure Active Directory (Azure AD) service-to-service authentication.
- **Anonymous** – APIs associated with this role are primarily used by e-Commerce sites without user authentication.
- **Customized APIs** – Access to APIs associated with this role can be restricted using any of the methods described above such as POS device activation, customer authentication, and anonymous authentication.

For the full list of Commerce Scale Unit APIs and their access restrictions, see [Commerce Scale Unit customer and consumer APIs](#).

Supported authentication methods

The following table describes the set of supported authentication methods for APIs that require either POS device activation that generates a device token or user authentication that generates a user token.

API category	Scenario	Supported authentication method	Required setup	Additional details
Employee	Dynamics 365 POS authentication flows*	Simple cashier user name and password	In Dynamics 365 Commerce headquarters, configure a user name and password for a worker.	Create a worker
Employee	Dynamics 365 POS authentication flows*	Azure AD credentials	In Commerce headquarters, configure a worker that is mapped to Azure AD credentials.	Enable Azure Active Directory authentication for POS sign-in
Employee	Dynamics 365 POS authentication flows*	Extended sign-in credentials (for example, by using a bar code or a magnetic stripe reader [MSR])	In Commerce headquarters, configure a worker for extended sign-in.	Set up extended sign-in functionality for Store Commerce app and Store Commerce for web
Customer	Dynamics 365 Commerce authentication flows	Site user authentication by using Azure AD B2C with implicit scope flow	<ol style="list-style-type: none"> <li data-bbox="969 1215 1187 1462">Create an Azure AD business-to-consumer (B2C) application. <li data-bbox="969 1462 1187 1799">In Commerce headquarters, add the Azure AD B2C application to the accepted list of identity providers. <li data-bbox="969 1799 1187 2023">In Commerce site builder, configure the Azure AD B2C application. 	Set up a B2C tenant in Commerce Set up custom pages for user sign-ins

API category	Scenario	Supported authentication method	Required setup	Additional details
Customer	Dynamics 365 Commerce authentication flows	Site user authentication by using an external identity provider that supports OpenID Connect with implicit scope flow	<ol style="list-style-type: none"> 1. Create an Azure AD B2C application, and configure it to support external identity providers. 2. In Commerce headquarters, add the Azure AD B2C application to the accepted list of identity providers. 3. In Commerce site builder, configure the Azure AD B2C application. 	Set up a B2C tenant in Commerce Set up custom pages for user sign-ins
Customer	Third-party e-Commerce authentication flows	Site user authentication by using an external identity provider that supports OpenID Connect with implicit scope flow	In Commerce headquarters, add the external identity provider to the accepted list of identity providers.	Configure authentication providers
Application	Third-party app or service authentication flows	Azure AD service-to-service authentication/application authentication	In Commerce headquarters, add the external identity provider to the accepted list of identity providers.	

* Sign-in to POS requires device activation for each terminal. For more information, see [Point of Sale \(POS\) device activation](#).

Unsupported authentication flows

Scenario	Unsupported authentication method	Details
Dynamics 365 POS authentication flows	Authentication without device activation (that is, without a device token)	All POS-related Commerce Scale Unit APIs require a device activation token for authentication.
Dynamics 365 Commerce authentication flows	Site user authentication by using Azure AD business-to-consumer (B2C) with authorization code or On-Behalf-Of flows	Authorization code and On-Behalf-Of flows are not currently supported with e-commerce site user authentication.
Third-party e-commerce authentication flows	Site user authentication by using an external identity provider that supports OpenID Connect with authorization code or On-behalf-of flows	Authorization code and On-behalf-of flows are not currently supported with e-commerce site user authentication.

Dynamics 365 POS employee authentication flows

The following illustration shows POS employee authentication flows in Commerce.



Dynamics 365 Commerce

Dynamics 365 POS Employee Authentication Flows

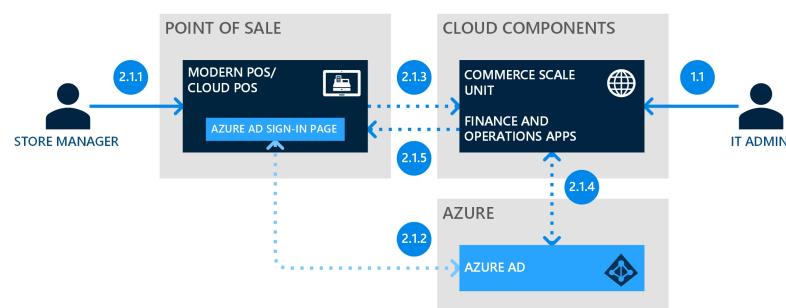
POS DEVICE ACTIVATION

1. GRANT DEVICE ACTIVATION PERMISSION

- 1.1. IT admin maps Azure AD user to worker in Finance and Operations apps to grant device activation permissions.

2. ACTIVATE DEVICE

- 2.1.1. Cashier opens the POS application and signs in with Azure AD credentials.
2.1.2. Azure AD user token is generated via Azure AD.
2.1.3. Azure AD user token is passed to Commerce Scale Unit to activate the device.
2.1.4. Commerce Scale Unit validates Azure AD user token with Azure AD.
2.1.5. Commerce Scale Unit issues device token to the POS application for further authentication.



POS CASHIER AUTHENTICATION USING SIMPLE USER NAME AND PASSWORD

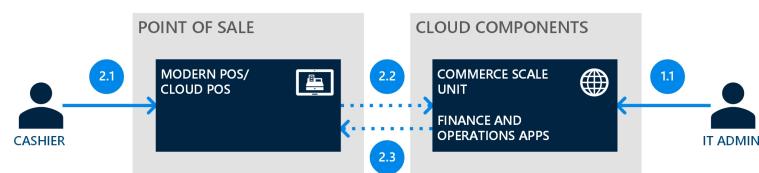
1. GRANT POS WORKER PERMISSION

- 1.1. IT admin grants worker permissions for POS sign-in, including worker ID and password.

2. SIGN IN TO POS

- 2.1. Cashier opens the POS application and signs in with worker credentials maintained in Finance and Operations apps.
2.2. Commerce Scale Unit validates user input against user credentials entered in Finance and Operations apps.
2.3. Commerce Scale Unit generates and issues user token to the POS application for further authentication.

Note: All subsequent requests are authenticated using the device token and user token.



POS CASHIER AUTHENTICATION USING EXTENDED SIGN-IN

1. GRANT POS WORKER PERMISSION

- 1.1. IT admin grants worker permissions for POS sign-in, setting up extended logon capabilities.

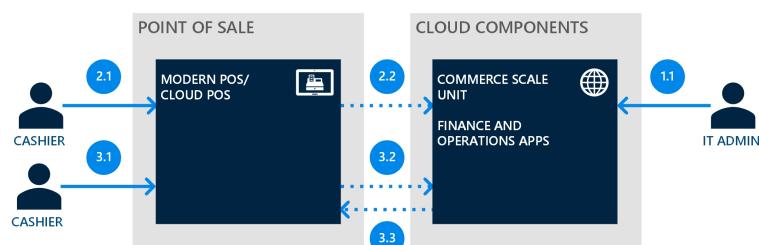
2. ENROLL IN EXTENDED SIGN-IN

- 2.1. Store manager signs in to the POS and enrolls cashier with extended sign-in credentials (for example, bar code or MSR).
2.2. Commerce Scale Unit stores hashed and encrypted authentication credentials.

3. SIGN IN TO POS WITH EXTENDED CREDENTIALS

- 3.1. Cashier signs in to the POS using extended credentials (for example, bar code or MSR).
3.2. Commerce Scale Unit validates user input against user credentials entered in Finance and Operations apps.
3.3. Commerce Scale Unit generates and issues user token to the POS application for further authentication.

Note: All subsequent requests are authenticated using the device token and user token.



POS CASHIER AUTHENTICATION USING AZURE AD

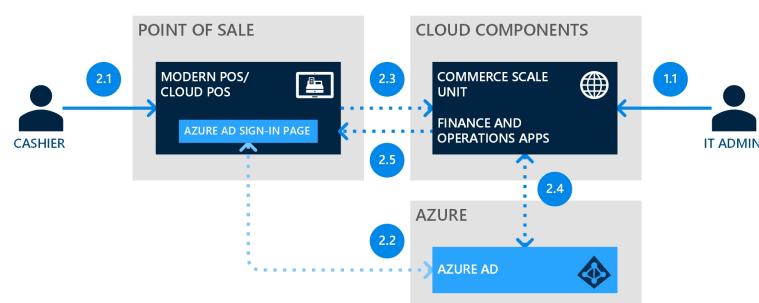
1. GRANT DEVICE ACTIVATION PERMISSION

- 1.1. IT admin maps Azure AD user to worker in Finance and Operations apps.

2. SIGN IN TO POS

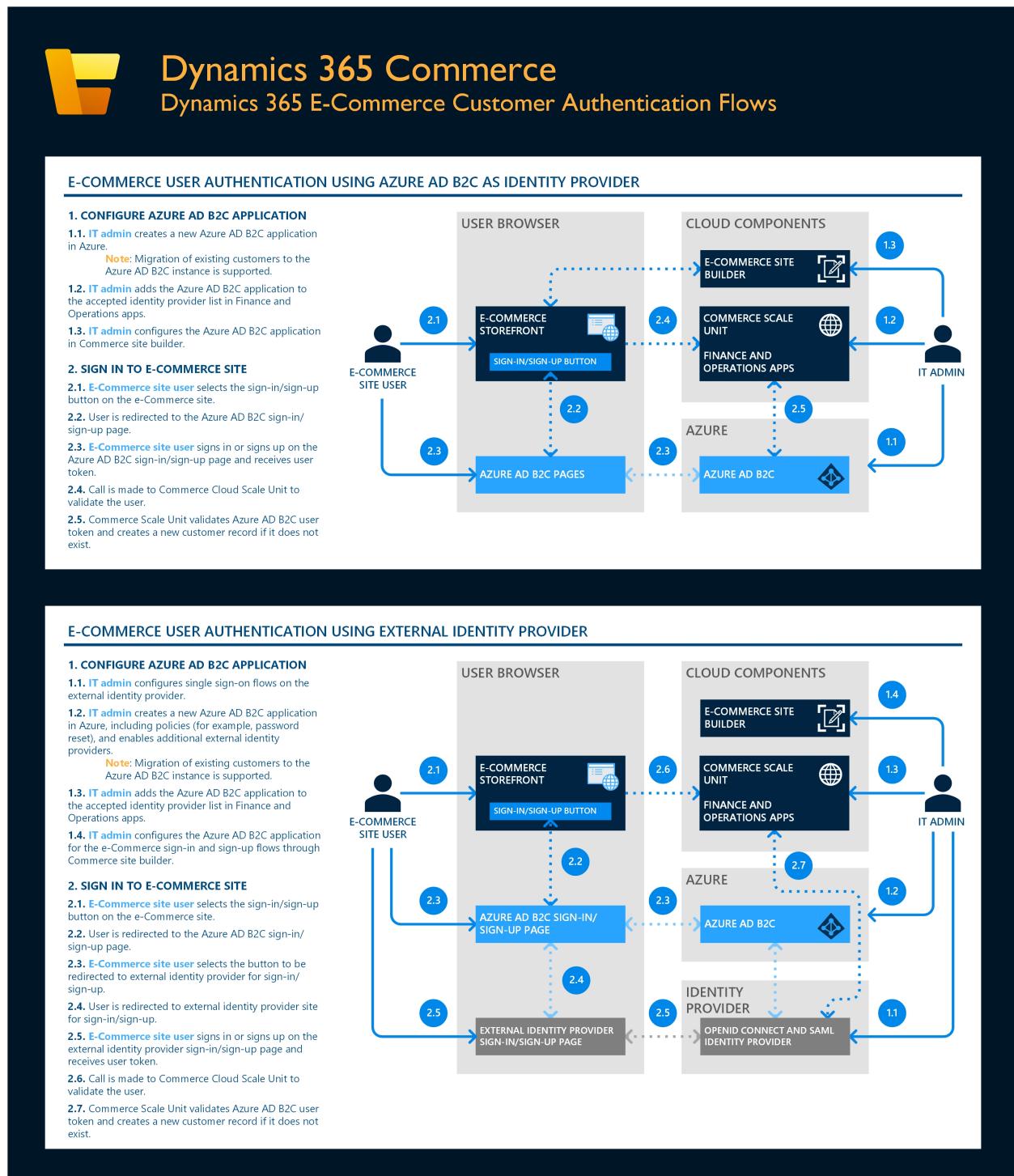
- 2.1. Cashier opens the POS application and signs in with Azure AD credentials.
2.2. Azure AD user token is generated via Azure AD.
2.3. Call is made to Commerce Scale Unit to authenticate the user.
2.4. Commerce Scale Unit validates Azure AD user token.
2.5. Commerce Scale Unit generates and issues user token to the POS application for further authentication.

Note: All subsequent requests are authenticated using the device token and user token.



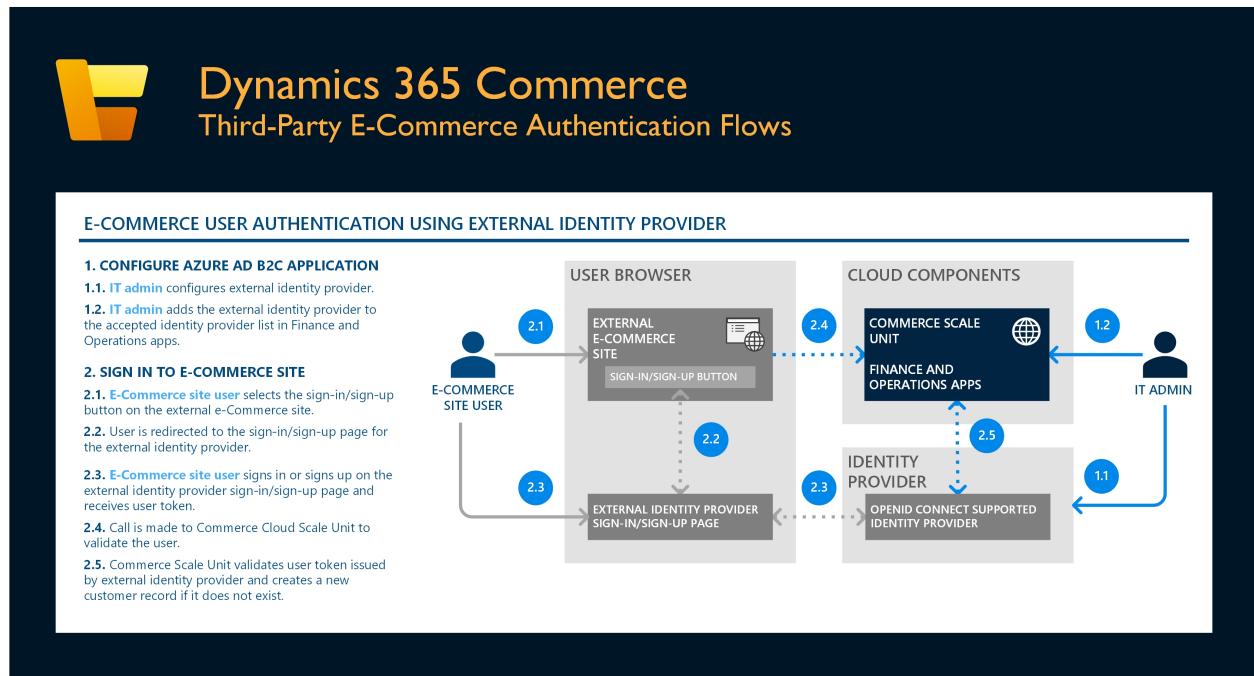
Dynamics 365 e-Commerce customer authentication flows

The following illustration shows e-Commerce customer authentication flows in Commerce.



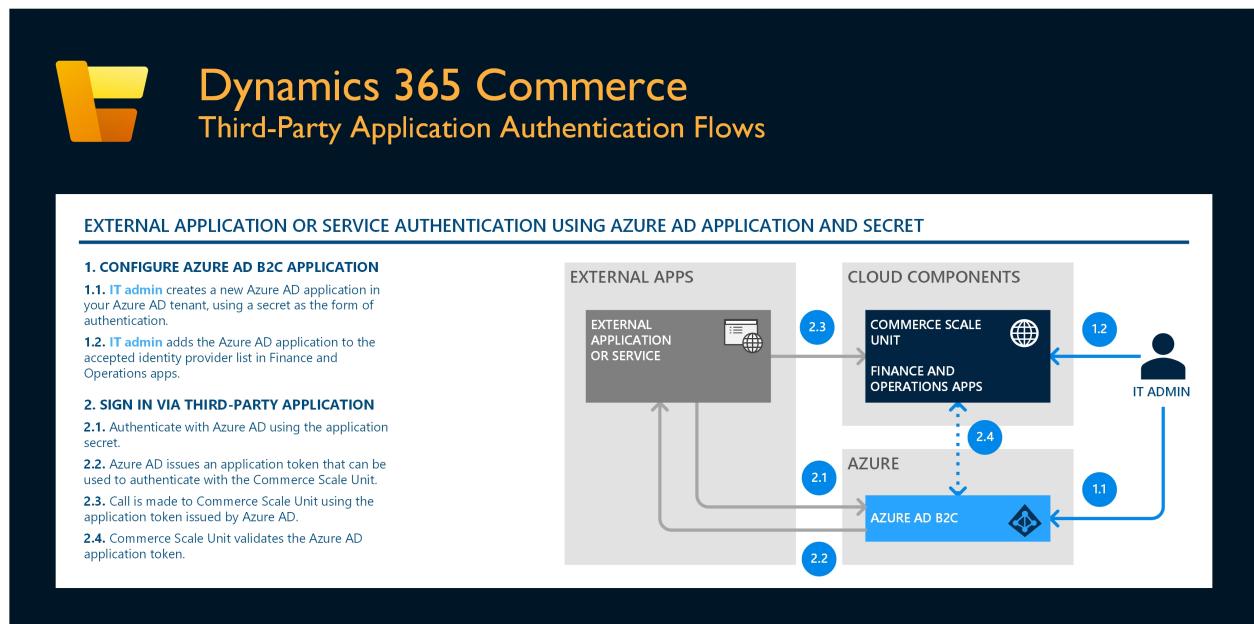
Third-party e-Commerce customer authentication flows

The following illustration shows third-party e-Commerce customer authentication flows in Commerce.



Third-party application authentication flows

The following illustration shows third-party application authentication flows in Commerce.



Additional resources

[Dynamics 365 Commerce architecture overview](#)

[Commerce Scale Unit customer and consumer APIs](#)

[POS worker logon](#)

[Enable Azure Active Directory authentication for POS sign-in](#)

[Set up extended sign-in functionality for Store Commerce app and Store Commerce for web](#)

[Set up a B2C tenant in Commerce](#)

[Set up custom pages for user sign-ins](#)

[Configure authentication providers](#)

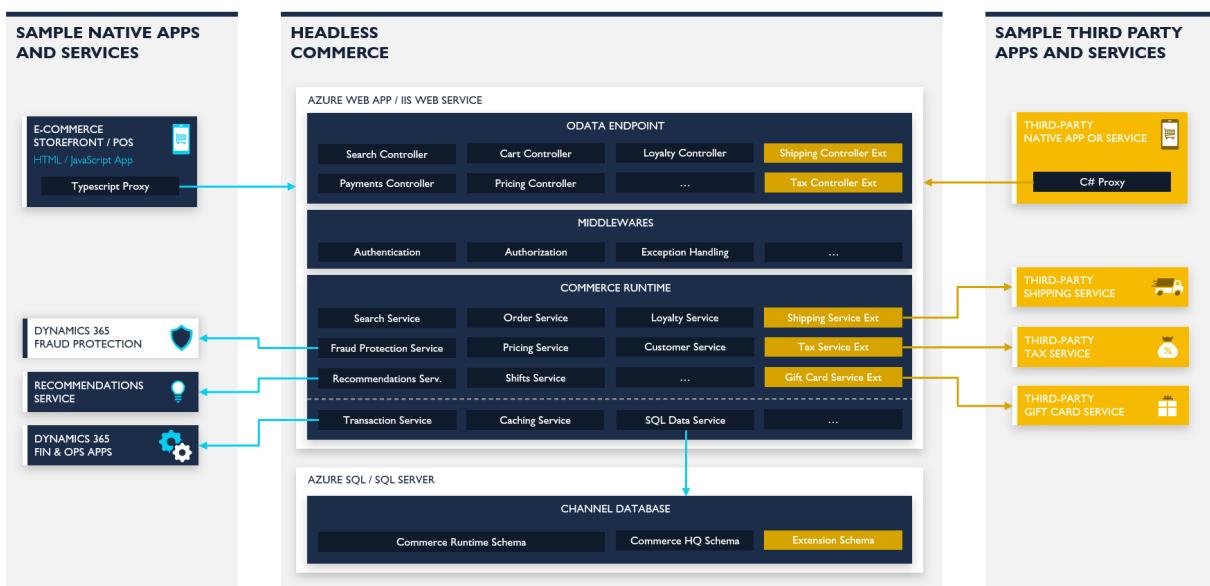
[Point of Sale \(POS\) device activation](#)

Headless commerce architecture

Article • 08/12/2022

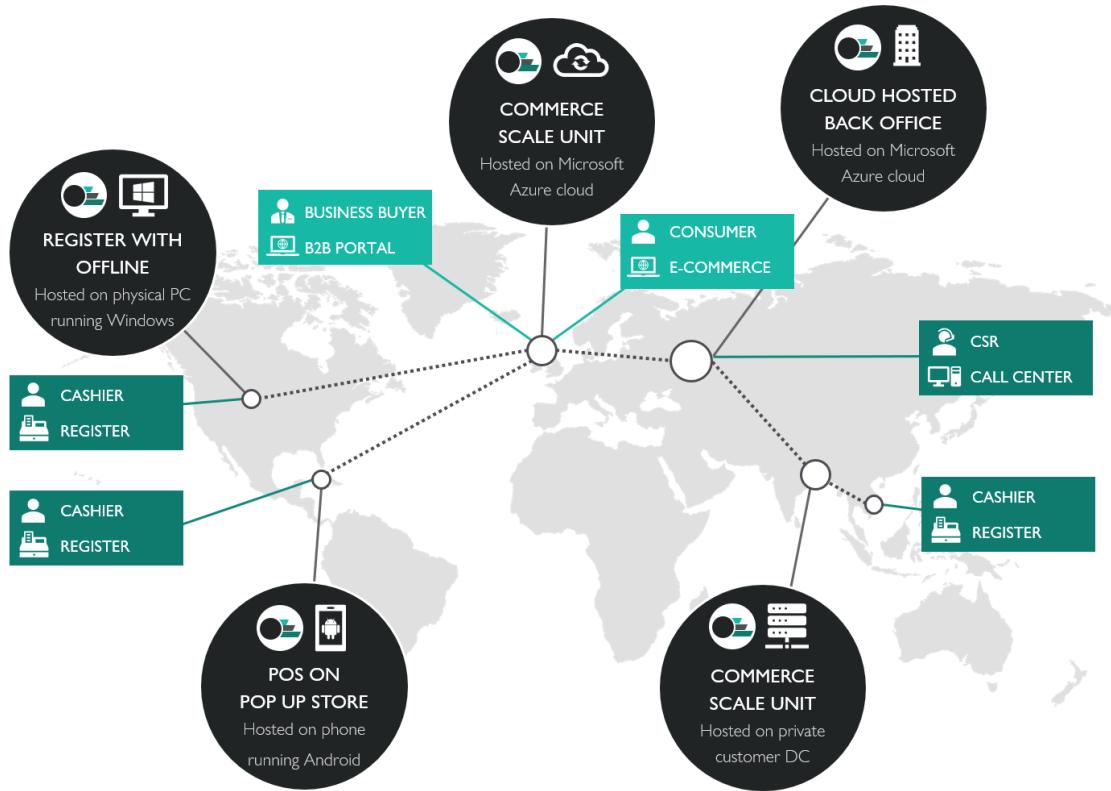
This article describes the architecture of the headless commerce (also known as Commerce Scale Unit). The headless commerce is an API-driven framework that enables extensible, personalized, friction-free commerce experiences, and integrated, optimized back-office operations.

HEADLESS COMMERCE



Omnichannel solution provided by the headless commerce

The commerce APIs of the headless commerce are consumed by Microsoft Dynamics 365 Commerce (back-office, in-store, call center, and e-commerce) and provide a complete omnichannel solution. The APIs can be consumed by third-party applications and Microsoft Power Platform connectors.



Components

The headless commerce contains these components:

- Consumer APIs
- Commerce runtime (CRT)
- Channel database

Consumer APIs

The headless commerce exposes Open Data Protocol (OData) APIs for Dynamics 365 Commerce and third-party applications to consume. The API layer is built by using ASP.NET Core. It provides different authentication options so that the clients can consume the APIs. The APIs are a wrapper that exposes the business logic. For more information, see the following articles:

- [Commerce Scale Unit customer and consumer APIs](#)
- [Consume APIs](#)
- [Custom APIs](#)

Commerce runtime

CRT is a collection of portable .NET libraries that contain the core commerce business logic. The consumer APIs expose the business logic for clients to consume. To add or modify business logic, customize CRT. For more information, see the following articles:

- [Commerce runtime \(CRT\) services](#)
- [CRT Extensions](#)

Channel database

The channel database holds transactional data and master data from one or more commerce channels, such as an online store or a brick-and-mortar store. Master data is pushed down from Commerce headquarters to the channel database by using Commerce Data Exchange (CDX). Transactional data that is stored in the channel database is pulled back to Commerce headquarters by using CDX. For more information, see [Channel database extensions](#).

Commerce Data Exchange and commerce channel communications

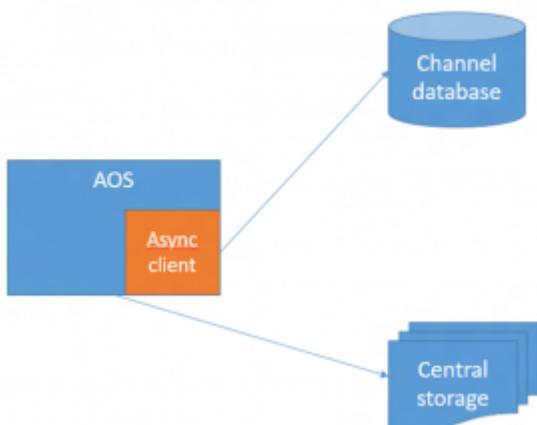
Article • 08/12/2022

This article provides an overview of Commerce Data Exchange and its components. It explains the part that each component plays in the transfer of data between Microsoft Dynamics 365 Commerce Headquarters and channels.

Commerce Data Exchange is a system that transfers data between Headquarters and channels, such as online stores or brick-and-mortar stores. The database that stores data for a channel is separate from the Commerce database. The channel database holds only the data that is required for transactions. Master data is configured in Headquarters and distributed to channels. Transactional data is created in the point of sale (POS) system or the online store, and then uploaded to Headquarters. Data distribution is asynchronous. In other words, the process of gathering and packaging data at the source occurs separately from the process of receiving and applying data at the destination. For some scenarios, such as price and inventory lookups, data must be retrieved in real time. To support these scenarios, Commerce Data Exchange also includes a service that enables real-time communication between Headquarters and a channel.

Async Service

Microsoft SQL Server change tracking on the Commerce database is used to determine the data changes that must be sent to channels. Based on a distribution schedule, Headquarters packages that data and saves it to central storage (Azure blob storage). A separate batch process uses the Commerce Data Exchange: Async Client library to insert this data package into the channel database.



Commerce scheduler

Scheduler jobs are the mechanism for distributing data to and from locations. Jobs are made up of subjobs, which specify the tables and table fields that contain the data to distribute. Headquarters includes predefined scheduler jobs and subjobs that meet the replication requirements of most organizations. The following types of predefined jobs are created:

- **Download jobs** – Download jobs send data that has changed from Headquarters to channel databases. Modifications to records are tracked through SQL Server change tracking.
- **Upload jobs (P jobs)** – Upload jobs pull sales transactions from a channel into the Commerce database. P jobs upload data incrementally. When a P job runs, the Async Client library checks the replication counter for records that have already been received from a location. A record is uploaded only if its replication counter is more than the largest value that is found. P jobs don't update data that was previously uploaded.

The distribution schedule is used to run the data transfer, either manually or by scheduling a batch job in Headquarters. A distribution schedule can contain one or more channel data groups, and one or more scheduler jobs. To ensure that the scheduler jobs are running smoothly, do not rename the "Default" database configured for the instance, and do not create a second database. All non-Commerce Scale Unit databases are managed by Microsoft, and only one default database is expected.

Realtime Service

Commerce Data Exchange: Real-time Service is an integrated service that provides real-time communication between Headquarters and channels. Real-time Service enables individual POS computers and online stores to retrieve specific data from Headquarters in real time. Although most key operations can be performed in the local channel database, the following scenarios require direct access to the data that is stored in Headquarters:

- Issuing and redeeming gift cards
- Redeeming loyalty points
- Issuing and redeeming credit memos
- Creating and updating customer records
- Creating, updating, and completing sales orders
- Receiving inventory against a purchase order or transfer order
- Performing inventory counts

- Retrieving sales transactions across stores and completing return transactions



A predefined Real-time Service profile is created.

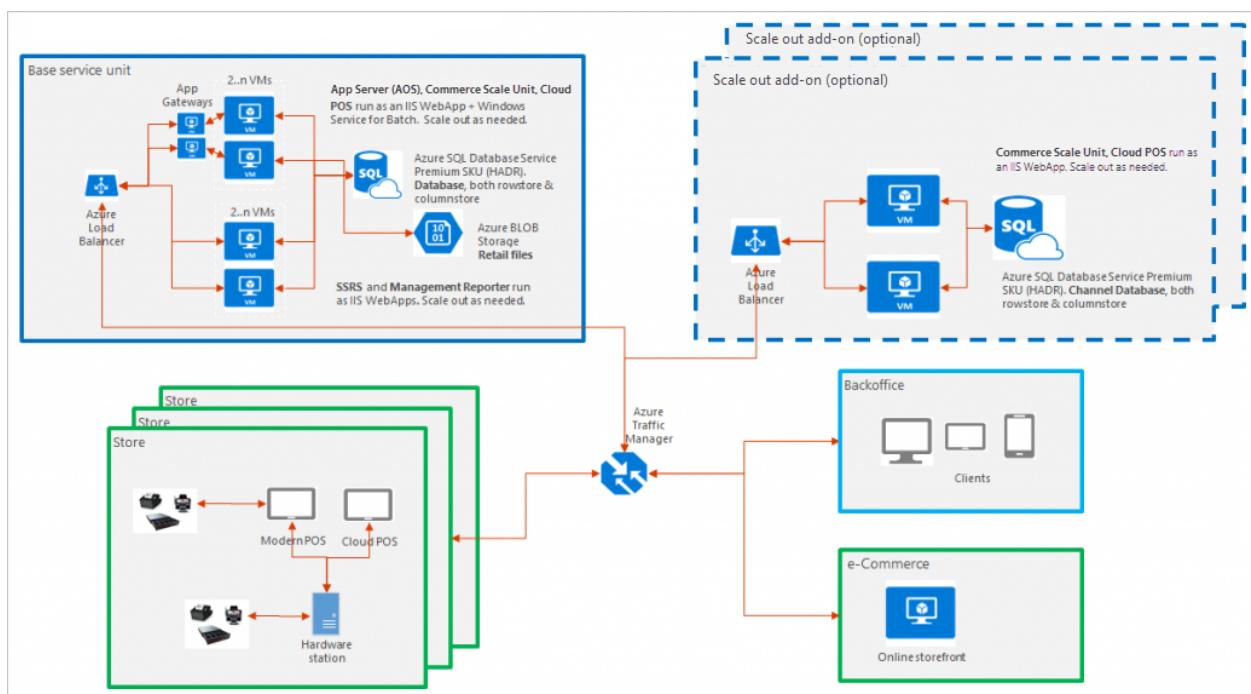
Modern POS (MPOS) architecture

Article • 08/12/2022

This article describes the POS topology.

Modern POS topology

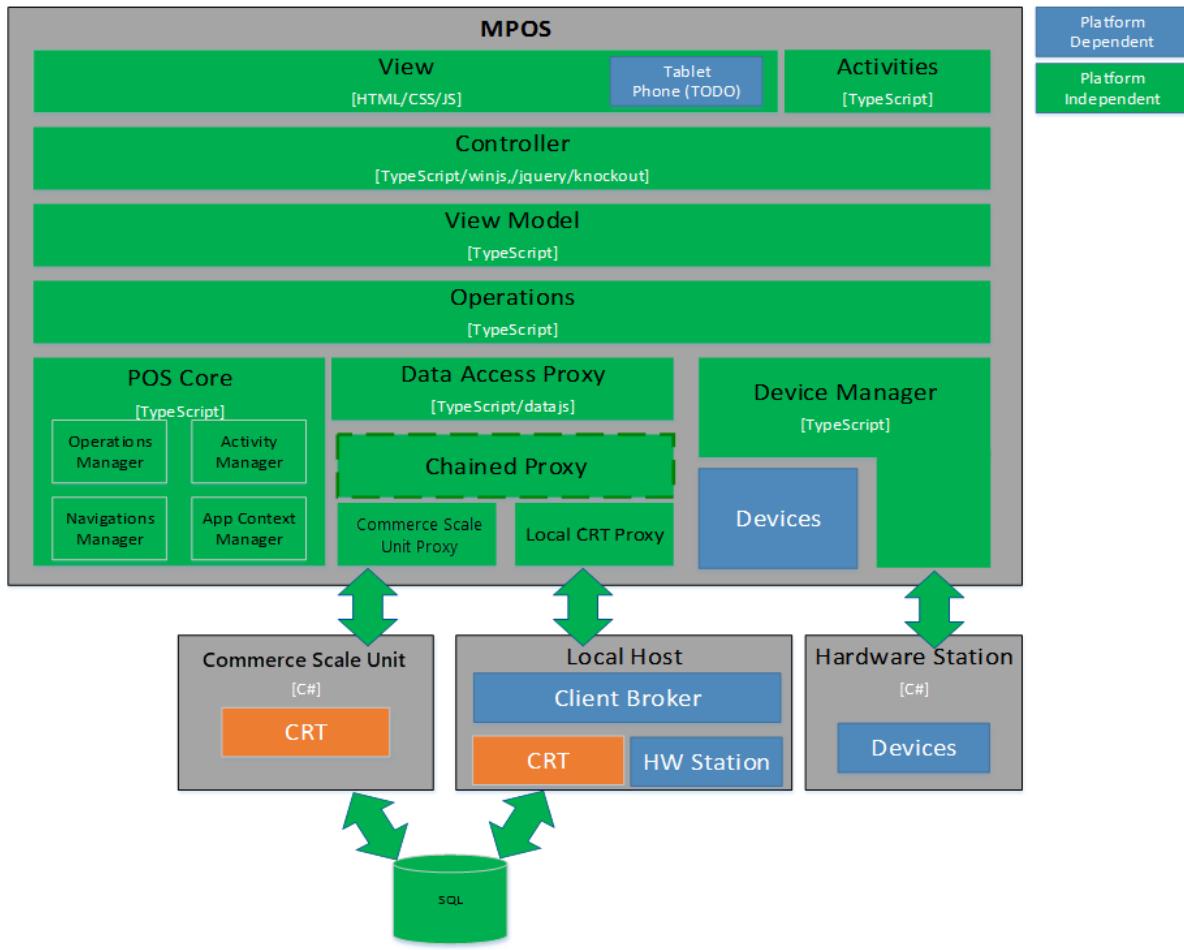
Users of Modern Point of Sale (POS) can perform various tasks on supported laptops, tablets, and phones. These tasks include processing sales transactions, viewing customer orders, managing daily operations and inventory, and viewing role-based reports. Both MPOS and Cloud POS are available in Microsoft Dynamics 365 Commerce. The Cloud POS is a hosted version of the POS app. Both the POS clients don't perform business functions or data processing. All business functions are provided by Commerce Scale Unit. Modern POS and Cloud POS clients can communicate with Commerce Scale Units. Modern POS client can also communicate with peripheral devices, such as cash drawers, credit card readers, and printers, by using Hardware Station. Hardware Station must be deployed in your store, and all Modern POS clients can connect to the same Hardware Station. The following diagram shows the high-level topology.



Modern POS architecture

The view, view-controller, and devices layers depend on the operating system (for example, Windows RT) that you plan to deploy Modern POS on. The other layers are independent of the operating system. These layers use TypeScript classes and modules

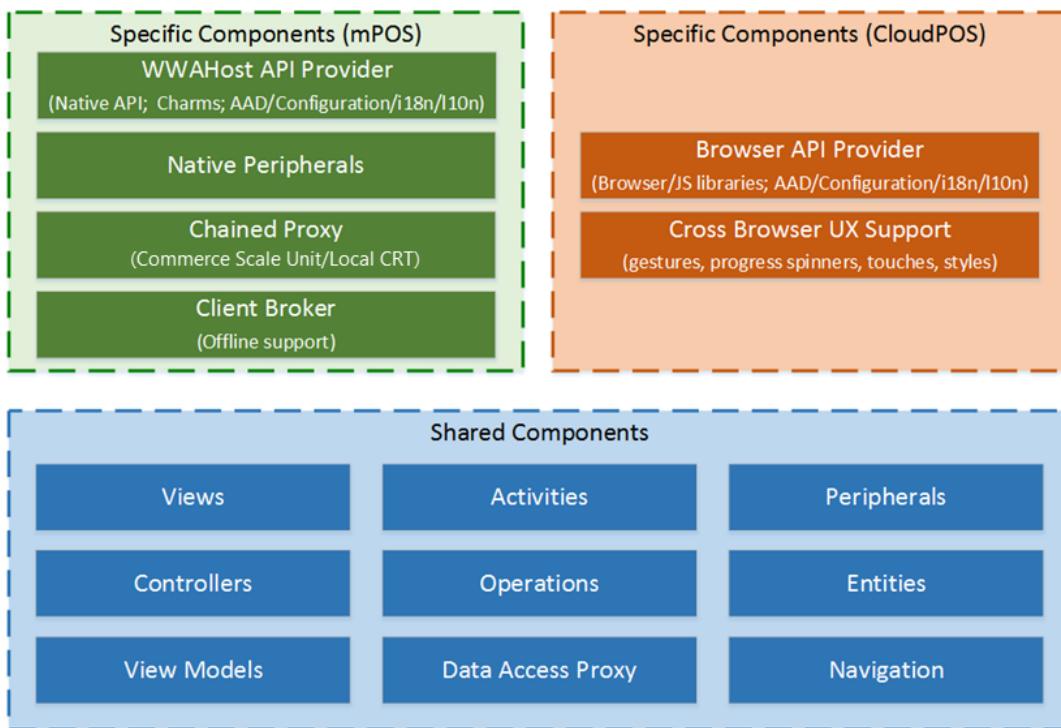
to implement Modern POS functionality such as workflows and entities. The following diagram shows the Modern POS technical architecture.



Cloud POS and Modern POS architecture

Cloud POS is a hosted version of Modern POS, and varies only in the way that it is rendered on specific devices or in specific browsers. Additionally, Modern POS supports offline mode and therefore a local CRT. Other native peripheral support is also specific to Modern POS.

CloudPOS v. mPOS. Shared and Specific Components



Set up new environments, Azure DevOps, and branches for projects

Article • 05/03/2023

Important

Support for the Retail SDK will end in October 2023. Please use or migrate to the [Commerce SDK](#), which provides several benefits including a simplified development and update experience, and improved performance.

Most environments for Microsoft Dynamics 365 Commerce projects are hosted in the cloud. They are either Microsoft-hosted on a Microsoft subscription or cloud-hosted on a customer subscription. By default, environments are Microsoft-hosted. You can use cloud-hosted environments to provide more control over a development or build environment. For more information, see the [Lifecycle Services \(LCS\) user guide](#).

Development Tier 1 environments

Development environments are called Tier 1 environments. There are three options for hosting a development environment:

- The Commerce app comes with one Sandbox Tier 1 environment. (For more information, see the [Microsoft Dynamics 365, Enterprise edition, Licensing Guide](#).)
- A cloud-hosted environment that you run on your own Microsoft Azure subscription. This type of environment is known as "cloud-hosted" in Microsoft Dynamics Lifecycle Services (LCS).
- A downloaded virtual machine (VM) that you host in a location of your choice.

If your implementation of Commerce includes code extensions, we recommend that you use a development environment where you have administrator privileges. If you don't have administrator privileges in your development environment, you won't be able to install programming tools or configure the operating system.

The hosting model that you choose has a financial impact. You can reduce some of the hosting cost by using a Tier 1 environment as a simple test environment or golden configuration environment. One Tier 1 environment is free with your Dynamics 365 subscription. Although this approach isn't ideal, it should work for most projects.

If you want to extend channel components, see [Prepare the development environment](#) to learn how to configure a development environment so that it's ready for development.

Note

You can shut down cloud-hosted environments at any time. This capability helps reduce the hosting cost.

A hosting alternative is to download a virtual hard disk (VHD) from LCS and host it locally on a server. From a development perspective, VHD images have the same capabilities as a hosted VM. The only difference is that LCS deployments aren't supported on VHDs. However, command-line deployments are supported.

The following table shows the advantages and disadvantages of each hosting model. Use this information to evaluate the model that will work best for your project.

Hosting model	Advantages	Disadvantages
Microsoft-hosted environment (in an LCS project, default or based on an add-on)	Your subscription includes one Tier 1 environment. We recommend that you use this environment as a build environment. Telemetry data is collected and is available on the LCS diagnostics page.	Users can't perform administrative actions. Users can't install any tools or certificates.
Cloud-hosted environment (in an LCS project, private subscription)	You have full administrative rights. You can install tools and certificates.	There is additional cost. You can mitigate this cost by shutting down the environment.
Self-hosted downloaded VM	The experience depends on the host. The experience can be much faster if the VM runs on a solid-state drive (SSD).	You can't deploy packages from LCS.

Tier 2 and higher machines are multi-box environments for multiple test and verification purposes. Production environments are hands-off, and the size of the environment is determined by the sizing process in LCS.

Branches, build definitions, and environments

Branching is an important practice in software development. The [Branching and Merging Primer](#) article describes the advantages of branching:

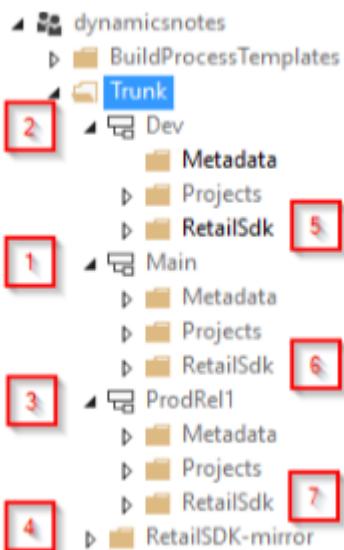
① Note

A branching and merging strategy involves a tradeoff between risk and productivity. You trade the safety of working in isolation for the increased productivity of working with other people. The productivity increases come with a cost—the additional effort required for merging software assets sometime in the future.

Using branches provides better isolation and control of individual software assets and increases productivity, because teams or individuals can work in parallel. However, using branches also requires an increase in merge activities and therefore risk, because you must later reassemble branches into a whole.

There is no single best strategy for creating branches. The strategy depends on the project and the size of the implementation.

The following illustration shows three code branches: Dev, Main, and ProdRel1. The numbers indicate the order of setup.



Here is an explanation of the setup. The numbers in brackets refer to the numbers in the preceding illustration:

- The **Dev** branch [2] is used for daily work that isn't ready for testing or might not be stable, but that must be shared with other developers. For larger teams, you might want to have multiple Dev branches for different features or purposes.

- The **Main** branch [1] is for changes that meet a certain quality bar and are ready for testing by other people. This testing might include user acceptance tests, performance tests, integration tests, and sanity tests after hotfixes. Deployable packages for this branch must be created by a build environment. As a best practice, you should not generate X++ packages in a Tier 1 environment and then deploy those packages into an official test or production environment. Otherwise, uncommitted source changes could be included. The correct approach is always to deploy packages that were built on official build environments.
- The **ProdRel1** branch [3] holds all source code exactly as it's deployed in a production environment at any given point. A build environment can be used but isn't required. If packages from the Main branch are deployed to a production environment, the code should be merged (from Main to ProdRel1) after a production deployment. By having a branch for production, you can generate official builds later if you require them.
- All three branches hold both X++ code (extensions and hotfixes in Metadata folders) and a copy of the Retail software development kit (SDK) in **RetailSdk** folders [5, 6, 7]. The Retail SDK includes base Microsoft code and code extensions. This base code and the code extensions can differ in each branch.
- The **RetailSdk-mirror** folder [4] is used to bring in Microsoft changes to the Retail SDK. It isn't used for development or build purposes. It should be updated only when a new version or hotfix is used.

For small projects, it's acceptable to have only two branches (Main = Dev branch). However, developers must be more disciplined, because any code submissions can immediately affect the quality of test builds.

You can build deployable packages out of multiple branches. In this case, you must have one build definition for each branch that can be built. The initial build definition is automatically created when a build environment is deployed (Main branch). You can make copies of the build for other branches. You must make small additions to incorporate the Commerce code.

The following high-level steps are used to set up an environment so that development work can begin. For details about the numbers in brackets, see the previous illustration and the related information.

1. Deploy a build environment and an empty Main branch in Microsoft Azure DevOps [1].
2. Deploy a development environment.
3. Create the Dev branch and the release branch (for example, ProdRel1 in the previous illustration) [2, 3].
4. Add the Retail SDK [4–7].

5. Prepare the development environment.
6. Optional: Deploy a second build environment for a different release branch.
7. Prepare the build definitions.

After you've completed all these steps, your branches, environment, and builds will be ready.

The following sections explain each step in detail.

Deploy a build environment and an empty Main branch in Azure DevOps

Use the LCS portal to deploy a new build environment. We recommend that you use a cloud-hosted environment, because you will have more options and capabilities if you have administrative rights. See the table about the various environment hosting models in the "Development Tier 1 environments" section, earlier in this article.

Start by creating a new Azure DevOps project if you don't already have one. In your Azure DevOps account, select **New project**.

Create new project

Projects contain your source code, work items, automated builds and more.

Project name *

 ✓

Description

Version control

Team Foundation Version Control ?

Work item process

Agile ?

Create Cancel

After you create the new Azure DevOps project, you must give Azure DevOps access to it. First, create a new personal access token on the Azure DevOps account. Then configure the LCS project with the correct URL and personal access token.

ⓘ Important

Do not disable "Classic build and classic release pipelines" in the Azure DevOps project.

Setup Visual Studio Team Services

1 Enter the Visual Studio Team Services site
Enter the Visual Studio Team Services site URL to allow Lifecycle Services to connect and manage resources.

2 Select the Visual Studio Team Services project
Choose the Visual Studio Team Services project in the selected site to link with this Lifecycle Services project.

3 Review and save
Review and save the Visual Studio Team Services settings for this Lifecycle Services project.

Enter a Visual Studio Team Services site URL to allow Lifecycle Services to connect and manage resources.

Example URL format accepted:
<https://org.visualstudio.com/>

Create a personal access token for the Visual Studio Team Services site and allow all authorized scopes. Personal access tokens are used instead of a password to allow Lifecycle services access the resources stored in your account.

To setup a personal access token, on the visualstudio.com site, click your name in the top right and select Security > Personal access tokens and create a new token for the Visual Studio Team Services account. You can get more information on how to create a person access token at:
<http://go.microsoft.com/fwlink/?LinkId=627398>

You will also need to choose a Visual Studio project. If you do not have a project, you can create one at
<https://www.visualstudio.com>

Visual Studio Team Services site URL:

Personal access token:

Privacy statement

Continue

After the LCS project is linked to Azure DevOps, you're ready to deploy.

Add a new environment, select the version, select **DEVTEST** as the topology, and select a build environment. On the next page, enter a meaningful name for the environment. Then enter a similar name for the build agent.

Deployment settings

Visual Studio Team Services >

Visual Studio Customization >

Supported version >

Customize SQL Database Configuration >

Disk space configuration >

Premium Storage Settings >

Customize virtual machine names >

Finance and Operations >

Customize virtual network >

VISUAL STUDIO TEAM SERVICES

Build Agent Name

Build Agent Pool (HINT: Must exist prior to deployment)

Branch Name

VSO CREDENTIALS (DEPRECATED: SET ACCESS TOKEN IN PROJECT SETTINGS)

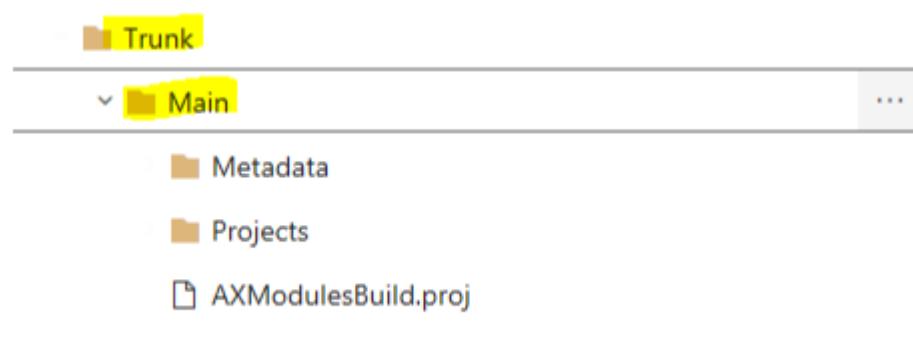
Alternate User ID

Alternate Password

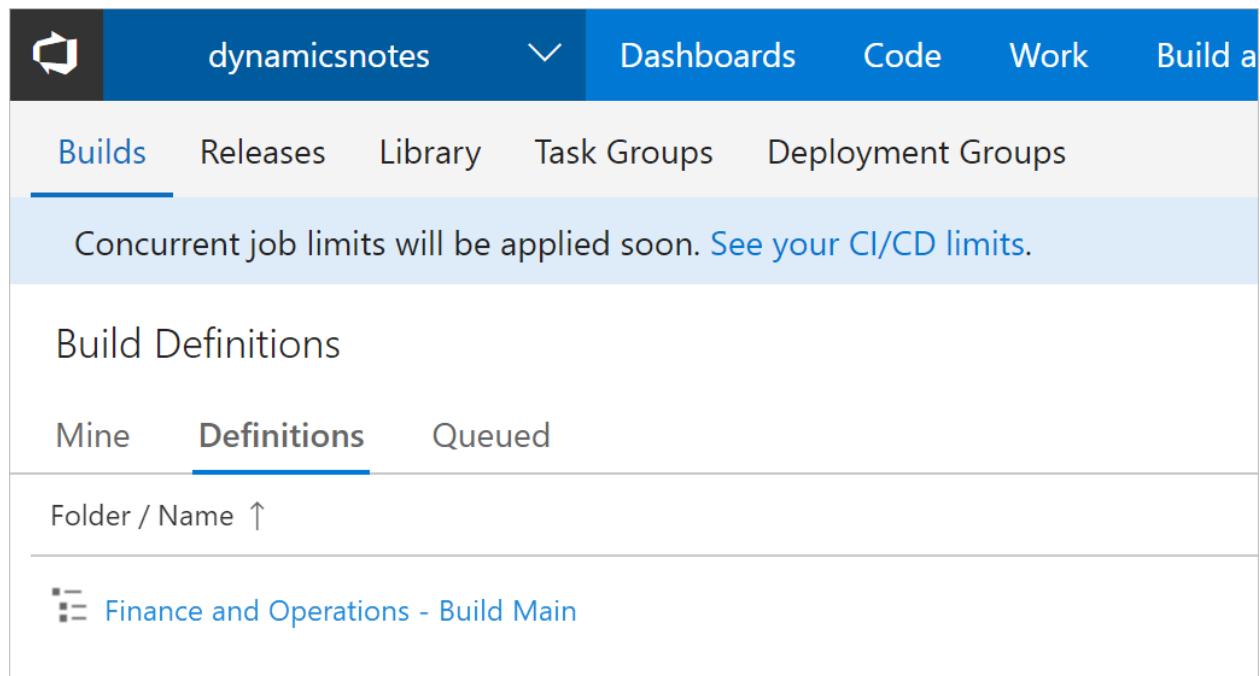
Continue

Next, under **Customize virtual machine names**, enter a unique name, and then deploy the VM.

The build box is deployed, and the build definition and Main branch are created, as shown in the following illustration. This process might take a couple of hours.



The build appears in the list of build definitions.

A screenshot of the 'Build Definitions' page in the Dynamics 365 interface. The top navigation bar includes a logo, the user name 'dynamicsnotes', and dropdown menus for 'Dashboards', 'Code', 'Work', and 'Build a'. Below the navigation is a secondary menu with links for 'Builds', 'Releases', 'Library', 'Task Groups', and 'Deployment Groups'. A message at the top states, 'Concurrent job limits will be applied soon. See your CI/CD limits.' The main content area is titled 'Build Definitions' and shows three tabs: 'Mine', 'Definitions' (which is underlined in blue), and 'Queued'. A search bar labeled 'Folder / Name ↑' is below the tabs. A single build definition is listed: 'Finance and Operations - Build Main'.

The build definition appears in the **Agents for pool Default** grid.

Agents for pool Default [Download agent](#)

Enabled	Name	State	Current Status
<input type="checkbox"/>	VSTAgent_RetailTC...	Offline	Idle
<input type="checkbox"/>	VSTAgent_Transact...	Online	Idle
<input type="checkbox"/>	VSTSagent-BLD17...	Offline	Idle
<input type="checkbox"/>	VSTSagent-SLD72...	Offline	Idle
<input type="checkbox"/>	VSTSagent-spring...	Offline	Idle
<input type="checkbox"/>	VSTSagent_D365F...	Offline	Idle
<input checked="" type="checkbox"/>	dynamicsnotesbuild	Offline	Idle

Deploy a development environment

Use the LCS portal of your implementation project to create a cloud-hosted development environment.

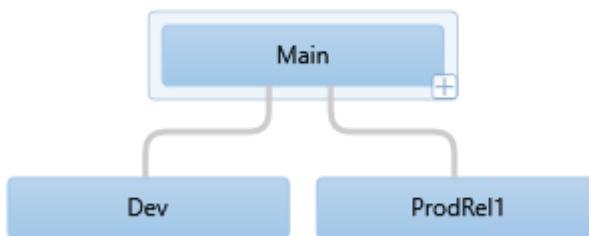
1. Make sure that you're signed in to the correct user account. This user account is used to create the tenant of the development machine. For example, if you're signed in to LCS as `lily@pad.com`, the environment is set up for the `@pad.com` tenant and expects users from that tenant. Although other users can be added, point of sale (POS) activation must be done by a user from that tenant. In some cases, user accounts from different domains can be used, such as when customers, partners, or other parties use email accounts from different domains. In these cases, coordination is required during POS activation, because only the tenant that was used during deployment can activate users.
2. Select the correct version, select **DEVTEST**, and then select **DEV**. Enter a meaningful and unique name, and make sure that the machine name is also unique in the advanced settings. The process of preparing the machine might take a couple of hours.

Because there is currently no Dev branch, you can skip the process of mapping Azure DevOps to the local directories. However, you will have to complete that process later.

Create the Dev and release branches

As previously mentioned, you must have a branch that holds changes that are often made but less often tested. You must also have a branch that holds the source code for production. The following illustration shows the expected hierarchy.

Main Branch Hierarchy



Follow these steps to create the branches.

1. Sign in to a development environment.
2. Start Microsoft Visual Studio as an administrator. Use an account that has access to the Azure DevOps project.
3. In Team Explorer, connect Visual Studio to the Azure DevOps project, if this connection doesn't already exist.
4. Map the **Trunk/Main** folder to a local folder (if this mapping doesn't already exist).
This mapping is temporary.
5. In Source Control Explorer, right-click the **Main** folder, and then select **Branching and Merging > Convert to Branch**.
6. Right-click the **Main** branch, select **Branching and Merging > Branch**, and name the new branch **Dev**.
7. Use **Pending Changes**, and submit this change to Azure DevOps.
8. Right-click the **Main** branch, select **Branching and Merging > Branch**, and name the new branch **ProdRel1**.
9. Use **Pending Changes**, and submit this change to Azure DevOps.

At this point, Source Depot Explorer in Visual Studio resembles the following illustration.



Add the Retail SDK

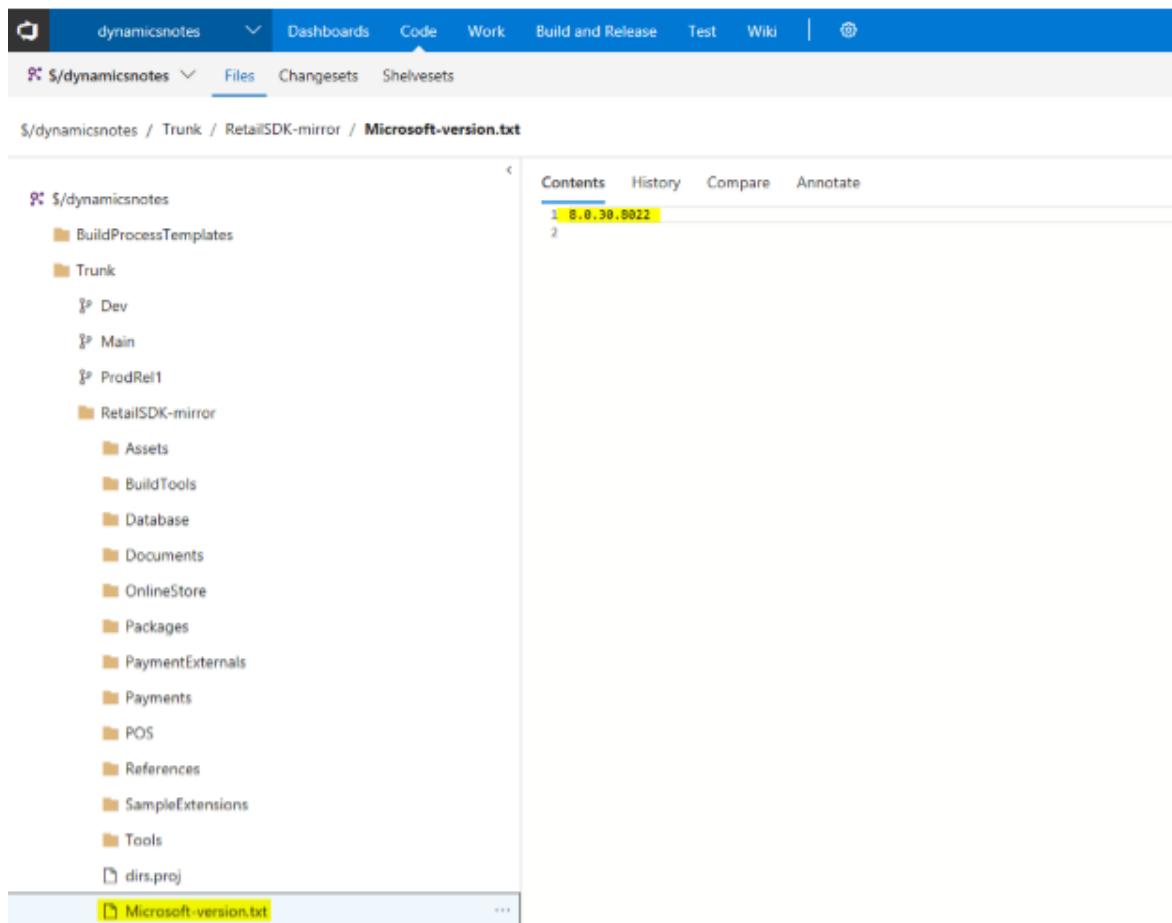
Next, you must add the Retail SDK to each of the three code branches, so that code changes can be propagated from Dev to Main and eventually to ProdRel1. This step also

enables separate changes between these branches, as for the X++ code. Therefore, we will have the Retail SDK in every branch, together with the X++ code.

First, add the mirror branch. The Retail SDK mirror branch is required as a baseline for code merges when updates from Microsoft are imported. The process for taking updates will be explained later in this article.

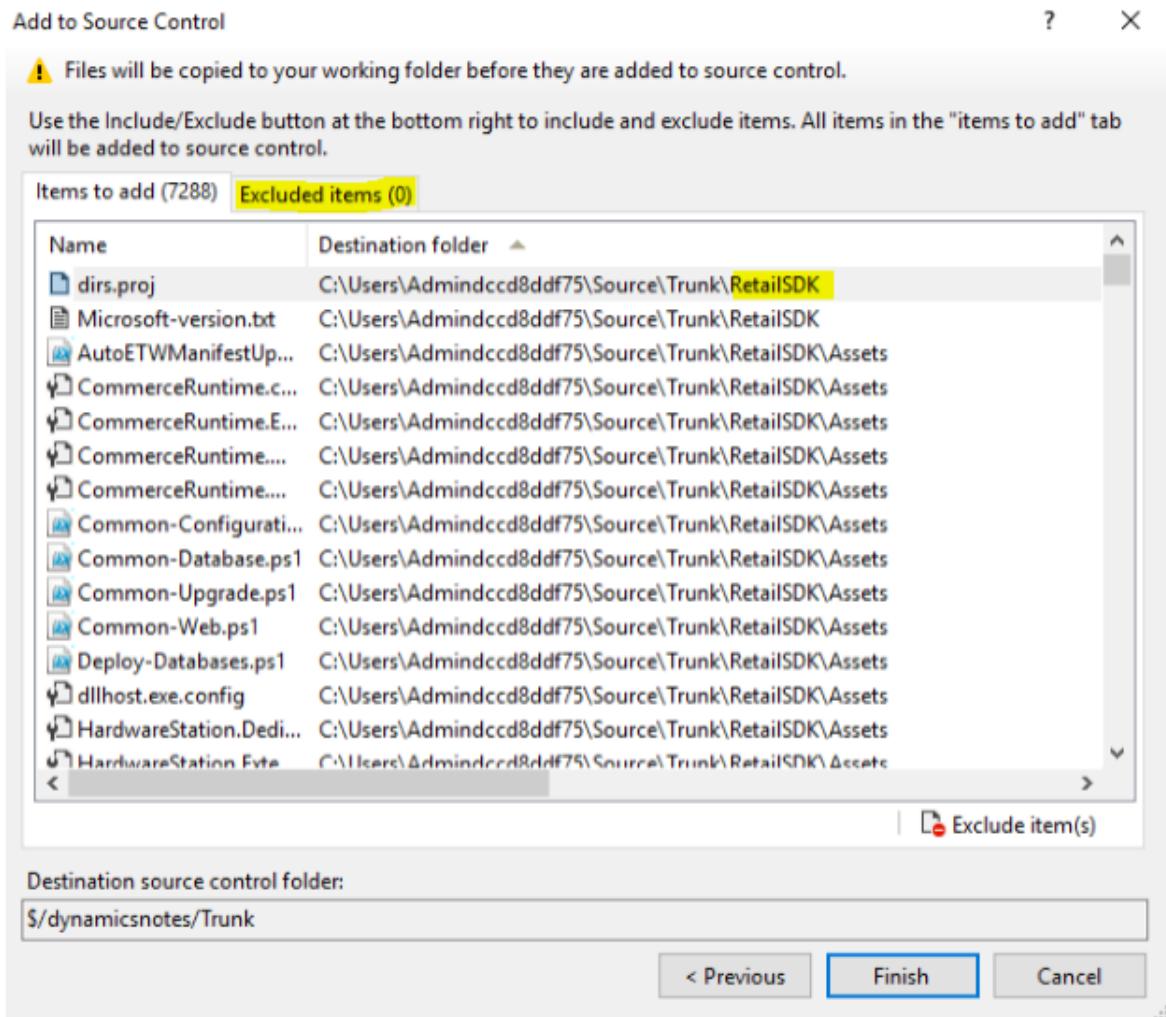
The mirror branch or folder is only required one time per project.

1. Find the unchanged Retail SDK that has the exact version that you want to start your development with. This Retail SDK can be found on every development machine on the service drive, or in every downloaded hotfix. You can uniquely identify a version of the Retail SDK by inspecting the Microsoft-version.txt file. This file should not be changed, except by an update to the Retail SDK mirror folder.



2. In Source Control Explorer, right-click the **Trunk** folder, and then select **Add Items to Folder**.
3. Select the top folder in the Retail SDK, and then select **Next**.
4. Visual Studio shows the number of files that will be added. Make sure that the **RetailSdk** folder is under the **Trunk** folder.

5. Make sure that there are 0 (zero) excluded items by selecting items and then selecting **Include items**.

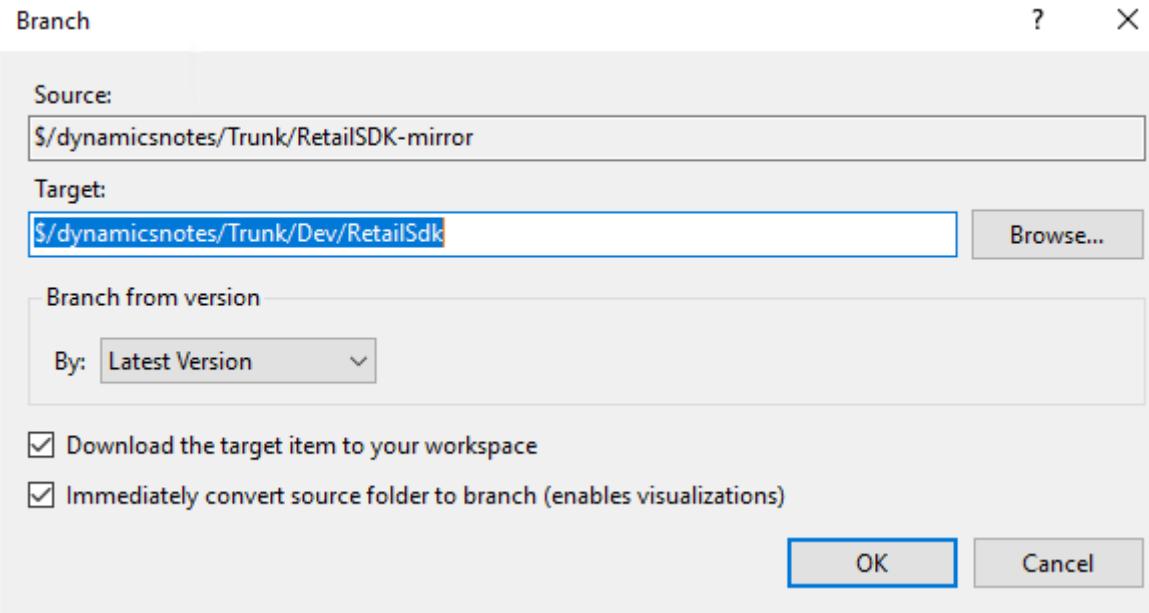


6. Select **Finish**. This process will take a few minutes.

7. When the process is completed, rename the folder **RetailSdk-mirror**.

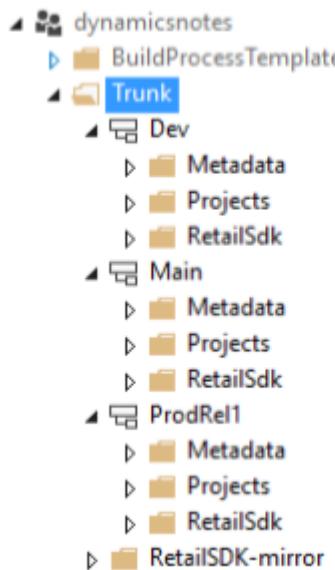
Next, you must branch to each branch. Follow the same path that the code changes will flow in: first to Dev, then to Main, and then to ProdRel1.

1. Select the folder for the mirror branch, right-click, and then select **Branching and Merging > Branch**.
2. Go to the **Dev** branch, append **/RetailSdk** to the name, and then select **OK**.



3. Use **Pending Changes**, and submit the changes.
4. Follow the same steps to branch the **RetailSdk** folder of the **Dev** branch to the **Main** branch.
5. Follow the same steps to branch the **RetailSdk** folder of the **Main** branch to the **ProdRel1** branch.

At this point, you have the code branches and code locations for the X++ and Commerce extensions setup. In Source Control Explorer, the file structure should resemble the following illustration.



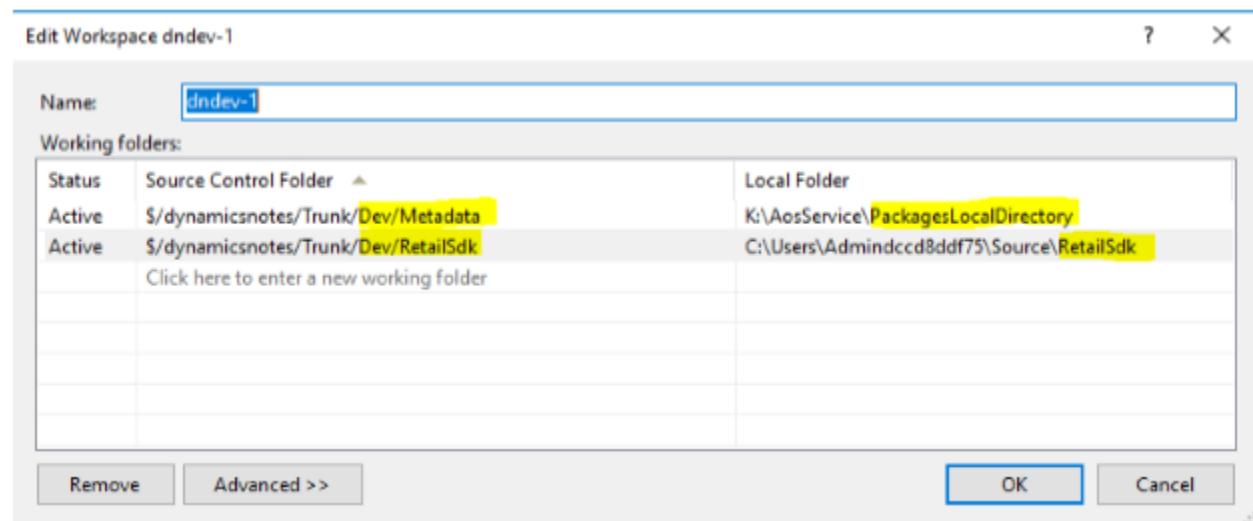
You should also change the version of the Commerce customization. This version should differ in the Dev, Main, and ProdRel1 branches. Either change the `Customization.settings` file, or add a new `global.props` file in the **RetailSdk\BuildTools** folder. For example, you can number Dev as 1.0.0.x, Main as 1.0.1.x, and ProdRel1 as 1.0.2.x.

Prepare the development environment

You can now prepare the development environment for Commerce development tasks. The development environment will map the code locations for both X++ and the Retail SDK in the Dev branch to local folders. The Metadata folder (X++) must be always mapped to the PackagesLocalDirectory folder. The location of the RetailSdk folder must follow these guidelines:

- The location should be somewhere inside the local user's folder.
- The file path of any file is limited to 256 characters. Therefore, use a short path for the root of the Retail SDK. For example, you can use `c:\users\<user name>\Source\RetailSdk`.

To map X++ and the Retail SDK, you must edit the current workspace. Select **Pending Changes** > **Actions** > **Workspaces**, and update the current workspace so that it resembles the following illustration. As was previously mentioned, you should map the Metadata folder of the branch to the PackagesLocalDirectory folder and the RetailSdk folder to a short folder of your choice.



The download of the files can take a few minutes.

Regardless of whether there are customizations in the code branches, the following steps prepare your development box so that you can write and run code. Some steps are optional, depending on the customizations that are planned.

1. Install your favorite development tools. For information about one automated script, see [Auto-Installing most needed dev tools in 5 mins with Chocolatey](#).
2. To help reduce the compile time, exclude the code folders from Microsoft Windows Defender.

3. If there is already code in the **Dev/Metadata** folder, build all Commerce models.
4. To speed up the development experience, switch to Microsoft Internet Information Services (IIS). For instructions, see [MSDyn365FO. How to switch from IIS Express to IIS on development VM](#). This step can be done only on the Tier 1 VM where you have administrative privileges (cloud-hosted environment).
5. Optional: Restore a recent copy of a production database that has good data.
 - a. Rename the existing database **AxDB_Orig**.
 - b. In Microsoft SQL Server Management Studio, restore the .bak file. (If a .bacpac file exists, follow the steps in [Copy a database from Azure SQL Database to a SQL Server environment](#).)
 - c. In Visual Studio, refresh the model store.
 - d. In Visual Studio, do a full build if the source and destination environments of the database are on different versions.
 - e. In Visual Studio, run a full database synchronization.
 - f. Make sure that the Batch service is running.
 - g. Run the Environment reprovisioning tool. (Find the latest version in the LCS Asset library, and deploy it by using the **Maintain** function.)
 - h. Verify that the tool succeeded. The following query should show the URLs of all local development machines that were updated.

SQL

```
select * from dbo.RETAILCHANNELPROFILEPROPERTY where ISSYSTEMRECORD = 1
```

- i. In Commerce, run the **Initialize Commerce Scheduler** job to delete old data.
6. Make sure that you can sign in to Commerce by using your user account. If you aren't the Admin user in the production database, run the Admin provisioning tool to take ownership. (This tool is in the **PackagesLocalDirectory/bin** folder.)
7. Verify that Commerce Data Exchange (CDX) data synchronization works. In Commerce, go to **Download sessions**. You should see many applied sessions. If you don't see them, select job **9999**, and run it.

8. Install TypeScript version 2.2.2 from

<https://www.microsoft.com/download/details.aspx?id=48593>.

9. Do a full build of the Retail SDK from a command prompt.

a. Open an MSbuild command prompt for Microsoft Visual Studio 2015 as an administrator.

b. Change the directory to the location of your Retail SDK on the local VM.

c. Type **msbuild**, and then press Enter. The build should succeed.

10. Add the development/sample Retail Modern POS (MPOS) certificate to the local machine's trusted root certificate store:

`...\\RetailSDK\\BuildTools\\ModernPOSAppxSigningCert-Contoso.pfx`. Set the password to an empty string.

11. Install MPOS or MPOSOffline by running the installer at `...\\RetailSDK`

`\\References\\YourCompany\\Contoso.ModernPOSSetupOffline.exe`. You must complete this step one time to deploy the ClientBroker files.

12. In Visual Studio, open **ModernPOS.sln** (as an administrator), and do a full rebuild.

13. Press F5 to start MPOS in the debugger.

14. In Commerce, open the **Channel profiles** page, and copy the Commerce Scale Unit URL for the default channel profile.

15. Open a browser window, and paste the URL into the address bar. You should be able to browse to your local Commerce Scale Unit.

16. In Commerce, add external user credentials to any worker (for activation), save the password, and don't allow a password reset on first sign-in.

17. In Commerce, run job **1060 (AX/Distribution schedule)**.

18. Activate MPOS by using the same Azure Active Directory (Azure AD) user that you added in step 16. Paste the Commerce Scale Unit URL, select a store and a register, and finish the activation.

You should now be able to run MPOS in the debugger from your local sources.

The process of preparing a development environment is now completed. At this point, any extension code (X++, Commerce runtime [CRT], Commerce Scale Unit, channel SQL, and POS) can be written, debugged, tested, and submitted to Azure DevOps.

Optional: Deploy a second build environment for a different release branch

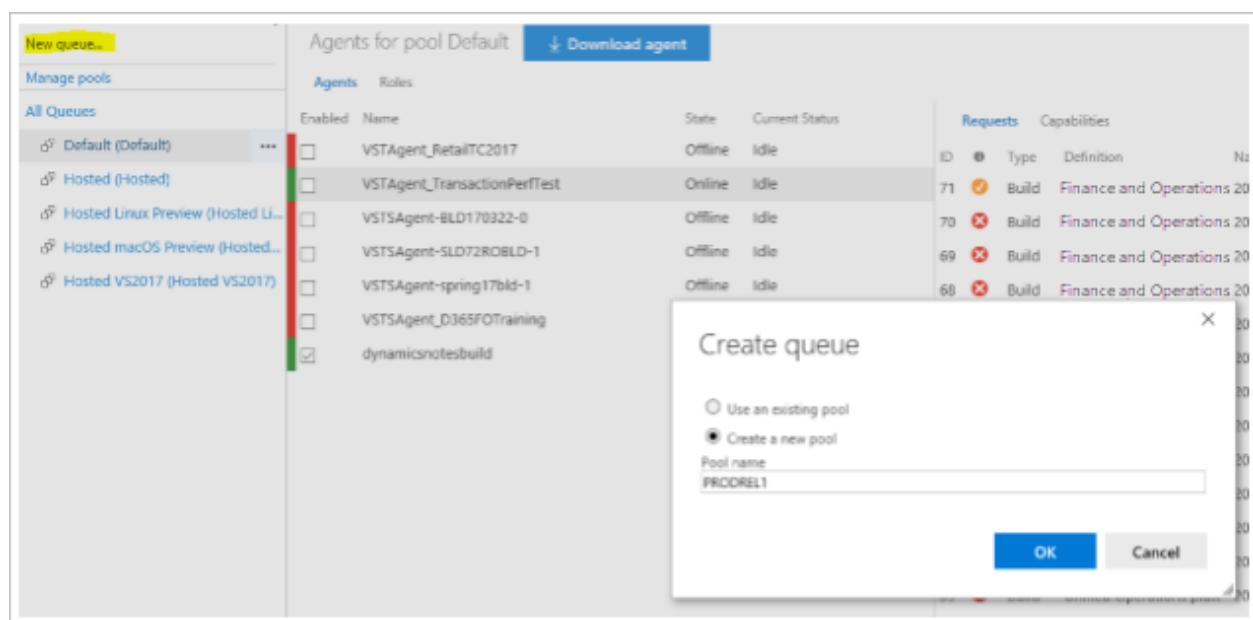
If you must maintain multiple releases at the same time, you must create deployable packages from different code branches (for example, Main2 or Main3, and/or ProdRel1 or ProdRel2).

The steps to set up a second build environment are the same as the steps for the first build environment. At this point, an Azure DevOps project, and the link between the LCS project and the Azure DevOps project, already exist.

To separate the build environments, we recommend that you create a new Azure DevOps agent queue for the release branch. Although there are ways to share an agent queue (and its build environment) for multiple branches, this approach can be tricky.

Currently, the build environment must be on the same platform and binary hotfix version as the target environment during deployment. Otherwise, LCS might reject the deployable package because of version incompatibility.

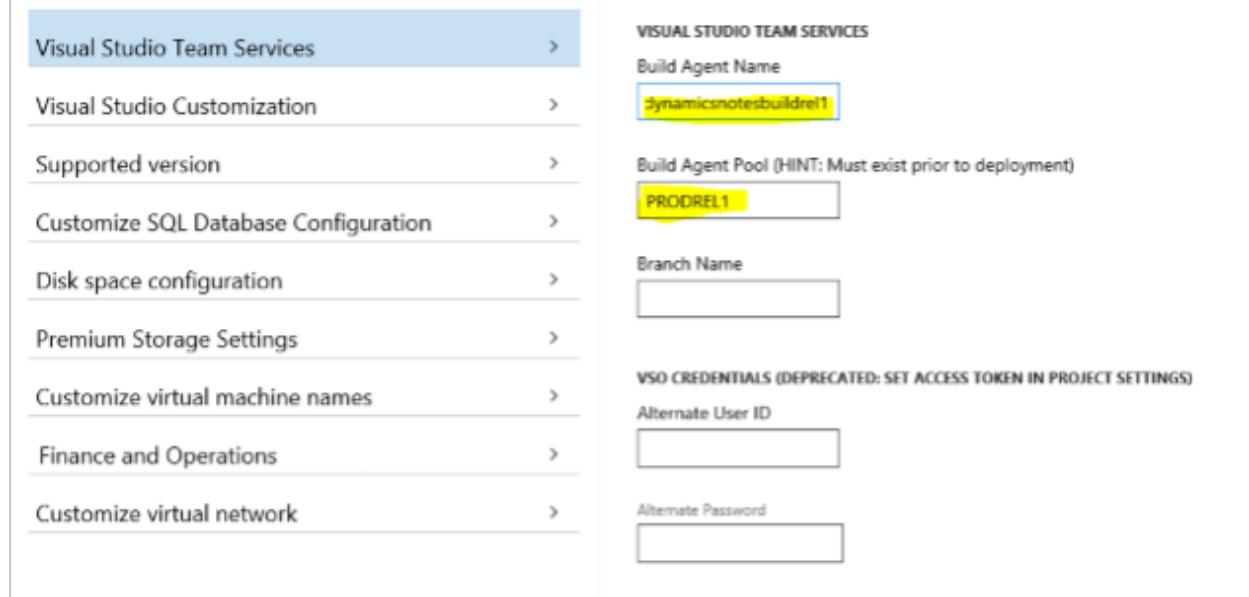
First, create a new Azure DevOps agent queue.



The screenshot shows the 'Agents for pool Default' page in the Azure DevOps interface. On the left, a sidebar lists 'All Queues' with items like 'Default (Default)', 'Hosted (Hosted)', and 'Hosted VS2017 (Hosted VS2017)'. On the right, a table lists agents: VSTAgent_RetailTC2017 (Offline, Idle), VSTAgent_TransactionPerfTest (Online, Idle), VSTSagent-BLD170322-0 (Offline, Idle), VSTSagent-SLD72ROBLD-1 (Offline, Idle), and VSTSagent-spring17bld-1 (Offline, Idle). A 'Create queue' dialog box is open in the foreground, showing options for 'Use an existing pool' (radio button) and 'Create a new pool' (radio button, selected). The 'Pool name' field contains 'PRODREL1'. At the bottom are 'OK' and 'Cancel' buttons.

When you deploy from LCS, use **PRODREL1** as the name of the agent pool.

Deployment settings



Visual Studio Team Services >

Visual Studio Customization >

Supported version >

Customize SQL Database Configuration >

Disk space configuration >

Premium Storage Settings >

Customize virtual machine names >

Finance and Operations >

Customize virtual network >

VISUAL STUDIO TEAM SERVICES

Build Agent Name

dynamicsnotesbuildrel1

Build Agent Pool (HINT: Must exist prior to deployment)

PRODREL1

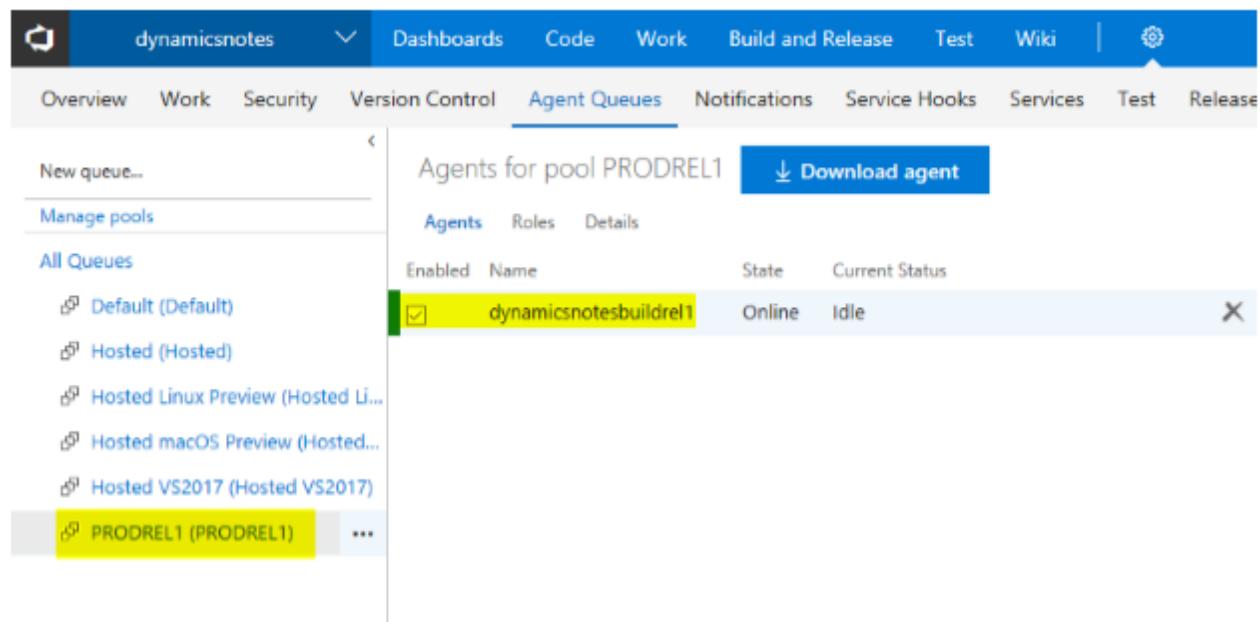
Branch Name

VSO CREDENTIALS (DEPRECATED: SET ACCESS TOKEN IN PROJECT SETTINGS)

Alternate User ID

Alternate Password

Next, on the **Customize virtual machine names** tab, enter a unique name, and then deploy the new build. The process of deploying a new build and creating a new agent queue can take a couple of hours.



dynamicsnotes > Agent Queues

New queue...

Manage pools

All Queues

- Default (Default)
- Hosted (Hosted)
- Hosted Linux Preview (Hosted Li...)
- Hosted macOS Preview (Hosted...
- Hosted VS2017 (Hosted VS2017)
- PRODREL1 (PRODREL1)

Agents for pool PRODREL1

Agents for pool PRODREL1

Download agent

Enabled	Name	State	Current Status
<input checked="" type="checkbox"/>	dynamicsnotesbuildrel1	Online	Idle

Prepare the build definitions

After you complete the steps earlier in this article, you should have one build definition and two agent queues, and each agent queue should have one agent. To build different branches, you must configure the build definition differently. Therefore, you must clone the build definition.

However, before you clone the build definition, you must add the Retail SDK into the build, so that you don't have to complete this step twice. To edit the existing build

definition, which is named **Unified Operations platform - Build Main**, follow the steps in [Integrate the Retail SDK with the continuous build system \(Azure DevOps\)](#) to integrate the Retail SDK into the metadata build of the Main branch.

If you had multiple build branches and environments, just clone the build definition, and name the new build definition so that it's clear which branch it's for. (The clone feature is available in the Azure DevOps portal). Select the new agent queue that you created, and change the following paths in any build steps or source mappings. (In the paths, change **Main** to **ProdRel1**.)

- Source mappings
- Retail SDK build step
- Retail SDK copy binaries step
- Build the solution step (X++ build)
- Retail SDK copy packages step

Tips

- You can speed up an official build by making these changes in the **Variables** section of the build definition:
 - Set **DeployReports** to 0.
 - Set **SkipSourcePackageGeneration** to 1.
- Change the version of the Commerce customization in each branch. The version should be different in the Dev, Main, and ProdRel1 branches. Either change the **Customization.settings** file, or add a new **global.props** file under the **RetailSdk\BuildTools** folder. You can use any kind of numbering for the file name. For example, you can number Dev as 1.0.0.x, Main as 1.0.1.x, and ProdRel1 as 1.0.2.x.
- For efficiency, shut down build or development environments when they aren't being used.
- If you're using cloud-hosted Tier 1 development environments (where you have administrative privileges), you can switch from IIS Express to IIS. Using IIS for running all web application is more robust, more performant, and avoids the switching. For more information, see [MSDyn365FO. How to switch from IIS Express to IIS on development VM](#).
- For prototyping purposes, a developer might want to change the Retail SDK on a development VM without using Azure DevOps source control. Always keep the original Retail SDK untouched, and make a copy that you can work in temporarily.

In that way, you can take the unchanged Retail SDK into your mirror branch later, if it's required.

- Currently, a build environment must be on the same platform and binary hotfix version as the target environment.

Additional resources

[Update code and environments for Retail projects](#)

[Testing and performance issues](#)

Configure and install Commerce Scale Unit (self-hosted)

Article • 01/31/2024

This article explains how you can use self-service to configure a Commerce Scale Unit in Microsoft Dynamics 365 Commerce headquarters, download it, and install it on one or more computers in a brick-and-mortar store. Commerce Scale Unit (CSU) combines the Commerce channel database, Commerce async client, Retail Server, and Cloud point of sale (POS) components. A Commerce environment already provides these components in the cloud. However, you can now configure them so that they work locally in a store or datacenter, in either a single-computer setup (the default option) or a multiple-computer setup. This article also explains how to uninstall and troubleshoot Commerce Scale Unit.

ⓘ Important

- A basic design principle to follow is that if you are not able to customize in a requested manner on a Commerce Scale Unit (Cloud), you should not customize this way with a CSU (self-hosted). It is critical to understand that direct database access is not supported and can easily cause breaks in customizations that use this concept. A CSU (self-hosted) is primarily for enabling cross-terminal scenarios, reducing latency or backup for poor WAN connectivity, and providing scale-out to spread the load of POS terminals across multiple CSU components.
- Do not install a CSU on a developer environment, which typically already has a configured Retail Server and Channel database.

ⓘ Important

- It's critical to note that this component utilizes a server certificate in addition to Azure Service-to-Service authentication. Both the generated Azure web application keys (formerly called *secrets*) and the server certificate must be managed for expiration. By default, a certificate and a generated Azure web application key expires in one calendar year (365 days).
- It's critical that you have a plan to rotate this key at least one month prior to expiration. Planning is necessary when working with a high number of stores

to ensure that there is sufficient time to roll out the change to all stores.

Before you begin

Important

To help maintain a high level of security across the company, we strongly recommend that you create a new application ID (client ID) and key (secret) for each store that is created. This step requires a new Web app.

1. Generate a Microsoft Azure Active Directory (Azure AD) app registration to create an application ID (client ID) and key (secret). For instructions, see [Create an Azure Active Directory application](#). This article reviews Azure user permissions and requirements, and explains how to generate an app registration.

Important

If you are installing Commerce Scale Unit for use with an on-premises environment using Active Directory Federation Services, instead of Azure, follow the instructions in the Commerce installation document for on-premises environments. For more information, see [Installation steps for Commerce channel components in an on-premises environment](#).

2. After an application ID (client ID) and key are created for Commerce Scale Unit, the client ID must be accepted in Commerce. Go to **System administration > Setup > Azure Active Directory applications**. Enter the application ID (client ID) in the **Client ID** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.

Configure a new Commerce Scale Unit

To create a functioning Commerce Scale Unit, complete the procedures in all sections of this article until the "Multiple-computer installation" section. To complete the configuration and installation, you must first do the initial configuration in headquarters. Next, you must complete the installation. Finally, you must return to headquarters to finish the configuration, so that Commerce Scale Unit works correctly.

Important

Channel functionality in an on-premises environment is enabled exclusively via use of Commerce Scale Unit (self-hosted). For an overview, see [Commerce Scale Unit \(self-hosted\)](#). Unlike a cloud deployment, an on-premises environment does not enable seamless, high-availability deployment of channel components via Lifecycle Services (LCS). The only way to use channel components is by installing Commerce Scale Unit (self-hosted).

For on-premises deployments, perform the following steps:

1. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database group**.
2. On the Action pane, select **New**.
3. In the **Name** field, enter **Default**. Enter a description in the **Description** field, if needed.
4. In the **Channel schema** field, select **AX7**.
5. In the **Working folders** field, select **File storage**.
6. On the Action Pane, select **Save**.

1. In headquarters, go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database**.
2. On the **Channel database** page, on the Action Pane, select **New**.
3. In the **Channel database ID** field, enter a unique value.
4. In the **Channel data group** field, select the **Default** option. Select any option that has been created.

ⓘ Important

For on-premises deployments, this value will be the **Default** value that was previously described in this article.

5. In the **Type** field, leave the default value (**Channel database**) selected.
6. You can leave the **Data sync interval** field blank. Alternatively, you can select a value in this field. For example, in the demo data, the value **D60-U15** specifies a 15-minute synchronization interval.

The **Data sync interval** value determines how often the data is synchronized between the channel database (Commerce Scale Unit) and headquarters. If no

value is entered, the default interval that is set up in Commerce Scale Unit is used. This default interval is three minutes.

7. On the **Commerce channel** FastTab, select **Add**, and then, in the **Channel** field, select the appropriate store channel. Repeat this step to add all the channels that should use this database.

You can also add channels that don't use this database. In this way, you keep the data for those channels in the Commerce Scale Unit channel database. The channels that actively use this database can then access that data locally.

ⓘ Important

For on-premises deployments, select the **Download** button on the Action pane and select **Commerce Scale Unit package**. This will cause a known error and initiate the upload logic so that the following step in this document can be correctly completed. Allow for at least one minute to pass while the upload logic completes.

8. On the **Commerce Scale Unit package** FastTab, in the **Package reference** field, select the appropriate Commerce Scale Unit package. Each environment generates a base Commerce Scale Unit package. Therefore, this field always contains at least one option.
9. On the Action Pane, select **Save**.
10. Go to **Retail and Commerce > Channel setup > Channel profiles**.
11. On the Action Pane, select **New**.
12. In the **Name** field, enter a unique name for the channel profile.

ⓘ Important

For on-premises deployments, any value can be entered in this field, however, the value **Default** is common.

13. On the Action Pane, select **Save**.
14. On the **Profile properties** FastTab for the new channel profile, select **Add**.
15. In the **Property key** field, select **Retail Server URL**.

16. In the **Property value** field, enter the URL of the Retail Server that will be installed.

The standard format for the URL of Commerce Scale Unit is `https://<Computer Name>:<Port>/RetailServer/Commerce`. In this format, **<Computer Name>** is either the fully qualified domain name (FQDN) of the computer where Commerce Scale Unit is installed or, for systems that aren't joined to a domain, the full computer name. **<Port>** is the port number that should be used in the installation. The port number must be a value between 1 and 65535. If you're using the default HTTPS port (443), you don't have to specify the port number.

17. On the **Profile properties** FastTab for the new channel profile, select **Add**.

18. In the **Property key** field, select **Cloud POS URL**.

19. In the **Property value** field, enter the URL of the Cloud POS instance that should be installed for Commerce Scale Unit.

The standard format for the URL of Cloud POS is `https://<Computer Name>:<Port>/POS`. In this format, **<Computer Name>** is either the FQDN of the computer where the Commerce Scale Unit is installed or, for systems that aren't joined to a domain, the full computer name. **<Port>** is the port number that will be used in the installation. The port number must be a value between 1 and 65535. If you're using the default HTTPS port (443), you don't have to specify the port number.

20. On the Action Pane, select **Save**.

 **Note**

If media is commonly used, it will be necessary to generate a **Media Server Base URL** for the profile. For testing and simplicity, the URL that exists for the **Default Channel** profile can be reused.

For on-premises deployments, the **Media Server Base URL** will be where all media is stored for POS devices.

21. Go to **Retail and Commerce > Channels > Stores > All stores**.

22. Select the channel ID for the store that will use the new channel database.

23. On the details page for the selected store, on the Action Pane, select **Edit**.

24. On the **General** FastTab for the store, in the **Live channel database** field, select the channel database that you created in step 3.

25. On the Action Pane, select **Save**.
26. On the **General** FastTab for the store, in the **Channel profile** field, select the channel profile that you created in step 12.
27. Go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel data group**.
28. Select the **Default** data group, and then, on the Action Pane, select **Full data sync**. In the **Select a distribution schedule** field, select job **9999**, and then select **OK**. In the dialog box that appears, select **OK** to confirm the full synchronization. All the data in the channel database is prepared for download.

 **Important**

For on-premises deployments, there is no **Default** channel data group. Create a new data group (and associate it to the channel database and distribution schedule jobs).

Download the Commerce Scale Unit installer

1. In headquarters, go to **Retail and Commerce > Headquarters setup > Commerce scheduler > Channel database**.
2. In the list of channel databases on the left, select the channel database that you created earlier.
3. On the Action Pane, select **Download**.
4. On the drop-down menu, select **Configuration file**.

 **Note**

To ensure that the Commerce Scale Unit installer correctly uses the configuration file (XML file), you must save the configuration file to the same location as the installer. If the configuration file is not the same file name as the installer executable, either the executable must be run using the command line to specify the configuration file or you need to rename the XML configuration file to have the same base name as the executable file name.

For on-premises deployments, the configuration file (at this time) requires manual editing:

- `StoreSystemAosUrl` should have the value used to access headquarters (AX). It is critical to keep a trailing slash at the end of this URL (for example, `https://myContosoURL.com/namespaces/AXSF/`).
- `AADTokenIssuerPrefix` should have the value `https://NOTUSED.microsoft.com`
- `TransactionServiceAzureAuthority` should have the value `https://<ADFS FQDN including .com>/adfs`.
- `TransactionServiceAzureResource` should have the base URL value of the `StoreSystemAosUrl` edited as shown above. For instance, based on the above example `https://myContosoURL.com` would be used as the value, removing the `/namespaces/AXSF/` portion of the URL.

5. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)

Browsers might block the download pop-up that is generated. Select either **Allow once** or **Options for this site > Always allow**. Then select **Download** again.

6. On the Action Pane, select **Download**.
 7. On the drop-down menu, select **Commerce Scale Unit package**.
 8. On the Notification bar that appears at the bottom of the Internet Explorer window, select **Save**. (The Notification bar might appear in a different place in other browsers.)
- The correct installation package is automatically selected for download, based on the Commerce Scale Unit package on the selected channel database.
9. After the setup installer has been saved, on the Notification bar, select **Run**. (This step might differ, depending on the type of browser.)

Run the Commerce Scale Unit installer

Note

- If the CSU (Self-hosted) installer is being run to update a current installation, verify that the version is the same or newer. If an earlier version is required to be installed for any reason, it is critical to uninstall the current CSU (Self-)

hosted) and delete the CSU SQL database. If you do not delete the database in the scenario where an earlier version is required, the database schema and data will be of a newer version than the CSU installation and errors and issues will occur.

- Do not install a CSU on a developer environment, which typically already has a configured Retail Server and Channel database.
- Before running the Commerce Scale Unit (self-hosted) installer, verify that the configuration file is named the same as the installer executable. This would look like **ExecutableInstallerName.xml** and put both files in the same folder. Alternatively, there is a command line delimiter to specify the configuration file manually.
- If you plan to install and use Retail Cloud POS, you must initialize the configuration the first time that you run the installer, as described in the following procedure.

Before you run the Commerce Scale Unit installer, make sure that all [system requirements](#) are met.

Important

If you are installing Commerce Scale Unit for use with an on-premises environment, you must start it from a command line using administrator privileges as follows:
`StoreSystemSetup.exe -UseAdfsAuthentication`

The Commerce Scale Unit installer first extracts the associated files. It then begins the installation.

1. On the first page of the installer, select the components to install. You can install the following components:

- Commerce channel database together with Async Client
- Retail Server
- Cloud POS

Note

- To install Cloud POS, you must also select and install Retail Server.

- By default, the installer installs all components on one computer. To install the components across multiple computers, you must complete additional manual steps. For more information, see the "Multiple-computer installation" section.

After you've selected all the components to install, select **Next** to continue.

2. The installer validates that all prerequisites are met. If a valid version of Microsoft SQL Server isn't found, the installer will fail during the prerequisites check. Install a supported version of SQL Server and try the installer again.

 **Note**

- To meet the prerequisites, SQL Server must have full-text search, and it must support, at a minimum, Transport Layer Security (TLS) 1.2. Review the system requirements for the supported versions of SQL Server. It is highly recommended to review [SQL Server versions and licenses](#).
- If a system restart is required, the installer will prompt the user. This prompt is based upon a Windows system registry key that notifies all applications when a restart is required. While it is recommended to restart prior to continuing the installation, a restart is not required and the installer can continue without restarting the computer.

3. Verify the URL for Application Object Server (AOS), and then select **Next**. (The AOS URL is the URL that is used to access headquarters.)

 **Note**

The Retail Server URL is automatically entered from the configuration file.

4. Select a valid Secure Sockets Layer (SSL) certificate to use for HTTPS communication.

- The SSL certificate must use private key storage, and server authentication must be listed in the enhanced key usage property.
- The SSL certificate must be trusted locally, and it can't be expired.
- The SSL certificate must be stored in the personal certificate store location on the local computer.

- The SSL certificate must contain the following `keyUsage` property values: `digitalSignature`, `keyEncipherment`, and `dataEncipherment`.
5. If a specific user is required, enter the user name and password that the application pool should run under. By default, the installer automatically generates a service account to use. This approach is more secure and is recommended.

 **Note**

It is important to note that service accounts, out of box, still function under the same password policy that is defined for all other accounts. This means that the minimum password age policy still applies to the Retail Server service account and must be updated when necessary. By default, on Windows Server 2012 R2, this is typically 42 days. If the password does expire on a used service account, the Commerce Scale Unit components will fail to continue functioning until the issue is resolved.

6. On the next page, enter the user account and password for the Retail Server application pool and Async Client. By default, this account is automatically generated. However, you can manually enter the user account and password.
7. Enter the HTTPS port to use, and verify that the host name of the computer is correct. Then select **Next** to continue.

 **Note**

- The installer automatically enters the host name. If, for any reason, the host name must be changed for the installation, change it here. The host name must be the FQDN of the system, and it must be entered in the **Host name** field for the selected Store system entry.

8. Enter the application ID (client ID) and key (secret) that are associated with this Commerce Scale Unit installation. Additionally, verify the channel database ID, which is automatically entered from the configuration file. Then select **Install**. If you will use Retail Cloud POS, make sure that the **Configure Retail Cloud POS** check box at the bottom of the page is selected. This configuration requests Azure AD sign-in and automatically generates all required information in Azure, so that Retail Cloud POS can be used on-premises. If you are installing Commerce Scale Unit for use with an on-premises environment, you must clear this option.

For information about how to create web applications in Azure, see [Create an Azure Active Directory application](#).

ⓘ Important

- When installing Commerce Scale Unit for use with an on-premises environment, Cloud POS does not require an Azure or AD FS application to be configured, so it is important to unmark **Configure Retail Cloud POS**.
- When installing Commerce Scale Unit for use with an on-premises environment, the Client ID (Application ID) and Secret (Key) used will be the values generated by the PowerShell script performed in the configuration steps performed in steps 6-8 in the [Installation steps for Commerce channel components in an on-premises environment](#) article. (Step 6 creates the Client ID and step 8 resets the Secret to be copied.)

When you create the Web App, the initial URI and URL don't have to be any specific value. Only the application ID (client ID) and key (secret) that are created are important.

9. After the application ID (client ID) and key (secret) are created for Commerce Scale Unit, the application ID (client ID) must be accepted in Commerce. Follow the next procedure to finish the configuration in headquarters.
10. After the installation is complete, the final health page appears. This page shows whether the installation was successful. It also shows the health of each component, based on basic connection tests, and the location of this article. If the installation wasn't successful, the page shows the location of the log files. We recommend that you keep this final health page open until you've completed the configuration of Commerce Scale Unit and all components are working correctly (Requiring the completion of the following section).

Finish the configuration in headquarters

The last steps require validation and verification that the Azure application ID (client ID) and key (secret) are correctly accepted in headquarters, so that connections can be made between the environment and the new Commerce Scale Unit.

1. After the application ID (client ID) and key (secret) are created for Commerce Scale Unit and entered in the installer, the application ID (client ID) must be accepted in headquarters.

In headquarters, go to **System administration > Setup > Azure Active Directory applications**. Enter the application ID (client ID) in the **Client ID** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.

2. If Cloud POS is configured for use, a client ID is shown at the end of the installation. You must add this client ID to the **Commerce shared parameters** page in Commerce.

ⓘ Important

In an on-premises environment, this step is not required to be completed.

- a. In headquarters, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
 - b. Select **Identity providers**.
 - c. On the **Identity providers** FastTab, select the provider that begins with `HTTPS://sts.windows.net/`. The values on the **Relying parties** FastTab are set, based on your selection.
 - d. On the **Relying parties** FastTab, select **Add**. Enter the client ID that is listed on the final health page of the Commerce Scale Unit installer. Set the **Type** field to **Public** and the **UserType** field to **Worker**. Then, on the Action Pane, select **Save**.
 - e. Select the new relying party, and then, on the **Server resource IDs** FastTab, select **Add**. In the **Server Resource ID** column, enter `https://retailstorescaleunit.retailserver.com`.
 - f. On the Action Pane, select **Save**.
3. In headquarters, go to **Retail and Commerce > Headquarters setup > Parameters > Commerce shared parameters**.
 4. Select **Identity providers**.
 5. On the **Identity providers** FastTab, select **Add**.
 6. In the new **Issuer** row, enter the URL of the newly installed Commerce Scale Unit. At the end of the URL, add `/auth`. The URL will resemble `https://<My Case-Sensitive Computer Name>:<Port Number>/RetailServer/auth`.

ⓘ Note

The URL described above is case sensitive. There will be a new identity provider line for each Commerce Scale Unit that is installed. Each will have a URL that resembles this URL.

7. In the **Name** column, enter a description for the store that the URL belongs to.

8. In the **Type** column, select **Open ID Connect**.

 **Note**

This new row must be duplicated for every Commerce Scale Unit installation (that is, for every unique URL).

9. On the Action Pane, select **Save**.

10. On the **Identity providers** FastTab, select the newly created line. The values on the **Relying parties** FastTab are set, based on your selection.

11. On the **Relying parties** FastTab, select **Add**, and add the following two entries:

- In the **ClientId** column, enter **Cloud POS**. Set the **Type** field to **Public** and the **UserType** field to **Worker**.
- In the **ClientId** column, enter **Modern POS**. Set the **Type** field to **Public** and the **UserType** field to **Worker**.

12. On the Action Pane, select **Save**.

13. In Commerce, go to **Retail and commerce** > **Retail and commerce IT** > **Distribution Schedule**, and run CDX Job 1110.

14. When you've finished, return to the installer, and select **Finish**.

The final page of the installer includes valuable information that you can use to test and validate that all components work correctly. Keep this page open until you've completed the validation.

 **Note**

If the installer doesn't show a check mark for Retail Server or Async Client, wait 10 minutes so that any cached values can be updated in the cloud. Then check again. If the installer still isn't fully successful, run a full synchronization on the new channel database that this installation uses.

If you followed all the steps correctly, your configuration should have these characteristics:

- In Azure, two web applications have been automatically generated through the installer:
 - Retail Store Scale Unit Cloud POS
 - Retail Store Scale Unit Retail Server for Cloud POS
- In Azure, a web application (that is, an App registration in the new Azure portal) has been manually created for each Commerce Scale Unit installation (for example, CommerceScaleUnitHouston). A key (secret) has been created that can be used in the installer, as described earlier.
- The application ID (client ID) of the manually created web application has been added to the **Azure Active Directory applications** page in Commerce, as explained in step 1 of the preceding procedure.
- The Cloud POS application ID (client ID) that was shown at the end of the Commerce Scale Unit installer has been added on the **Identity providers** FastTab, as explained in the final steps of the "Run the Commerce Scale Unit installer" section.

Multiple-computer installation

Only advanced users should install Commerce Scale Unit across multiple computers. The following set of procedures explains how to install the Commerce channel database and Async Client on one computer, and Retail Server and Cloud POS on a second computer. The instructions assume that both systems are on the same domain, and that users for the services that will be installed have already been created on both systems. It's important that you do all configuration in headquarters.

Installation on the first computer

On the first computer, run the Commerce Scale Unit self-service installer as described earlier in this article, but make the following changes.

1. Select only Commerce channel database and Async Client as the components to install. Then select **Next** to continue with the installation.

 **Note**

You can use a generated service account for Async Client, because Async Client won't be accessed outside the computer that it's installed on.

2. Enter the client ID and key (secret). Keep these details available, so that you can use them again on the second computer.
3. After a client ID and key (secret) are created for Commerce Scale Unit, the client ID must be accepted in Commerce. Go to **System administration > Setup > Azure Active Directory applications**. Enter the client ID in the **Client ID** column, enter descriptive text in the **Name** column, and enter **RetailServiceAccount** in the **User ID** column.
4. When setup is successful, start SQL Server Configuration Manager.
5. Go to **SQL Server Network Configuration > Protocols** for the SQL Server instance.
6. Right-click, and then select **Properties**.
7. In the **Flags** section, change the value of **Set Force Encryption** to **Yes**.
8. In the **Certificate** section, select the SSL certificate on the drop-down menu. This SQL Server SSL certificate is the same certificate that is used in the installer.
9. Select **OK**.
10. Go back to **Protocols** in SQL Server Configuration Manager, and enable the following protocols:
 - Named Pipes
 - TCP/IP
11. Right-click the **TCP/IP** protocol, and then select **Properties**.
12. In the **IP Address** section, scroll down the list to **IPALL**.
13. Enter **TCPPort = 1433**.
14. Select **OK**.
15. Start Microsoft Windows Firewall with Advanced Security.
16. In Windows Firewall, create an inbound rule to open TCP port 1433.

For detailed information about SQL Server and Windows Firewall, see [Configure a Windows Firewall for Database Engine Access](#).

Installation on the second computer

On the second computer, run the Commerce Scale Unit Self-service installer as described earlier in this article, but make the following changes.

1. Select only Retail Server and Cloud POS as the components to install. If you are installing only Retail Server, don't select Cloud POS. Then select **Next** to continue with the installation.
2. Enter the domain user credentials (user name and password) that have permission to access SQL Server on the first computer. Then select **Next**.

 **Note**

A generated service account can't be used, because Retail Server requires access to the SQL database on the first computer. You must use a domain account.

3. Enter the same Client ID and key (secret) that are used on the first computer.

 **Important**

It's critical that you add this Client ID to Commerce headquarters as described earlier.

4. Select **Configure Cloud POS**, and then enter Azure AD credentials that have the correct permissions to create Azure Web Apps.

For more information about Azure Web Apps, how to create them, and how to generate a new key (secret), see [Use portal to create an Azure Active Directory application and service principal that can access resources](#). Note that the sign-in URL and the App ID URI are not important.

5. When setup is successful, don't exit the installer.

 **Note**

At first, the health check ping won't be successful, because the database isn't yet set up correctly. After you've completed the remaining steps of this

procedure, you can test the health check again.

6. Start Microsoft Internet Information Services (IIS), select the **Retail Server** website, and select the **Retail Server** web application.
7. Explore the working directory.
8. Open the Web.config file, and then, in the **connectionStrings** section, add **Server name**. **Server name** is the name of the first computer where you installed components. Save the file.
9. If the certificate that is used isn't a valid, trusted certificate from a trusted authority, open CERTMGR.MSC, and follow these steps:
 - a. Import the SQL Server SSL certificate that you created earlier, and add it to **Trusted Root**.
 - b. Open a **Command Prompt** window as an administrator, type **IISRESET**, and then press Enter.
10. If Cloud POS is configured for use, a client ID is shown. You must add this client ID to the **Commerce shared parameters** page.
 - a. In Commerce, go to **Retail and commerce > Headquarters setup > Parameters > Commerce shared parameters**.
 - b. Select **Identity providers**.
 - c. On the **Identity providers** FastTab, select the provider that begins with `HTTPS://sts.windows.net/`. The values on the **Relying parties** FastTab are set, based on your selection.
 - d. On the **Relying parties** FastTab, select **Add**. Enter the client ID that is listed in the Commerce Scale Unit installer. Set the **Type** field to **Public** and the **UserType** field to **Worker**. Then, on the Action Pane, select **Save**.
 - e. Select the new relying party, and then, on the **Server resource IDs** FastTab, select **Add**. In the **Server Resource ID** column, enter `https://retailstorescaleunit.retailserver.com`.
 - f. On the Action Pane, select **Save**.
11. In Commerce, go to **Retail and commerce > Headquarters setup > Commerce scheduler > Channel database**, and follow these steps:
 - a. Select the channel database that you created at the beginning of this article.
 - b. On the Action Pane, select **Full Sync > Job 9999**. Full synchronization might require several minutes.
 - c. In the Commerce Scale Unit installer, retest to verify that all functionality is working correctly.