

Contents

[Azure Security Documentation](#)

[Architecture and design](#)

[Advanced threat detection](#)

[Azure logging and auditing](#)

[Azure network security](#)

[Azure serverless platform security](#)

[Container security in Azure](#)

[Enabling operational security](#)

[Isolation in the Azure cloud](#)

[Secure hybrid network architecture](#)

[Security technical capabilities](#)

[Develop secure apps on Azure](#)

[Data security and encryption](#)

[Database security](#)

[Best practices](#)

[Security checklist](#)

[Disk encryption](#)

[Best practices](#)

[Data encryption-at-rest](#)

[Disk encryption for IaaS VMs](#)

[About Azure Disk Encryption](#)

[Quickstarts](#)

[Encrypt VM - Azure PowerShell](#)

[Azure Disk Encryption](#)

[Disk encryption prerequisites](#)

[Disk encryption for Windows VMs](#)

[Disk encryption for Linux VMs](#)

[Appendix for disk encryption](#)

[Disk encryption FAQ](#)

Troubleshooting

[Azure Disk Encryption with Azure AD app \(previous release\)](#)

[Disk encryption with Azure AD app prerequisites](#)

[Disk encryption with Azure AD app for Windows VMs](#)

[Disk encryption with Azure AD app for Linux VMs](#)

Azure Storage security

[Storage security guide](#)

Platform and infrastructure

[Infrastructure security](#)

[Physical security](#)

[Availability](#)

[Components and boundaries](#)

[Network architecture](#)

[Production network](#)

[SQL Database](#)

[Operations](#)

[Monitoring](#)

[Integrity](#)

[Data protection](#)

Microsoft Antimalware

IaaS security

[Best practices - IaaS workloads](#)

[Azure marketplace images](#)

Identity management

[Choose Azure AD authentication](#)

[Security checklist](#)

[Best practices](#)

[Enforce MFA on subscription administrators](#)

Network security

[Best practices](#)

[DDoS Protection](#)

[Boundary security](#)

Application

PaaS

Azure App Service for PaaS

Azure Storage for PaaS

DB best practices for PaaS

IoT

IoT security best practices

IoT security

Secure your IoT deployment

Azure Service Fabric security

Best practices

Security checklist

Monitoring, auditing, and operations

Auditing and logging

Security management

Remote management security

Azure log integration

Introduction

Get started

Integrate Azure AD audit logs

Integrate Security Center alerts

Integrate logs from Key Vault

FAQ

Operational security

Best practices

Security checklist

Governance and compliance

AU PROTECTED

IaaS web application

PaaS web application

FedRAMP

Data analytics

Data warehouse

IaaS web application

PaaS web application

FFIEC

Data analytics

Data warehouse

IaaS web application

PaaS web application

GDPR

Data analytics

Data warehouse

IaaS web application

PaaS web application

HIPAA/HITRUST

Health Data and AI

NIST SP 800-171

Data analytics

Data warehouse

IaaS web application

PaaS web application

PCI DSS

Data analytics

Data warehouse

IaaS web application

PaaS web application

TIC

Trusted Internet Connection with Azure

UK NHS

Data analytics

Data warehouse

IaaS web application

PaaS web application

UK OFFICIAL

IaaS web application

PaaS web application

White papers

Azure security services

Technical overviews

Best practices

Resources

Azure security MVP program

Cybersecurity consulting

Log a security event support ticket

Pen testing

Microsoft Threat Modeling tool

Getting started

Feature overview

Threats

Mitigations

Auditing and logging

Authentication

Authorization

Communication security

Configuration management

Cryptography

Exception management

Input validation

Sensitive data

Session management

Security is integrated into every aspect of Azure. Azure offers you unique security advantages derived from global security intelligence, sophisticated customer-facing controls, and a secure hardened infrastructure. This powerful combination helps protect your applications and data, support your compliance efforts, and provide cost-effective security for organizations of all sizes.

Learn about Azure security

[I'm considering Azure for my company. What security does Azure have to offer?](#)

[How does Microsoft share security responsibilities with my organization?](#)

[How does Microsoft secure the Azure infrastructure?](#)

[Storage security overview](#)

[Network security overview](#)

[Data encryption overview](#)

[What monitoring and logging options are available in Azure?](#)

[How does Azure secure my data at rest?](#)

[How do I encrypt Azure virtual machines](#)

White papers

[Azure security response in the cloud](#)

[Azure advanced threat detection](#)

[Azure network security](#)

[Container security in Microsoft Azure](#)

Best practices

[Security best practices for Azure](#)

[Network security](#)

[Data security](#)

[Virtual machine security](#)

[Identity and access](#)

[IaaS security](#)

[Service Fabric security](#)

[Securing the Azure Admin accounts](#)

Checklists

Securing databases
Operational security
Service Fabric security

Compliance

FFIEC

HIPAA/HITRUST

PCI DSS

FEDRAMP

UK-OFFICIAL

Resources & Services

MSFT Trust Center

Azure security partners

Cybersecurity consulting

Pen testing

Azure Security Center

Azure Key Vault

Disk Encryption

Azure Information Protection

Multi-factor authentication (MFA)

Introduction to Azure Security

11/12/2018 • 30 minutes to read • [Edit Online](#)

Overview

We know that security is job one in the cloud and how important it is that you find accurate and timely information about Azure security. One of the best reasons to use Azure for your applications and services is to take advantage of its wide array of security tools and capabilities. These tools and capabilities help make it possible to create secure solutions on the secure Azure platform. Microsoft Azure provides confidentiality, integrity, and availability of customer data, while also enabling transparent accountability.

To help you better understand the collection of security controls implemented within Microsoft Azure from both the customer's and Microsoft operations' perspectives, this white paper, "Introduction to Azure Security", is written to provide a comprehensive look at the security available with Microsoft Azure.

Azure Platform

Azure is a public cloud service platform that supports a broad selection of operating systems, programming languages, frameworks, tools, databases, and devices. It can run Linux containers with Docker integration; build apps with JavaScript, Python, .NET, PHP, Java, and Node.js; build back-ends for iOS, Android, and Windows devices.

Azure public cloud services support the same technologies millions of developers and IT professionals already rely on and trust. When you build on, or migrate IT assets to, a public cloud service provider you are relying on that organization's abilities to protect your applications and data with the services and the controls they provide to manage the security of your cloud-based assets.

Azure's infrastructure is designed from facility to applications for hosting millions of customers simultaneously, and it provides a trustworthy foundation upon which businesses can meet their security requirements.

In addition, Azure provides you with a wide array of configurable security options and the ability to control them so that you can customize security to meet the unique requirements of your organization's deployments. This document helps you understand how Azure security capabilities can help you fulfill these requirements.

NOTE

The primary focus of this document is on customer-facing controls that you can use to customize and increase security for your applications and services.

We do provide some overview information, but for detailed information on how Microsoft secures the Azure platform itself, see information provided in the [Microsoft Trust Center](#).

Abstract

Initially, public cloud migrations were driven by cost savings and agility to innovate. Security was considered a major concern for some time, and even a show stopper, for public cloud migration. However, public cloud security has transitioned from a major concern to one of the drivers for cloud migration. The rationale behind this is the superior ability of large public cloud service providers to protect applications and the data of cloud-based assets.

Azure's infrastructure is designed from the facility to applications for hosting millions of customers simultaneously, and it provides a trustworthy foundation upon which businesses can meet their security needs. In addition, Azure provides you with a wide array of configurable security options and the ability to control them so that you can customize security to meet the unique requirements of your deployments to meet your IT control policies and

adhere to external regulations.

This paper outlines Microsoft's approach to security within the Microsoft Azure cloud platform:

- Security features implemented by Microsoft to secure the Azure infrastructure, customer data, and applications.
- Azure services and security features available to you to manage the Security of the Services and your data within your Azure subscriptions.

Summary Azure Security Capabilities

The table following provide a brief description of the security features implemented by Microsoft to secure the Azure infrastructure, customer data, and secure applications.

Security Features Implemented to Secure the Azure Platform:

The features listed following are capabilities you can review to provide the assurance that the Azure Platform is managed in a secure manner. Links have been provided for further drill-down on how Microsoft addresses customer trust questions in four areas: Secure Platform, Privacy & Controls, Compliance, and Transparency.

SECURE PLATFORM	PRIVACY & CONTROLS	COMPLIANCE	TRANSPARENCY
Security Development Cycle, Internal audits	Manage your data all the time	Trust Center	How Microsoft secures customer data in Azure services
Mandatory Security training, background checks	Control on data location	Common Controls Hub	How Microsoft manage data location in Azure services
Penetration testing, intrusion detection, DDoS, Audits & logging	Provide data access on your terms	The Cloud Services Due Diligence Checklist	Who in Microsoft can access your data on what terms
State of the art data center, physical security, Secure Network	Responding to law enforcement	Compliance by service, location & Industry	How Microsoft secures customer data in Azure services
Security Incident response, Shared Responsibility	Stringent privacy standards		Review certification for Azure services, Transparency hub

Security Features Offered by Azure to Secure Data and Application

Depending on the cloud service model, there is variable responsibility for who is responsible for managing the security of the application or service. There are capabilities available in the Azure Platform to assist you in meeting these responsibilities through built-in features, and through partner solutions that can be deployed into an Azure subscription.

The built-in capabilities are organized in six (6) functional areas: Operations, Applications, Storage, Networking, Compute, and Identity. Additional detail on the features and capabilities available in the Azure Platform in these six (6) areas are provided through summary information.

Operations

This section provides additional information regarding key features in security operations and summary information about these capabilities.

Security and Audit Dashboard

The [Security and Audit solution](#) provides a comprehensive view into your organization's IT security posture with [built-in search queries](#) for notable issues that require your attention. The [Security and Audit](#) dashboard is the

home screen for everything related to security in Log Analytics. It provides high-level insight into the Security state of your computers. It also includes the ability to view all events from the past 24 hours, 7 days, or any other custom time frame.

In addition, you can configure Security & Compliance to [automatically carry out specific actions](#) when a specific event is detected.

Azure Resource Manager

[Azure Resource Manager](#) enables you to work with the resources in your solution as a group. You can deploy, update, or delete all the resources for your solution in a single, coordinated operation. You use an [Azure Resource Manager template](#) for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help you manage your resources after deployment.

Azure Resource Manager template-based deployments help improve the security of solutions deployed in Azure because standard security control settings and can be integrated into standardized template-based deployments. This reduces the risk of security configuration errors that might take place during manual deployments.

Application Insights

[Application Insights](#) is an extensible Application Performance Management (APM) service for web developers. With Application Insights, you can monitor your live web applications and automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your apps. It monitors your application all the time it's running, both during testing and after you've published or deployed it.

Application Insights creates charts and tables that show you, for example, what times of day you get most users, how responsive the app is, and how well it is served by any external services that it depends on.

If there are crashes, failures or performance issues, you can search through the telemetry data in detail to diagnose the cause. And the service sends you emails if there are any changes in the availability and performance of your app. Application Insight thus becomes a valuable security tool because it helps with the availability in the confidentiality, integrity, and availability security triad.

Azure Monitor

[Azure Monitor](#) offers visualization, query, routing, alerting, auto scale, and automation on data both from the Azure infrastructure ([Activity Log](#)) and each individual Azure resource ([Diagnostic Logs](#)). You can use Azure Monitor to alert you on security-related events that are generated in Azure logs.

Log Analytics

[Log Analytics](#) – Provides an IT management solution for both on-premises and third-party cloud-based infrastructure (such as AWS) in addition to Azure resources. Data from Azure Monitor can be routed directly to Log Analytics so you can see metrics and logs for your entire environment in one place.

Log Analytics can be a useful tool in forensic and other security analysis, as the tool enables you to quickly search through large amounts of security-related entries with a flexible query approach. In addition, on-premises [firewall and proxy logs can be exported into Azure and made available for analysis using Log Analytics](#).

Azure Advisor

[Azure Advisor](#) is a personalized cloud consultant that helps you to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry. It then recommends solutions to help improve the [performance](#), [security](#), and [high availability](#) of your resources while looking for opportunities to [reduce your overall Azure spend](#). Azure Advisor provides security recommendations, which can significantly improve your overall security posture for solutions you deploy in Azure. These recommendations are drawn from security analysis performed by [Azure Security Center](#).

Azure Security Center

Azure Security Center helps you prevent, detect, and respond to threats with increased visibility into and control over the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

In addition, Azure Security Center helps with security operations by providing you a single dashboard that surfaces alerts and recommendations that can be acted upon immediately. Often, you can remediate issues with a single click within the Azure Security Center console.

Applications

The section provides additional information regarding key features in application security and summary information about these capabilities.

Web Application vulnerability scanning

One of the easiest ways to get started with testing for vulnerabilities on your [App Service app](#) is to use the [integration with Tinfoil Security](#) to perform one-click vulnerability scanning on your app. You can view the test results in an easy-to-understand report, and learn how to fix each vulnerability with step-by-step instructions.

Penetration Testing

If you prefer to perform your own penetration tests or want to use another scanner suite or provider, you must follow the [Azure penetration testing approval process](#) and obtain prior approval to perform the desired penetration tests.

Web Application firewall

The web application firewall (WAF) in [Azure Application Gateway](#) helps protect web applications from common web-based attacks like SQL injection, cross-site scripting attacks, and session hijacking. It comes preconfigured with protection from threats identified by the [Open Web Application Security Project \(OWASP\) as the top 10 common vulnerabilities](#).

Authentication and authorization in Azure App Service

[App Service Authentication / Authorization](#) is a feature that provides a way for your application to sign in users so that you don't have to change code on the app backend. It provides an easy way to protect your application and work with per-user data.

Layered Security Architecture

Since [App Service Environments](#) provide an isolated runtime environment deployed into an [Azure Virtual Network](#), developers can create a layered security architecture providing differing levels of network access for each application tier. A common desire is to hide API back-ends from general Internet access, and only allow APIs to be called by upstream web apps. [Network Security groups \(NSGs\)](#) can be used on Azure Virtual Network subnets containing App Service Environments to restrict public access to API applications.

Web server diagnostics and application diagnostics

App Service web apps provide diagnostic functionality for logging information from both the web server and the web application. These are logically separated into [web server diagnostics](#) and [application diagnostics](#). Web server includes two major advances in diagnosing and troubleshooting sites and applications.

The first new feature is real-time state information about application pools, worker processes, sites, application domains, and running requests. The second new advantages are the detailed trace events that track a request throughout the complete request-and-response process.

To enable the collection of these trace events, IIS 7 can be configured to automatically capture full trace logs, in XML format, for any particular request based on elapsed time or error response codes.

Web server diagnostics

You can enable or disable the following kinds of logs:

- Detailed Error Logging - Detailed error information for HTTP status codes that indicate a failure (status code 400 or greater). This may contain information that can help determine why the server returned the error code.
- Failed Request Tracing - Detailed information on failed requests, including a trace of the IIS components used to process the request and the time taken in each component. This can be useful if you are attempting to increase site performance or isolate what is causing a specific HTTP error to be returned.
- Web Server Logging - Information about HTTP transactions using the W3C extended log file format. This is useful when determining overall site metrics such as the number of requests handled or how many requests are from a specific IP address.

Application diagnostics

[Application diagnostics](#) allows you to capture information produced by a web application. ASP.NET applications can use the [System.Diagnostics.Trace](#) class to log information to the application diagnostics log. In Application Diagnostics, there are two major types of events, those related to application performance and those related to application failures and errors. The failures and errors can be divided further into connectivity, security, and failure issues. Failure issues are typically related to a problem with the application code.

In Application Diagnostics, you can view events grouped in these ways:

- All (displays all events)
- Application Errors (displays exception events)
- Performance (displays performance events)

Storage

The section provides additional information regarding key features in Azure storage security and summary information about these capabilities.

Role-Based Access Control (RBAC)

You can secure your storage account with Role-Based Access Control (RBAC). Restricting access based on the [need to know](#) and [least privilege](#) security principles is imperative for organizations that want to enforce Security policies for data access. These access rights are granted by assigning the appropriate RBAC role to groups and applications at a certain scope. You can use [built-in RBAC roles](#), such as Storage Account Contributor, to assign privileges to users. Access to the storage keys for a storage account using the [Azure Resource Manager](#) model can be controlled through Role-Based Access Control (RBAC).

Shared Access Signature

A [shared access signature \(SAS\)](#) provides delegated access to resources in your storage account. The SAS means that you can grant a client limited permissions to objects in your storage account for a specified period and with a specified set of permissions. You can grant these limited permissions without having to share your account access keys.

Encryption in Transit

Encryption in transit is a mechanism of protecting data when it is transmitted across networks. With Azure Storage, you can secure data using:

- [Transport-level encryption](#), such as HTTPS when you transfer data into or out of Azure Storage.
- [Wire encryption](#), such as [SMB 3.0 encryption](#) for [Azure File shares](#).
- Client-side encryption, to encrypt the data before it is transferred into storage and to decrypt the data after it is transferred out of storage.

Encryption at rest

For many organizations, data encryption at rest is a mandatory step towards data privacy, compliance, and data sovereignty. There are three Azure storage security features that provide encryption of data that is "at rest":

- [Storage Service Encryption](#) allows you to request that the storage service automatically encrypt data when writing it to Azure Storage.
- [Client-side Encryption](#) also provides the feature of encryption at rest.
- [Azure Disk Encryption](#) allows you to encrypt the OS disks and data disks used by an IaaS virtual machine.

Storage Analytics

[Azure Storage Analytics](#) performs logging and provides metrics data for a storage account. You can use this data to trace requests, analyze usage trends, and diagnose issues with your storage account. Storage Analytics logs detailed information about successful and failed requests to a storage service. This information can be used to monitor individual requests and to diagnose issues with a storage service. Requests are logged on a best-effort basis. The following types of authenticated requests are logged:

- Successful requests.
- Failed requests, including timeout, throttling, network, authorization, and other errors.
- Requests using a Shared Access Signature (SAS), including failed and successful requests.
- Requests to analytics data.

Enabling Browser-Based Clients Using CORS

[Cross-Origin Resource Sharing \(CORS\)](#) is a mechanism that allows domains to give each other permission for accessing each other's resources. The User Agent sends extra headers to ensure that the JavaScript code loaded from a certain domain is allowed to access resources located at another domain. The latter domain then replies with extra headers allowing or denying the original domain access to its resources.

Azure storage services now support CORS so that once you set the CORS rules for the service, a properly authenticated request made against the service from a different domain is evaluated to determine whether it is allowed according to the rules you have specified.

Networking

The section provides additional information regarding key features in Azure network security and summary information about these capabilities.

Network Layer Controls

Network access control is the act of limiting connectivity to and from specific devices or subnets and represents the core of network security. The goal of network access control is to make sure that your virtual machines and services are accessible to only users and devices to which you want them accessible.

Network Security Groups

A [Network Security Group \(NSG\)](#) is a basic stateful packet filtering firewall and it enables you to control access based on a [5-tuple](#). NSGs do not provide application layer inspection or authenticated access controls. They can be used to control traffic moving between subnets within an Azure Virtual Network and traffic between an Azure Virtual Network and the Internet.

Route Control and Forced Tunneling

The ability to control routing behavior on your Azure Virtual Networks is a critical network security and access control capability. For example, if you want to make sure that all traffic to and from your Azure Virtual Network goes through that virtual security appliance, you need to be able to control and customize routing behavior. You can do this by configuring User-Defined Routes in Azure.

[User-Defined Routes](#) allow you to customize inbound and outbound paths for traffic moving into and out of individual virtual machines or subnets to insure the most secure route possible. [Forced tunneling](#) is a mechanism you can use to ensure that your services are not allowed to initiate a connection to devices on the Internet.

This is different from being able to accept incoming connections and then responding to them. Front-end web servers need to respond to requests from Internet hosts, and so Internet-sourced traffic is allowed inbound to these web servers and the web servers can respond.

Forced tunneling is commonly used to force outbound traffic to the Internet to go through on-premises security proxies and firewalls.

Virtual Network Security Appliances

While Network Security Groups, User-Defined Routes, and forced tunneling provide you a level of security at the network and transport layers of the [OSI model](#), there may be times when you want to enable security at higher levels of the stack. You can access these enhanced network security features by using an Azure partner network security appliance solution. You can find the most current Azure partner network security solutions by visiting the [Azure Marketplace](#) and searching for "security" and "network security."

Azure Virtual Network

An Azure virtual network (VNet) is a representation of your own network in the cloud. It is a logical isolation of the Azure network fabric dedicated to your subscription. You can fully control the IP address blocks, DNS settings, security policies, and route tables within this network. You can segment your VNet into subnets and place Azure IaaS virtual machines (VMs) and/or [Cloud services \(PaaS role instances\)](#) on Azure Virtual Networks.

Additionally, you can connect the virtual network to your on-premises network using one of the [connectivity options](#) available in Azure. In essence, you can expand your network to Azure, with complete control on IP address blocks with the benefit of enterprise scale Azure provides.

Azure networking supports various secure remote access scenarios. Some of these include:

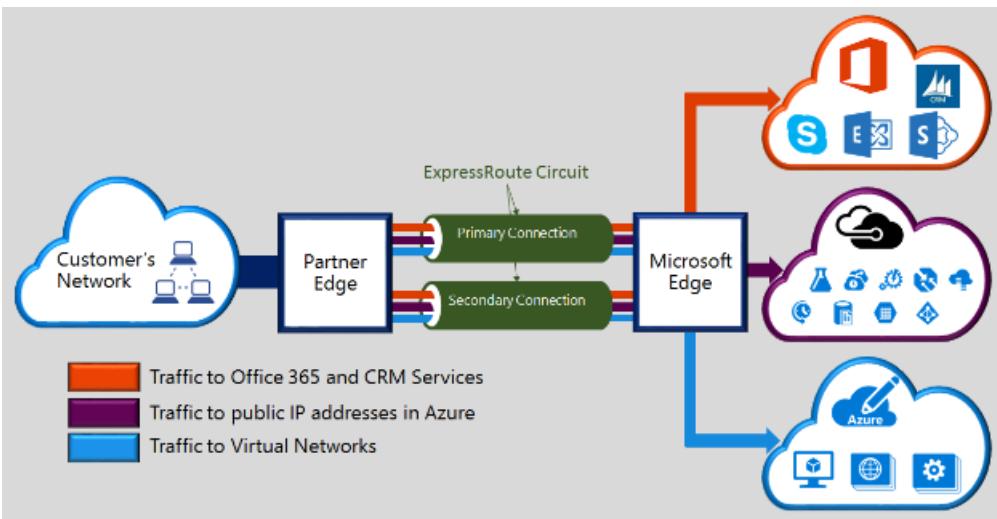
- [Connect individual workstations to an Azure Virtual Network](#)
- [Connect on-premises network to an Azure Virtual Network with a VPN](#)
- [Connect on-premises network to an Azure Virtual Network with a dedicated WAN link](#)
- [Connect Azure Virtual Networks to each other](#)

VPN Gateway

To send network traffic between your Azure Virtual Network and your on-premises site, you must create a VPN gateway for your Azure Virtual Network. A [VPN gateway](#) is a type of virtual network gateway that sends encrypted traffic across a public connection. You can also use VPN gateways to send traffic between Azure Virtual Networks over the Azure network fabric.

Express Route

Microsoft Azure [ExpressRoute](#) is a dedicated WAN link that lets you extend your on-premises networks into the Microsoft cloud over a dedicated private connection facilitated by a connectivity provider.

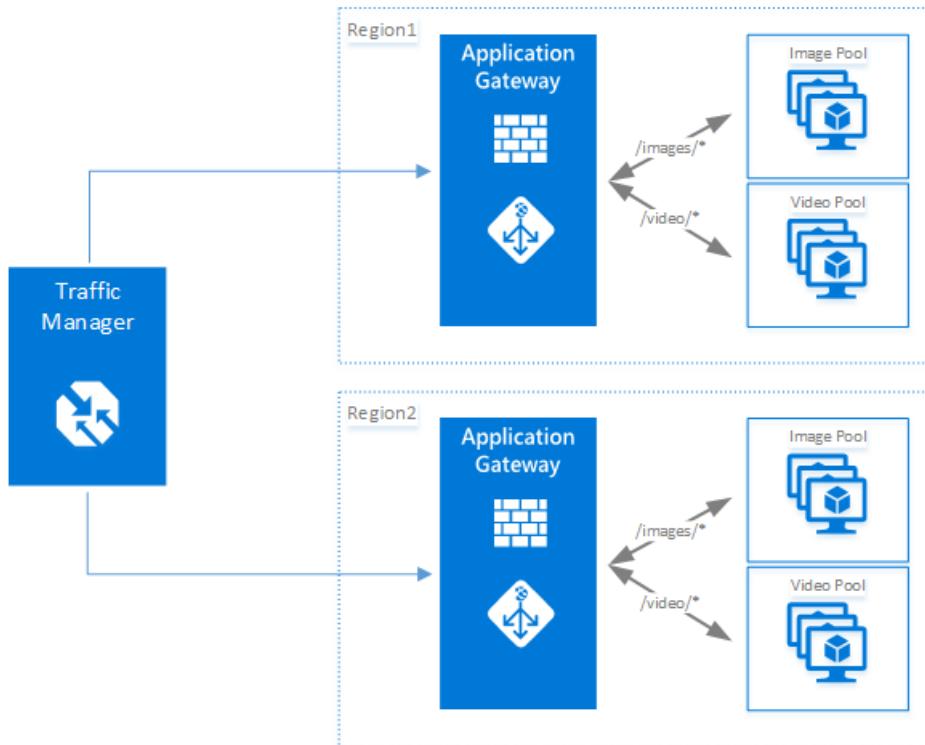


With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, Office 365, and CRM Online. Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a co-location facility.

ExpressRoute connections do not go over the public Internet and thus can be considered more secure than VPN-based solutions. This allows ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.

Application Gateway

Microsoft [Azure Application Gateway](#) provides an [Application Delivery Controller \(ADC\)](#) as a service, offering various layer 7 load balancing capabilities for your application.



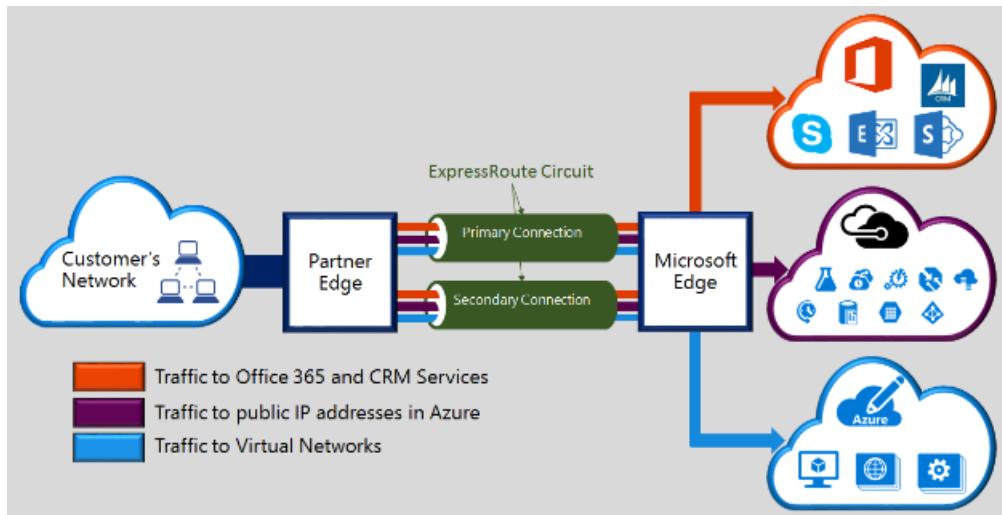
It allows you to optimize web farm productivity by offloading CPU intensive SSL termination to the Application Gateway (also known as "SSL offload" or "SSL bridging"). It also provides other Layer 7 routing capabilities including round-robin distribution of incoming traffic, cookie-based session affinity, URL path-based routing, and the ability to host multiple websites behind a single Application Gateway. Azure Application Gateway is a layer-7 load balancer.

It provides failover, performance-routing HTTP requests between different servers, whether they are on the cloud or on-premises.

Application provides many Application Delivery Controller (ADC) features including HTTP load balancing, cookie-based session affinity, [Secure Sockets Layer \(SSL\)](#) offload, custom health probes, support for multi-site, and many others.

Web Application Firewall

Web Application Firewall is a feature of [Azure Application Gateway](#) that provides protection to web applications that use application gateway for standard Application Delivery Control (ADC) functions. Web application firewall does this by protecting them against most of the OWASP top 10 common web vulnerabilities.



- SQL injection protection
- Common Web Attacks Protection such as command injection, HTTP request smuggling, HTTP response splitting, and remote file inclusion attack
- Protection against HTTP protocol violations
- Protection against HTTP protocol anomalies such as missing host user-agent and accept headers
- Prevention against bots, crawlers, and scanners
- Detection of common application misconfigurations (that is, Apache, IIS, etc.)

A centralized web application firewall to protect against web attacks makes security management much simpler and gives better assurance to the application against the threats of intrusions. A WAF solution can also react to a security threat faster by patching a known vulnerability at a central location versus securing each of individual web applications. Existing application gateways can be converted to an application gateway with web application firewall easily.

Traffic Manager

Microsoft [Azure Traffic Manager](#) allows you to control the distribution of user traffic for service endpoints in different data centers. Service endpoints supported by Traffic Manager include Azure VMs, Web Apps, and Cloud services. You can also use Traffic Manager with external, non-Azure endpoints. Traffic Manager uses the Domain Name System (DNS) to direct client requests to the most appropriate endpoint based on a [traffic-routing method](#) and the health of the endpoints.

Traffic Manager provides a range of traffic-routing methods to suit different application needs, endpoint health [monitoring](#), and automatic failover. Traffic Manager is resilient to failure, including the failure of an entire Azure region.

Azure Load Balancer

[Azure Load Balancer](#) delivers high availability and network performance to your applications. It is a Layer 4 (TCP, UDP) load balancer that distributes incoming traffic among healthy instances of services defined in a load-balanced set. Azure Load Balancer can be configured to:

- Load balance incoming Internet traffic to virtual machines. This configuration is known as [Internet-facing load balancing](#).
- Load balance traffic between virtual machines in a virtual network, between virtual machines in cloud services, or between on-premises computers and virtual machines in a cross-premises virtual network. This configuration is known as [internal load balancing](#).
- Forward external traffic to a specific virtual machine

Internal DNS

You can manage the list of DNS servers used in a VNet in the Management Portal, or in the network configuration file. Customer can add up to 12 DNS servers for each VNet. When specifying DNS servers, it's important to verify that you list customer's DNS servers in the correct order for customer's environment. DNS server lists do not work round-robin. They are used in the order that they are specified. If the first DNS server on the list is able to be reached, the client uses that DNS server regardless of whether the DNS server is functioning properly or not. To change the DNS server order for customer's virtual network, remove the DNS servers from the list and add them back in the order that customer wants. DNS supports the availability aspect of the "CIA" security triad.

Azure DNS

The [Domain Name System](#), or DNS, is responsible for translating (or resolving) a website or service name to its IP address. [Azure DNS](#) is a hosting service for DNS domains, providing name resolution using Microsoft Azure infrastructure. By hosting your domains in Azure, you can manage your DNS records using the same credentials, APIs, tools, and billing as your other Azure services. DNS supports the availability aspect of the "CIA" security triad.

Log Analytics NSGs

You can enable the following diagnostic log categories for NSGs:

- Event: Contains entries for which NSG rules are applied to VMs and instance roles based on MAC address. The status for these rules is collected every 60 seconds.
- Rules counter: Contains entries for how many times each NSG rule is applied to deny or allow traffic.

Azure Security Center

Security Center helps you prevent, detect, and respond to threats, and provides you increased visibility into, and control over, the Security of your Azure resources. It provides integrated Security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of Security solutions. Network recommendations center around firewalls, Network Security Groups, configuring inbound traffic rules, and more.

Available network recommendations are as follows:

- [Add a Next Generation Firewall](#) Recommends that you add a Next Generation Firewall (NGFW) from a Microsoft partner to increase your security protections
- [Route traffic through NGFW only](#) Recommends that you configure network security group (NSG) rules that force inbound traffic to your VM through your NGFW.
- [Enable Network Security Groups on subnets or virtual machines](#) Recommends that you enable NSGs on subnets or VMs.
- [Restrict access through Internet facing endpoint](#) Recommends that you configure inbound traffic rules for NSGs.

Compute

The section provides additional information regarding key features in this area and summary information about these capabilities.

Antimalware & Antivirus

With Azure IaaS, you can use antimalware software from security vendors such as Microsoft, Symantec, Trend Micro, McAfee, and Kaspersky to protect your virtual machines from malicious files, adware, and other threats.

[Microsoft Antimalware](#) for Azure Cloud Services and Virtual Machines is a protection capability that helps identify and remove viruses, spyware, and other malicious software. Microsoft Antimalware provides configurable alerts when known malicious or unwanted software attempts to install itself or run on your Azure systems. Microsoft Antimalware can also be deployed using Azure Security Center

Hardware Security Module

Encryption and authentication do not improve security unless the keys themselves are protected. You can simplify the management and security of your critical secrets and keys by storing them in [Azure Key Vault](#). Key Vault provides the option to store your keys in hardware Security modules (HSMs) certified to FIPS 140-2 Level 2 standards. Your SQL Server encryption keys for backup or [transparent data encryption](#) can all be stored in Key Vault with any keys or secrets from your applications. Permissions and access to these protected items are managed through [Azure Active Directory](#).

Virtual machine backup

[Azure Backup](#) is a solution that protects your application data with zero capital investment and minimal operating costs. Application errors can corrupt your data, and human errors can introduce bugs into your applications that can lead to security issues. With Azure Backup, your virtual machines running Windows and Linux are protected.

Azure Site Recovery

An important part of your organization's [business continuity/disaster recovery \(BCDR\)](#) strategy is figuring out how to keep corporate workloads and apps up and running when planned and unplanned outages occur. [Azure Site Recovery](#) helps orchestrate replication, failover, and recovery of workloads and apps so that they are available from a secondary location if your primary location goes down.

SQL VM TDE

[Transparent data encryption \(TDE\)](#) and column level encryption (CLE) are SQL server encryption features. This form of encryption requires customers to manage and store the cryptographic keys you use for encryption.

The Azure Key Vault (AKV) service is designed to improve the security and management of these keys in a secure and highly available location. The SQL Server Connector enables SQL Server to use these keys from Azure Key Vault.

If you are running SQL Server with on-premises machines, there are steps you can follow to access Azure Key Vault from your on-premises SQL Server machine. But for SQL Server in Azure VMs, you can save time by using the Azure Key Vault Integration feature. With a few Azure PowerShell cmdlets to enable this feature, you can automate the configuration necessary for a SQL VM to access your key vault.

VM Disk Encryption

[Azure Disk Encryption](#) is a new capability that helps you encrypt your Windows and Linux IaaS virtual machine disks. It applies the industry standard BitLocker feature of Windows and the DM-Crypt feature of Linux to provide volume encryption for the OS and the data disks. The solution is integrated with Azure Key Vault to help you control and manage the disk-encryption keys and secrets in your Key Vault subscription. The solution also ensures that all data on the virtual machine disks are encrypted at rest in your Azure storage.

Virtual networking

Virtual machines need network connectivity. To support that requirement, Azure requires virtual machines to be connected to an Azure Virtual Network. An Azure Virtual Network is a logical construct built on top of the physical Azure network fabric. Each logical [Azure Virtual Network](#) is isolated from all other Azure Virtual Networks. This

isolation helps insure that network traffic in your deployments is not accessible to other Microsoft Azure customers.

Patch Updates

Patch Updates provide the basis for finding and fixing potential problems and simplify the software update management process, both by reducing the number of software updates you must deploy in your enterprise and by increasing your ability to monitor compliance.

Security policy management and reporting

[Azure Security Center](#) helps you prevent, detect, and respond to threats, and provides you increased visibility into, and control over, the security of your Azure resources. It provides integrated Security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Azure Security Center

Security Center helps you prevent, detect, and respond to threats with increased visibility into and control over the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Identity and access management

Securing systems, applications, and data begins with identity-based access controls. The identity and access management features that are built into Microsoft business products and services help protect your organizational and personal information from unauthorized access while making it available to legitimate users whenever and wherever they need it.

Secure Identity

Microsoft uses multiple security practices and technologies across its products and services to manage identity and access.

- [Multi-Factor Authentication](#) requires users to use multiple methods for access, on-premises and in the cloud. It provides strong authentication with a range of easy verification options, while accommodating users with a simple sign-in process.
- [Microsoft Authenticator](#) provides a user-friendly Multi-Factor Authentication experience that works with both Microsoft Azure Active Directory and Microsoft accounts, and includes support for wearables and fingerprint-based approvals.
- [Password policy enforcement](#) increases the security of traditional passwords by imposing length and complexity requirements, forced periodic rotation, and account lockout after failed authentication attempts.
- [Token-based authentication](#) enables authentication via Azure Active Directory.
- [Role-based access control \(RBAC\)](#) enables you to grant access based on the user's assigned role, making it easy to give users only the amount of access they need to perform their job duties. You can customize RBAC per your organization's business model and risk tolerance.
- [Integrated identity management \(hybrid identity\)](#) enables you to maintain control of users' access across internal datacenters and cloud platforms, creating a single user identity for authentication and authorization to all resources.

Secure Apps and data

[Azure Active Directory](#), a comprehensive identity and access management cloud solution, helps secure access to data in applications on site and in the cloud, and simplifies the management of users and groups. It combines core directory services, advanced identity governance, security, and application access management, and makes it easy

for developers to build policy-based identity management into their apps. To enhance your Azure Active Directory, you can add paid capabilities using the Azure Active Directory Basic, Premium P1, and Premium P2 editions.

FREE / COMMON FEATURES	BASIC FEATURES	PREMIUM P1 FEATURES	PREMIUM P2 FEATURES	AZURE ACTIVE DIRECTORY JOIN – WINDOWS 10 ONLY RELATED FEATURES
Directory Objects, User/Group Management (add/update/delete)/ User-based provisioning, Device registration, Single Sign-On (SSO), Self-Service Password Change for cloud users, Connect (Sync engine that extends on-premises directories to Azure Active Directory), Security / Usage Reports	Group-based access management / provisioning, Self-Service Password Reset for cloud users, Company Branding (Logon Pages/Access Panel customization), Application Proxy, SLA 99.9%	Self-Service Group and app Management/Self-Service application additions/Dynamic Groups, Self-Service Password Reset/Change/Unlock with on-premises write-back, Multi-Factor Authentication (Cloud and On-premises (MFA Server)), MIM CAL + MIM Server, Cloud App Discovery, Connect Health, Automatic password rollover for group accounts	Identity Protection, Privileged Identity Management	Join a device to Azure AD, Desktop SSO, Microsoft Passport for Azure AD, Administrator Bitlocker recovery, MDM auto-enrollment, Self-Service Bitlocker recovery, Additional local administrators to Windows 10 devices via Azure AD Join

- [Cloud App Discovery](#) is a premium feature of Azure Active Directory that enables you to identify cloud applications that are used by the employees in your organization.
- [Azure Active Directory Identity Protection](#) is a security service that uses Azure Active Directory anomaly detection capabilities to provide a consolidated view into risk events and potential vulnerabilities that could affect your organization's identities.
- [Azure Active Directory Domain Services](#) enables you to join Azure VMs to a domain without the need to deploy domain controllers. Users sign in to these VMs by using their corporate Active Directory credentials, and can seamlessly access resources.
- [Azure Active Directory B2C](#) is a highly available, global identity management service for consumer-facing apps that can scale to hundreds of millions of identities and integrate across mobile and web platforms. Your customers can sign in to all your apps through customizable experiences that use existing social media accounts, or you can create new standalone credentials.
- [Azure Active Directory B2B Collaboration](#) is a secure partner integration solution that supports your cross-company relationships by enabling partners to access your corporate applications and data selectively by using their self-managed identities.
- [Azure Active Directory Join](#) enables you to extend cloud capabilities to Windows 10 devices for centralized management. It makes it possible for users to connect to the corporate or organizational cloud through Azure Active Directory and simplifies access to apps and resources.
- [Azure Active Directory Application Proxy](#) provides SSO and secure remote access for web applications hosted on-premises.

Next Steps

- [Getting started with Microsoft Azure Security](#)

Azure services and features you can use to help secure your services and data within Azure

- [Azure Security Center](#)

Prevent, detect, and respond to threats with increased visibility and control over the security of your Azure resources

- [Security health monitoring in Azure Security Center](#)

The monitoring capabilities in Azure Security Center to monitor compliance with policies.

Azure advanced threat detection

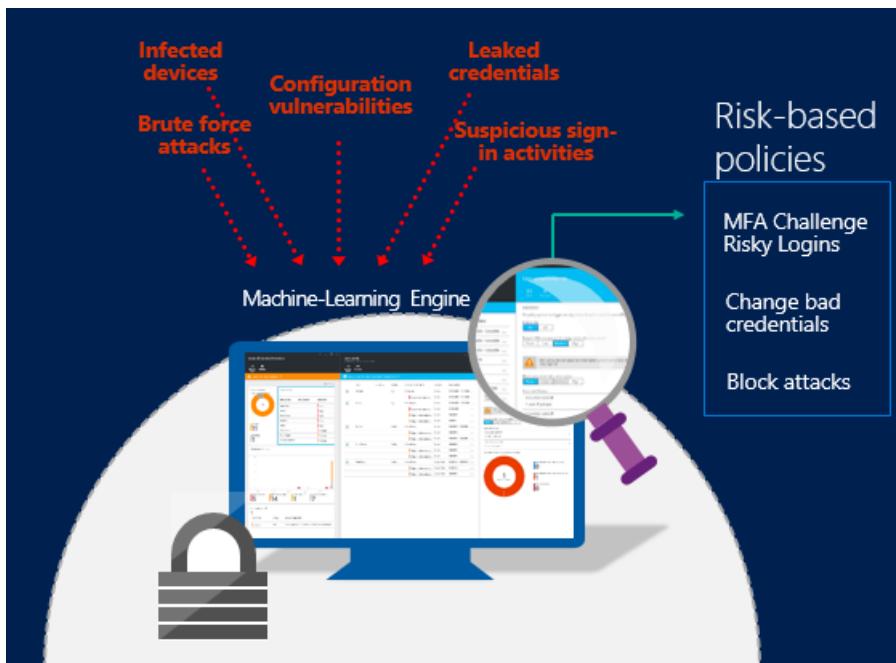
10/18/2018 • 21 minutes to read • [Edit Online](#)

Azure offers built in advanced threat detection functionality through services such as Azure Active Directory (Azure AD), Azure Log Analytics, and Azure Security Center. This collection of security services and capabilities provides a simple and fast way to understand what is happening within your Azure deployments.

Azure provides a wide array of options to configure and customize security to meet the requirements of your app deployments. This article discusses how to meet these requirements.

Azure Active Directory Identity Protection

[Azure AD Identity Protection](#) is an [Azure Active Directory Premium P2](#) edition feature that provides an overview of the risk events and potential vulnerabilities that can affect your organization's identities. Identity Protection uses existing Azure AD anomaly-detection capabilities that are available through [Azure AD Anomalous Activity Reports](#), and introduces new risk event types that can detect real time anomalies.



Identity Protection uses adaptive machine learning algorithms and heuristics to detect anomalies and risk events that might indicate that an identity has been compromised. Using this data, Identity Protection generates reports and alerts so that you can investigate these risk events and take appropriate remediation or mitigation action.

Azure Active Directory Identity Protection is more than a monitoring and reporting tool. Based on risk events, Identity Protection calculates a user risk level for each user, so that you can configure risk-based policies to automatically protect the identities of your organization.

These risk-based policies, in addition to other [conditional access controls](#) that are provided by Azure Active Directory and [EMS](#), can automatically block or offer adaptive remediation actions that include password resets and multi-factor authentication enforcement.

Identity Protection capabilities

Azure Active Directory Identity Protection is more than a monitoring and reporting tool. To protect your organization's identities, you can configure risk-based policies that automatically respond to detected issues when a specified risk level has been reached. These policies, in addition to other conditional access controls provided by

Azure Active Directory and EMS, can either automatically block or initiate adaptive remediation actions including password resets and multi-factor authentication enforcement.

Examples of some of the ways that Azure Identity Protection can help secure your accounts and identities include:

Detecting risk events and risky accounts

- Detect six risk event types using machine learning and heuristic rules.
- Calculate user risk levels.
- Provide custom recommendations to improve overall security posture by highlighting vulnerabilities.

Investigating risk events

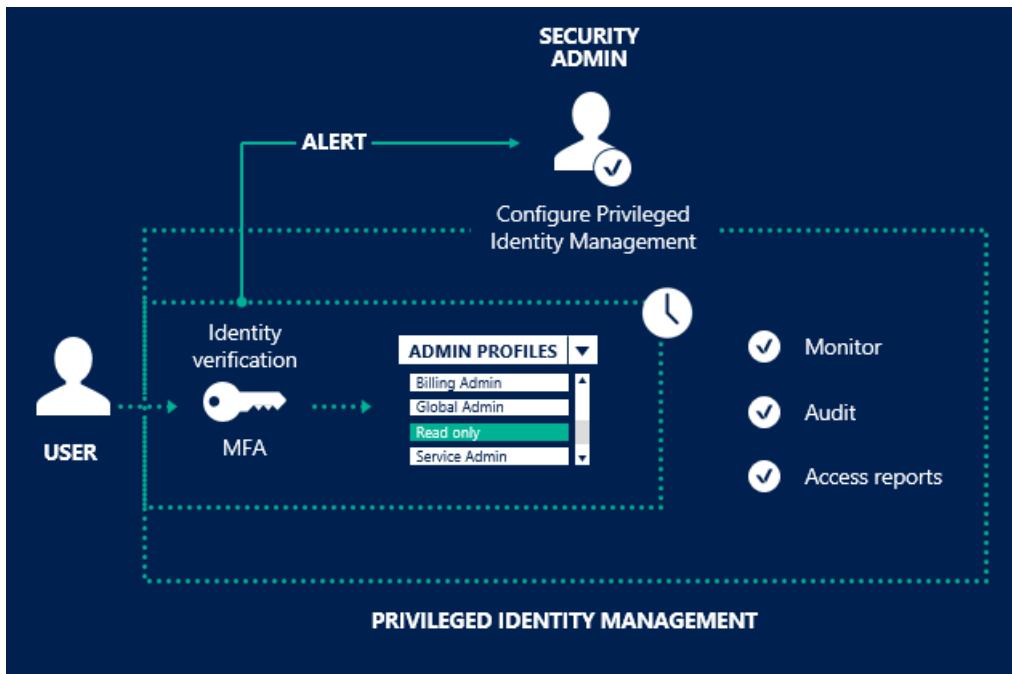
- Send notifications for risk events.
- Investigate risk events using relevant and contextual information.
- Provide basic workflows to track investigations.
- Provide easy access to remediation actions such as password reset.

Risk-based, conditional-access policies

- Mitigate risky sign-ins by blocking sign-ins or requiring multi-factor authentication challenges.
- Block or secure risky user accounts.
- Require users to register for multi-factor authentication.

Azure AD Privileged Identity Management

With [Azure Active Directory Privileged Identity Management \(PIM\)](#), you can manage, control, and monitor access within your organization. This feature includes access to resources in Azure AD and other Microsoft online services, such as Office 365 or Microsoft Intune.



PIM helps you:

- Get alerts and reports about Azure AD administrators and just-in-time (JIT) administrative access to Microsoft online services, such as Office 365 and Intune.
- Get reports about administrator access history and changes in administrator assignments.
- Get alerts about access to a privileged role.

Azure Log Analytics

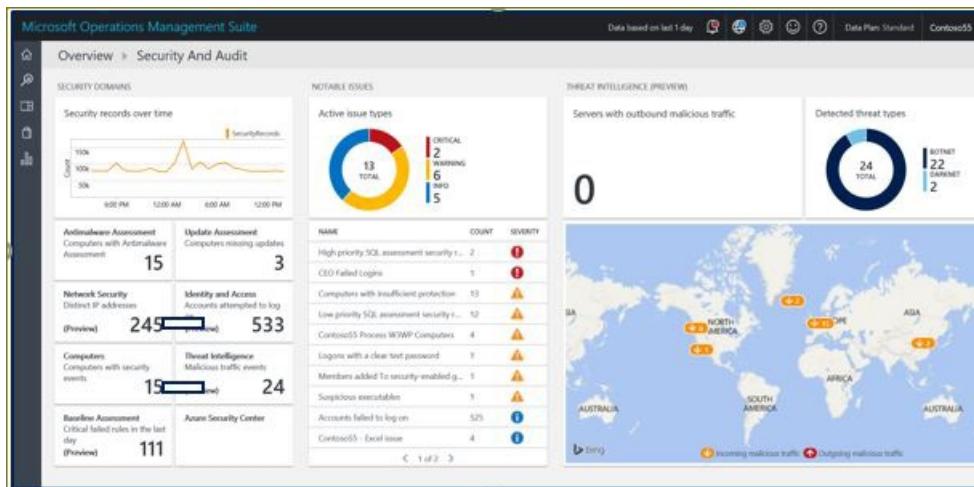
[Log Analytics](#) is a Microsoft cloud-based IT management solution that helps you manage and protect your on-premises and cloud infrastructure. Because Log Analytics is implemented as a cloud-based service, you can have it up and running quickly with minimal investment in infrastructure services. New security features are delivered automatically, saving ongoing maintenance and upgrade costs.

In addition to providing valuable services on its own, Log Analytics can integrate with System Center components, such as [System Center Operations Manager](#), to extend your existing security management investments into the cloud. System Center and Log Analytics can work together to provide a full hybrid management experience.

Holistic security and compliance posture

The [Log Analytics Security and Audit dashboard](#) provides a comprehensive view into your organization's IT security posture, with built-in search queries for notable issues that require your attention. The Security and Audit dashboard is the home screen for everything related to security in Log Analytics. It provides high-level insight into the security state of your computers. You can also view all events from the past 24 hours, 7 days, or any other custom timeframe.

Log Analytics help you quickly and easily understand the overall security posture of any environment, all within the context of IT Operations, including software update assessment, antimalware assessment, and configuration baselines. Security log data is readily accessible to streamline the security and compliance audit processes.

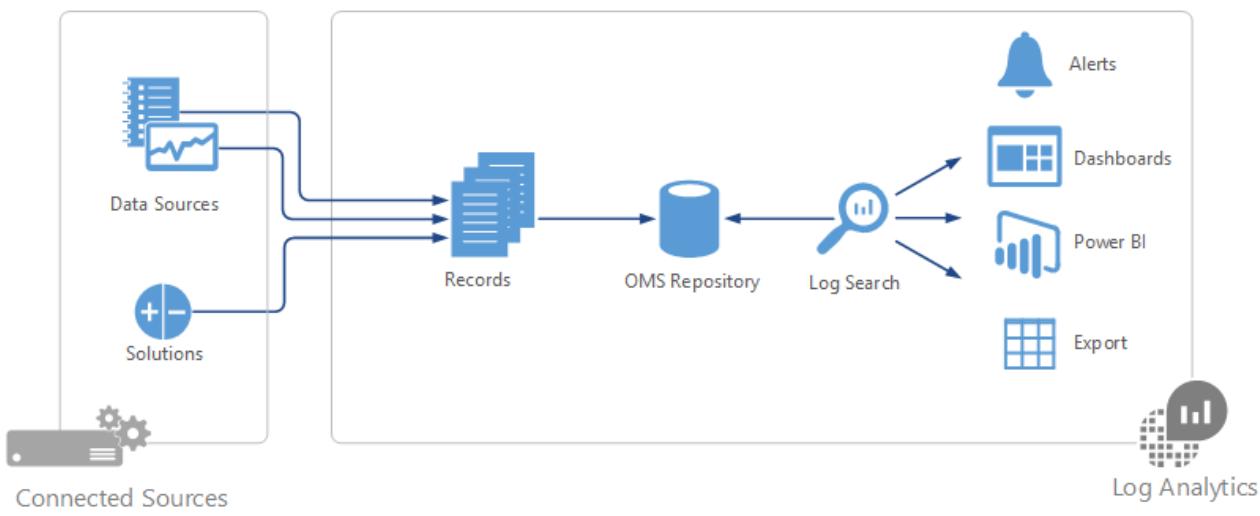


The Log Analytics Security and Audit dashboard is organized into four major categories:

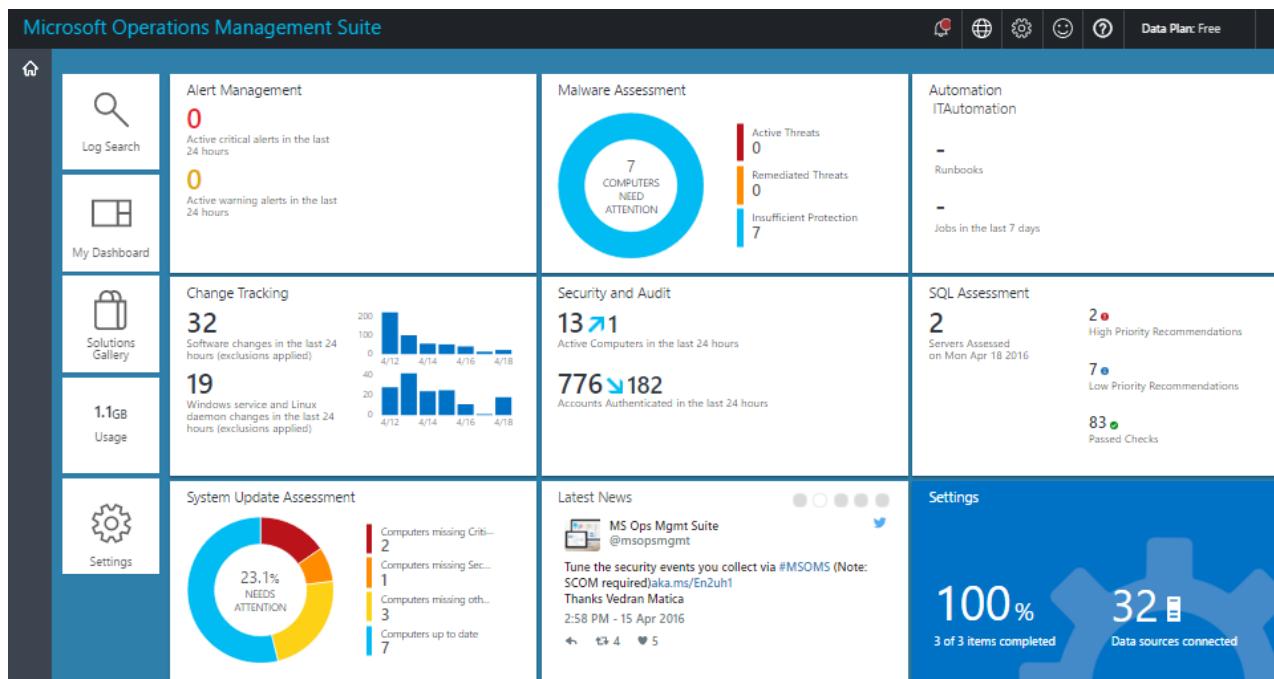
- **Security Domains:** Lets you further explore security records over time; access malware assessments; update assessments; view network security, identity, and access information; view computers with security events; and quickly access the Azure Security Center dashboard.
- **Notable Issues:** Lets you quickly identify the number of active issues and the severity of the issues.
- **Detections (Preview):** Lets you identify attack patterns by displaying security alerts as they occur against your resources.
- **Threat Intelligence:** Lets you identify attack patterns by displaying the total number of servers with outbound malicious IP traffic, the malicious threat type, and a map of the IPs locations.
- **Common security queries:** Lists the most common security queries that you can use to monitor your environment. When you select any query, the Search pane opens and displays the results for that query.

Insight and analytics

At the center of [Log Analytics](#) is the repository, which is hosted by Azure.



You collect data into the repository from connected sources by configuring data sources and adding solutions to your subscription.



Data sources and solutions each create separate record types with their own set of properties, but you can still analyze them together in queries to the repository. You can use the same tools and methods to work with a variety of data that's collected by various sources.

Most of your interaction with Log Analytics is through the Azure portal, which runs in any browser and provides you with access to configuration settings and multiple tools to analyze and act on collected data. From the portal, you can use:

- **Log searches** where you construct queries to analyze collected data.
- **Dashboards**, which you can customize with graphical views of your most valuable searches.
- **Solutions**, which provide additional functionality and analysis tools.

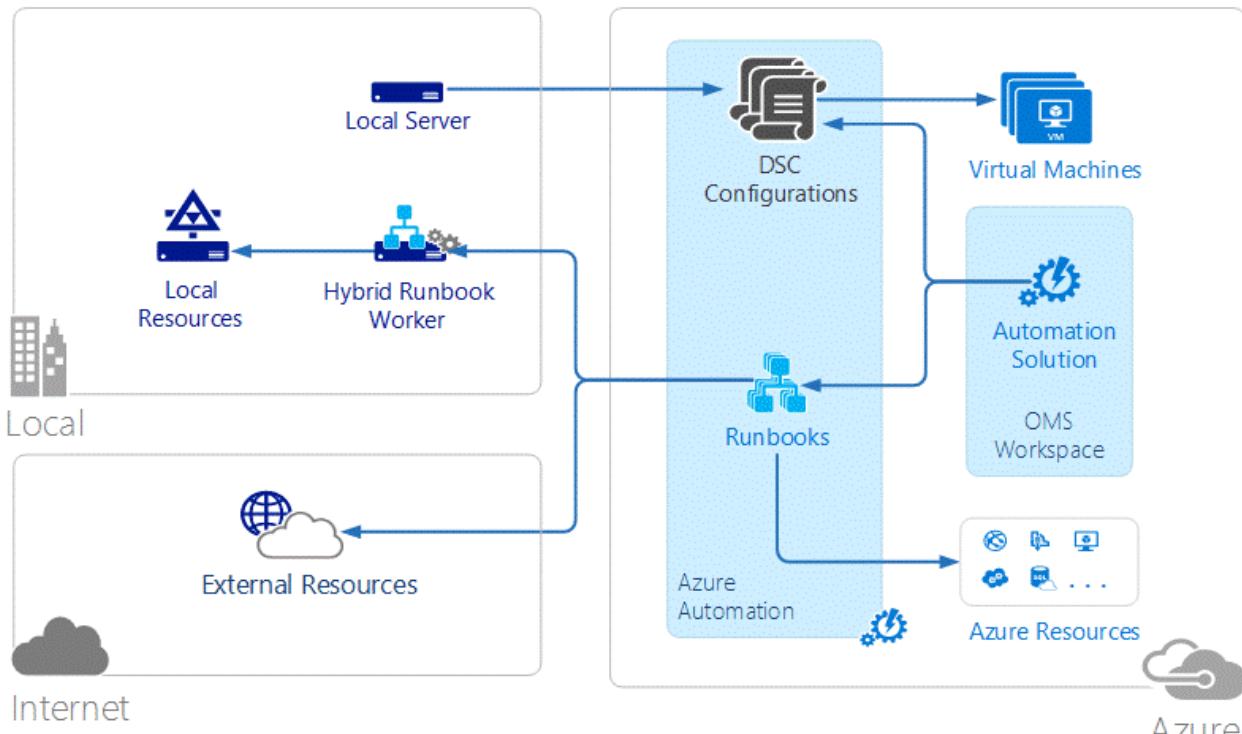
Solutions Gallery							
							
App Dependency Monitor Coming Soon Automatically discover and map servers and their dependencies in real-time.	Malware Assessment Owned View status of antivirus and antimalware scans across your servers.	Containers Coming Soon See Docker container performance metrics and logs from containers across your public or private cloud environments.	Network Performance Monitor Coming Soon Offers near real time monitoring of network performance parameters like loss and latency.	Security and Audit Owned Provides the ability to explore security related data and helps identify security breaches.	System Update Assessment Owned Identify missing system updates across your servers.	AD Replication Status Owned Identify Active Directory replication issues in your environment.	Malware Assessment Owned View status of antivirus and antimalware scans across your servers.
							
Azure Networking Analytics Coming Soon Gain insight into your Azure Network data	Security and Audit Owned Provides the ability to explore security related data and helps identify security breaches.	Wire Data Coming Soon Provides the ability to explore wire data and helps identify network related issues.	Office 365 Coming Soon Get full visibility into your Office 365 user activities, perform forensics as well as audit and compliance.	SQL Assessment Free Assess the risk and health of SQL Server environments.	AD Assessment Owned Assess the risk and health of Active Directory environments.	Alert Management Owned View your Operations Manager and OMS alerts to easily triage alerts and identify the root causes of problems in your environment.	Automation Owned Automate time consuming and frequently repeated tasks in the cloud and on-premises.

Solutions add functionality to Log Analytics. They primarily run in the cloud and provide analysis of data that's collected in the Log Analytics repository. Solutions might also define new record types to be collected that can be analyzed with log searches or by using an additional user interface that the solution provides in the Log Analytics dashboard.

The Security and Audit dashboard is an example of these types of solutions.

Automation and control: Alert on security configuration drifts

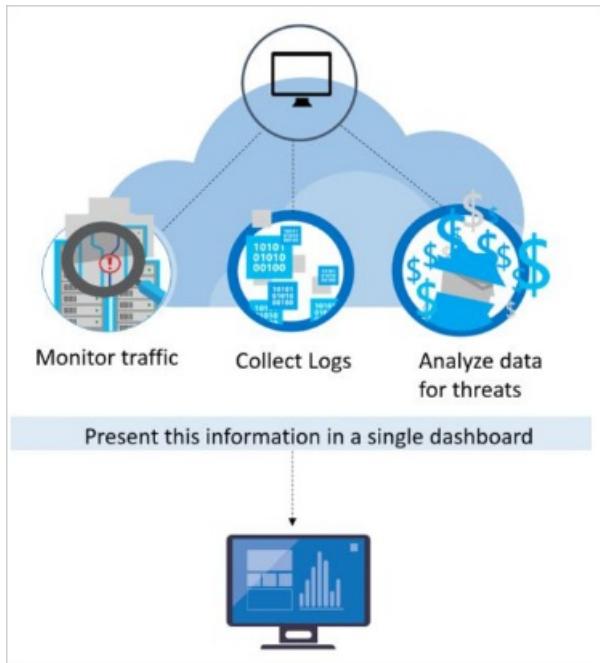
Azure Automation automates administrative processes with runbooks that are based on PowerShell and run in the cloud. Runbooks can also be executed on a server in your local data center to manage local resources. Azure Automation provides configuration management with PowerShell Desired State Configuration (DSC).



You can create and manage DSC resources that are hosted in Azure and apply them to cloud and on-premises systems. By doing so, you can define and automatically enforce their configuration or get reports on drift to help ensure that security configurations remain within policy.

Azure Security Center

Azure Security Center helps protect your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions. Within the service, you can define policies against both your Azure subscriptions and [resource groups](#) for greater granularity.



Microsoft security researchers are constantly on the lookout for threats. They have access to an expansive set of telemetry gained from Microsoft's global presence in the cloud and on-premises. This wide-reaching and diverse collection of datasets enables Microsoft to discover new attack patterns and trends across its on-premises consumer and enterprise products, as well as its online services.

Thus, Security Center 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.



Security Center threat detection works by automatically collecting security information from your Azure resources, the network, and connected partner solutions. It analyzes this information, correlating information from multiple sources, to identify threats.

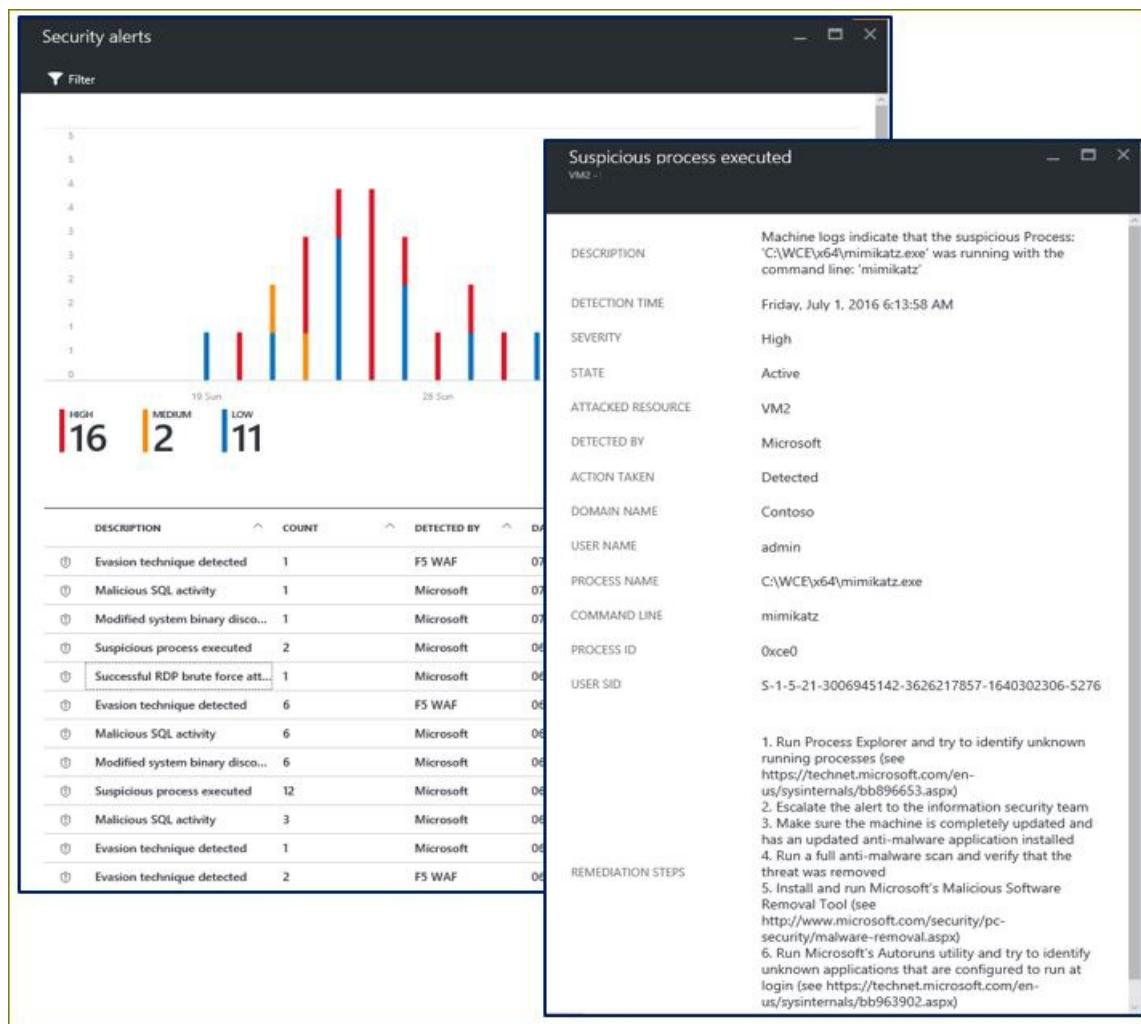
Security alerts are prioritized in Security Center along with recommendations on how to remediate the threat.

Security Center employs advanced security analytics, which go far beyond signature-based approaches. Breakthroughs in big data and [machine learning](#) technologies are used to evaluate events across the entire cloud fabric. Advanced analytics can detect threats that would be impossible to identify through manual approaches and predicting the evolution of attacks. These security analytics types are covered in the next sections.

Threat intelligence

Microsoft has access to an immense amount of global threat intelligence.

Telemetry flows in from multiple sources, such as Azure, Office 365, Microsoft CRM online, Microsoft Dynamics AX, outlook.com, MSN.com, the Microsoft Digital Crimes Unit (DCU), and Microsoft Security Response Center (MSRC).



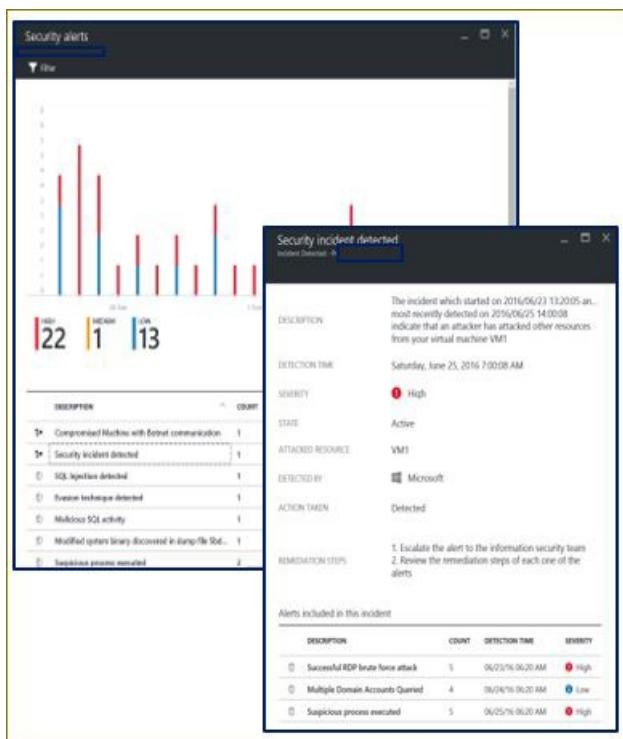
Researchers also receive threat intelligence information that is shared among major cloud service providers, and they subscribe to threat intelligence feeds from third parties. Azure Security Center can use this information to alert you to threats from known bad actors. Some examples include:

- **Harnessing the power of machine learning:** Azure Security Center has access to a vast amount of data about cloud network activity, which can be used to detect threats targeting your Azure deployments.
- **Brute force detection:** Machine learning is used to create a historical pattern of remote access attempts, which allows it to detect brute force attacks against Secure Shell (SSH), Remote Desktop Protocol (RDP), and SQL ports.

- **Outbound DDoS and botnet detection:** A common objective of attacks that target cloud resources is to use the compute power of these resources to execute other attacks.
- **New behavioral analytics servers and VMs:** After a server or virtual machine is compromised, attackers employ a wide variety of techniques to execute malicious code on that system while avoiding detection, ensuring persistence, and obviating security controls.
- **Azure SQL Database Threat Detection:** Threat detection for Azure SQL Database, which identifies anomalous database activities that indicate unusual and potentially harmful attempts to access or exploit databases.

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.



The patterns are also determined through careful analysis of malicious behaviors by expert analysts. Azure Security Center can use behavioral analytics to identify compromised resources based on analysis of virtual machine logs, virtual network device logs, fabric logs, crash dumps, and other sources.

In addition, patterns are correlated with other signals to check for supporting evidence of a widespread campaign. This correlation helps to identify events that are consistent with established indicators of compromise.

Some examples include:

- **Suspicious process execution:** Attackers employ several techniques to execute malicious software without detection. For example, an attacker might give malware the same names as legitimate system files but place these files in an alternate location, use a name that is similar to that of a benign file, or mask the file's true extension. Security Center models process behaviors and monitor process executions to detect outliers such as these.
- **Hidden malware and exploitation attempts:** Sophisticated malware can evade traditional antimalware products by either never writing to disk or encrypting software components stored on disk. However, such malware can be detected by using memory analysis, because the malware must leave traces in memory to function. When software crashes, a crash dump captures a portion of memory at the time of the crash. By analyzing the memory in the crash dump, Azure Security Center can detect techniques used to exploit

vulnerabilities in software, access confidential data, and surreptitiously persist within a compromised machine without affecting the performance of your machine.

- **Lateral movement and internal reconnaissance:** To persist in a compromised network and locate and harvest valuable data, attackers often attempt to move laterally from the compromised machine to others within the same network. Security Center monitors process and login activities to discover attempts to expand an attacker's foothold within the network, such as remote command execution, network probing, and account enumeration.
- **Malicious PowerShell scripts:** PowerShell can be used by attackers to execute malicious code on target virtual machines for various purposes. Security Center inspects PowerShell activity for evidence of suspicious activity.
- **Outgoing attacks:** Attackers often target cloud resources with the goal of using those resources to mount additional attacks. Compromised virtual machines, for example, might be used to launch brute force attacks against other virtual machines, send spam, or scan open ports and other devices on the internet. By applying machine learning to network traffic, Security Center can detect when outbound network communications exceed the norm. When spam is detected, Security Center also correlates unusual email traffic with intelligence from Office 365 to determine whether the mail is likely nefarious or the result of a legitimate email campaign.

Anomaly detection

Azure Security Center also uses anomaly detection to identify threats. In contrast to behavioral analytics (which 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. Here's an example:

- **Inbound RDP/SSH brute force attacks:** Your deployments might have busy virtual machines with many logins each day and other virtual machines that have few, if any, logins. Azure Security Center can determine baseline login activity for these virtual machines and use machine learning to define around the normal login activities. If there is any discrepancy with the baseline defined for login related characteristics, an alert might be generated. Again, machine learning determines what is significant.

Continuous threat intelligence monitoring

Azure Security Center operates with security research and data science teams throughout the world that continuously monitor for changes in the threat landscape. This includes the following initiatives:

- **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 the broad Microsoft portfolio of cloud and on-premises services, servers, and client endpoint devices are shared and analyzed.
- **Microsoft security specialists:** Ongoing engagement with teams across Microsoft that work in specialized security fields, such as forensics and web attack detection.
- **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.

Advanced threat detection features: Other Azure services

Virtual machines: Microsoft antimalware

[Microsoft antimalware](#) for Azure is a single-agent solution for applications and tenant environments, designed to run in the background without human intervention. You can deploy protection based on the needs of your application workloads, with either basic secure-by-default or advanced custom configuration, including antimalware monitoring. Azure antimalware is a security option for Azure virtual machines that's automatically installed on all Azure PaaS virtual machines.

Microsoft antimalware core features

Here are the features of Azure that deploy and enable Microsoft antimalware for your applications:

- **Real-time protection:** Monitors activity in cloud services and on virtual machines to detect and block malware execution.
- **Scheduled scanning:** Periodically performs targeted scanning to detect malware, including actively running programs.
- **Malware remediation:** Automatically acts on detected malware, such as deleting or quarantining malicious files and cleaning up malicious registry entries.
- **Signature updates:** Automatically installs the latest protection signatures (virus definitions) to ensure that protection is up to date on a pre-determined frequency.
- **Antimalware Engine updates:** Automatically updates the Microsoft Antimalware Engine.
- **Antimalware platform updates:** Automatically updates the Microsoft antimalware platform.
- **Active protection:** Reports telemetry metadata about detected threats and suspicious resources to Microsoft Azure to ensure rapid response to the evolving threat landscape, enabling real-time synchronous signature delivery through the Microsoft active protection system.
- **Samples reporting:** Provides and reports samples to the Microsoft antimalware service to help refine the service and enable troubleshooting.
- **Exclusions:** Allows application and service administrators to configure certain files, processes, and drives for exclusion from protection and scanning for performance and other reasons.
- **Antimalware event collection:** Records the antimalware service health, suspicious activities, and remediation actions taken in the operating system event log and collects them into the customer's Azure storage account.

Azure SQL Database Threat Detection

[Azure SQL Database Threat Detection](#) is a new security intelligence feature built into the Azure SQL Database service. Working around the clock to learn, profile, and detect anomalous database activities, Azure SQL Database Threat Detection identifies potential threats to the database.

Security officers or other designated administrators can get an immediate notification about suspicious database activities as they occur. Each notification provides details of the suspicious activity and recommends how to further investigate and mitigate the threat.

Currently, Azure SQL Database Threat Detection detects potential vulnerabilities and SQL injection attacks, and anomalous database access patterns.

Upon receiving a threat-detection email notification, users are able to navigate and view the relevant audit records through a deep link in the mail. The link opens an audit viewer or a preconfigured auditing Excel template that shows the relevant audit records around the time of the suspicious event, according to the following:

- Audit storage for the database/server with the anomalous database activities.
- Relevant audit storage table that was used at the time of the event to write the audit log.

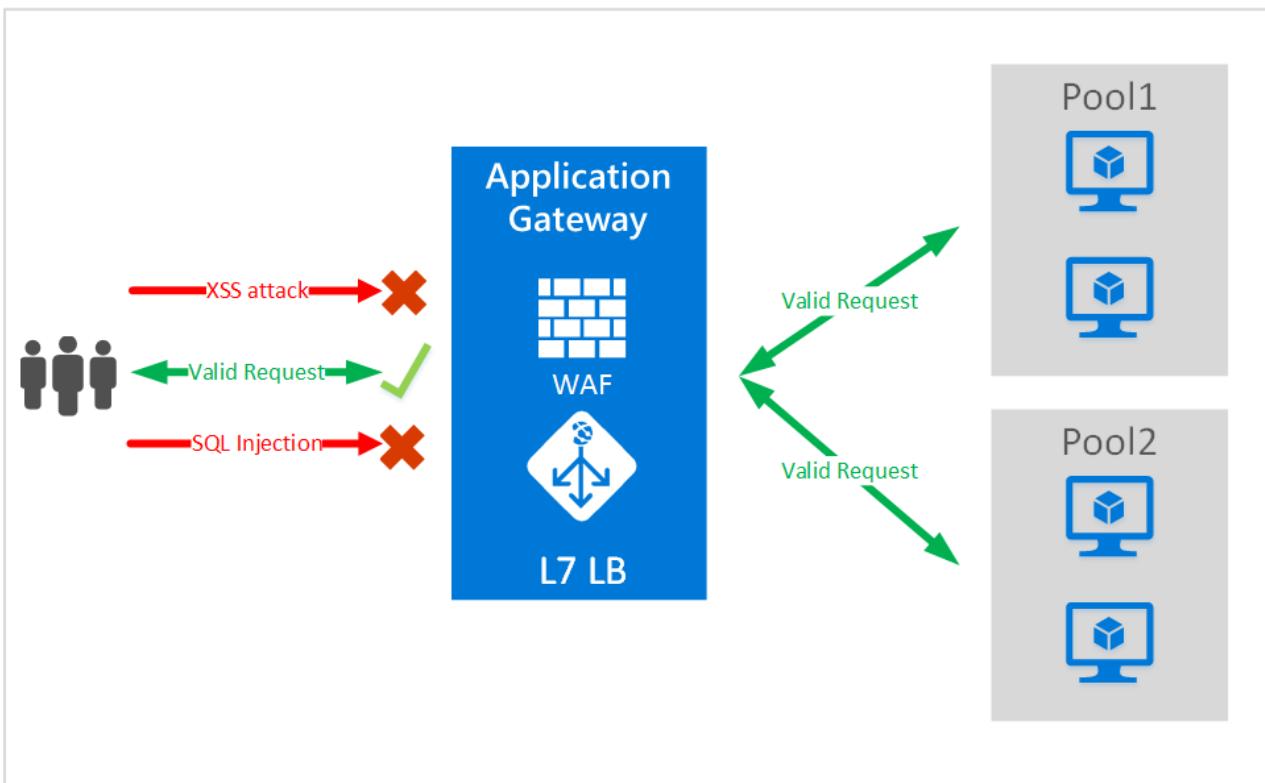
- Audit records of the hour immediately following the event occurrence.
- Audit records with a similar event ID at the time of the event (optional for some detectors).

SQL Database threat detectors use one of the following detection methodologies:

- **Deterministic detection:** Detects suspicious patterns (rules based) in the SQL client queries that match known attacks. This methodology has high detection and low false positive, but limited coverage because it falls within the category of "atomic detections."
- **Behavioral detection:** Detects anomalous activity, which is abnormal behavior in the database that was not seen during the most recent 30 days. Examples of SQL client anomalous activity can be a spike of failed logins or queries, a high volume of data being extracted, unusual canonical queries, or unfamiliar IP addresses used to access the database.

Application Gateway Web Application Firewall

[Web Application Firewall \(WAF\)](#) is a feature of [Azure Application Gateway](#) that provides protection to web applications that use an application gateway for standard [application delivery control](#) functions. Web Application Firewall does this by protecting them against most of the [Open Web Application Security Project \(OWASP\) top 10 common web vulnerabilities](#).



Protections include:

- SQL injection protection.
- Cross site scripting protection.
- Common Web Attacks Protection, such as command injection, HTTP request smuggling, HTTP response splitting, and remote file inclusion attack.
- Protection against HTTP protocol violations.
- Protection against HTTP protocol anomalies, such as missing host user-agent and accept headers.
- Prevention against bots, crawlers, and scanners.
- Detection of common application misconfigurations (that is, Apache, IIS, and so on).

Configuring WAF at your application gateway provides the following benefits:

- Protects your web application from web vulnerabilities and attacks without modification of the back-end code.
- Protects multiple web applications at the same time behind an application gateway. An application gateway supports hosting up to 20 websites.
- Monitors web applications against attacks by using real-time reports that are generated by application gateway WAF logs.
- Helps meet compliance requirements. Certain compliance controls require all internet-facing endpoints to be protected by a WAF solution.

Anomaly Detection API: Built with Azure Machine Learning

The Anomaly Detection API is an API that's useful for detecting a variety of anomalous patterns in your time series data. The API assigns an anomaly score to each data point in the time series, which can be used for generating alerts, monitoring through dashboards, or connecting with your ticketing systems.

The [Anomaly Detection API](#) can detect the following types of anomalies on time series data:

- **Spikes and dips:** When you're monitoring the number of login failures to a service or number of checkouts in an e-commerce site, unusual spikes or dips could indicate security attacks or service disruptions.
- **Positive and negative trends:** When you're monitoring memory usage in computing, shrinking free memory size indicates a potential memory leak. For service queue length monitoring, a persistent upward trend might indicate an underlying software issue.
- **Level changes and changes in dynamic range of values:** Level changes in latencies of a service after a service upgrade or lower levels of exceptions after upgrade can be interesting to monitor.

The machine learning-based API enables:

- **Flexible and robust detection:** The anomaly detection models allow users to configure sensitivity settings and detect anomalies among seasonal and non-seasonal data sets. Users can adjust the anomaly detection model to make the detection API less or more sensitive according to their needs. This would mean detecting the less or more visible anomalies in data with and without seasonal patterns.
- **Scalable and timely detection:** The traditional way of monitoring with present thresholds set by experts' domain knowledge are costly and not scalable to millions of dynamically changing data sets. The anomaly detection models in this API are learned, and models are tuned automatically from both historical and real-time data.
- **Proactive and actionable detection:** Slow trend and level change detection can be applied for early anomaly detection. The early abnormal signals that are detected can be used to direct humans to investigate and act on the problem areas. In addition, root cause analysis models and alerting tools can be developed on top of this anomaly-detection API service.

The anomaly-detection API is an effective and efficient solution for a wide range of scenarios, such as service health and KPI monitoring, IoT, performance monitoring, and network traffic monitoring. Here are some popular scenarios where this API can be useful:

- IT departments need tools to track events, error code, usage log, and performance (CPU, memory, and so on) in a timely manner.
- Online commerce sites want to track customer activities, page views, clicks, and so on.
- Utility companies want to track consumption of water, gas, electricity, and other resources.

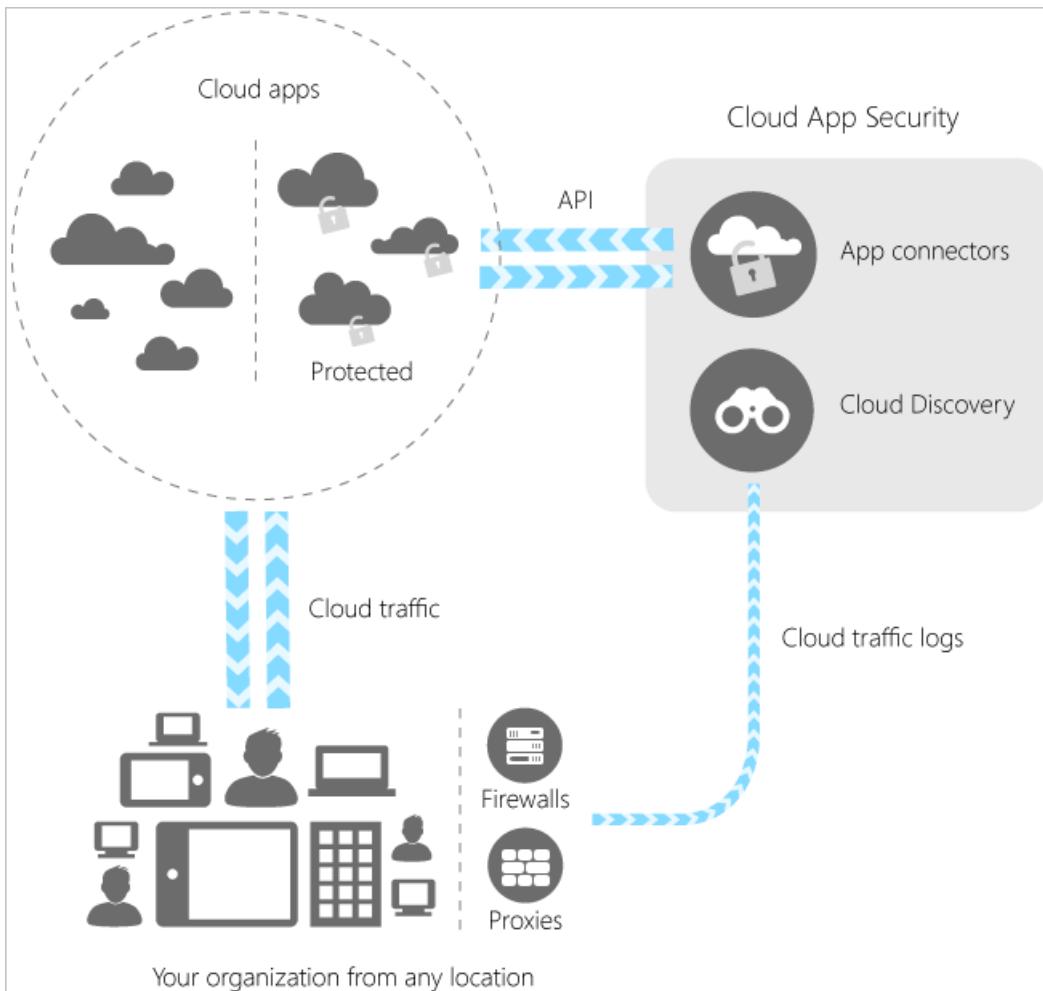
- Facility or building management services want to monitor temperature, moisture, traffic, and so on.
- IoT/manufacturers want to use sensor data in time series to monitor work flow, quality, and so on.
- Service providers, such as call centers, need to monitor service demand trend, incident volume, wait queue length, and so on.
- Business analytics groups want to monitor business KPIs' (such as sales volume, customer sentiments, or pricing) abnormal movement in real time.

Cloud App Security

[Cloud App Security](#) is a critical component of the Microsoft Cloud Security stack. It's a comprehensive solution that can help your organization as you move to take full advantage of the promise of cloud applications. It keeps you in control, through improved visibility into activity. It also helps increase the protection of critical data across cloud applications.

With tools that help uncover shadow IT, assess risk, enforce policies, investigate activities, and stop threats, your organization can more safely move to the cloud while maintaining control of critical data.

Discover	Uncover shadow IT with Cloud App Security. Gain visibility by discovering apps, activities, users, data, and files in your cloud environment. Discover third-party apps that are connected to your cloud.
Investigate	Investigate your cloud apps by using cloud forensics tools to deep-dive into risky apps, specific users, and files in your network. Find patterns in the data collected from your cloud. Generate reports to monitor your cloud.
Control	Mitigate risk by setting policies and alerts to achieve maximum control over network cloud traffic. Use Cloud App Security to migrate your users to safe, sanctioned cloud app alternatives.
Protect	Use Cloud App Security to sanction or prohibit applications, enforce data loss prevention, control permissions and sharing, and generate custom reports and alerts.
Control	Mitigate risk by setting policies and alerts to achieve maximum control over network cloud traffic. Use Cloud App Security to migrate your users to safe, sanctioned cloud app alternatives.



Cloud App Security integrates visibility with your cloud by:

- Using Cloud Discovery to map and identify your cloud environment and the cloud apps your organization is using.
- Sanctioning and prohibiting apps in your cloud.
- Using easy-to-deploy app connectors that take advantage of provider APIs, for visibility and governance of apps that you connect to.
- Helping you have continuous control by setting, and then continually fine-tuning, policies.

On collecting data from these sources, Cloud App Security runs sophisticated analysis on it. It immediately alerts you to anomalous activities, and gives you deep visibility into your cloud environment. You can configure a policy in Cloud App Security and use it to protect everything in your cloud environment.

Third-party Advanced Threat Detection capabilities through the Azure Marketplace

Web Application Firewall

Web Application Firewall inspects inbound web traffic and blocks SQL injections, cross-site scripting, malware uploads, application DDoS attacks, and other attacks targeted at your web applications. It also inspects the responses from the back-end web servers for data loss prevention (DLP). The integrated access control engine enables administrators to create granular access control policies for authentication, authorization, and accounting (AAA), which gives organizations strong authentication and user control.

Web Application Firewall provides the following benefits:

- Detects and blocks SQL injections, Cross-Site Scripting, malware uploads, application DDoS, or any other

attacks against your application.

- Authentication and access control.
- Scans outbound traffic to detect sensitive data and can mask or block the information from being leaked out.
- Accelerates the delivery of web application contents, using capabilities such as caching, compression, and other traffic optimizations.

For examples of web application firewalls that are available in the Azure Marketplace, see [Barracuda WAF](#), [Brocade virtual web application firewall \(vWAF\)](#), [Imperva SecureSphere](#), and the [ThreatSTOP IP firewall](#).

Next steps

- [Azure Security Center detection capabilities](#): Helps identify active threats that target your Azure resources and provides the insights you need to respond quickly.
- [Azure SQL Database Threat Detection](#): Helps address your concerns about potential threats to your databases.

Azure logging and auditing

12/4/2018 • 20 minutes to read • [Edit Online](#)

Azure provides a wide array of configurable security auditing and logging options to help you identify gaps in your security policies and mechanisms. This article discusses generating, collecting, and analyzing security logs from services hosted on Azure.

NOTE

Certain recommendations in this article might result in increased data, network, or compute resource usage, and increase your license or subscription costs.

Types of logs in Azure

Cloud applications are complex, with many moving parts. Logs provide data to help keep your applications up and running. Logs help you troubleshoot past problems or prevent potential ones. And they can help improve application performance or maintainability, or automate actions that would otherwise require manual intervention.

Azure logs are categorized into the following types:

- **Control/management logs** provide information about Azure Resource Manager CREATE, UPDATE, and DELETE operations. For more information, see [Azure activity logs](#).
- **Data plane logs** provide information about events raised as part Azure resource usage. Examples of this type of log are the Windows event system, security, and application logs in a virtual machine (VM) and the [diagnostics logs](#) that are configured through Azure Monitor.
- **Processed events** provide information about analyzed events/alerts that have been processed on your behalf. Examples of this type are [Azure Security Center alerts](#) where [Azure Security Center](#) has processed and analyzed your subscription and provides concise security alerts.

The following table lists the most important types of logs available in Azure:

LOG CATEGORY	LOG TYPE	USAGE	INTEGRATION
Activity logs	Control-plane events on Azure Resource Manager resources	Provides insight into the operations that were performed on resources in your subscription.	Rest API, Azure Monitor
Azure diagnostics logs	Frequent data about the operation of Azure Resource Manager resources in subscription	Provides insight into operations that your resource itself performed.	Azure Monitor, Stream
Azure AD reporting	Logs and reports	Reports user sign-in activities and system activity information about users and group management.	Graph API

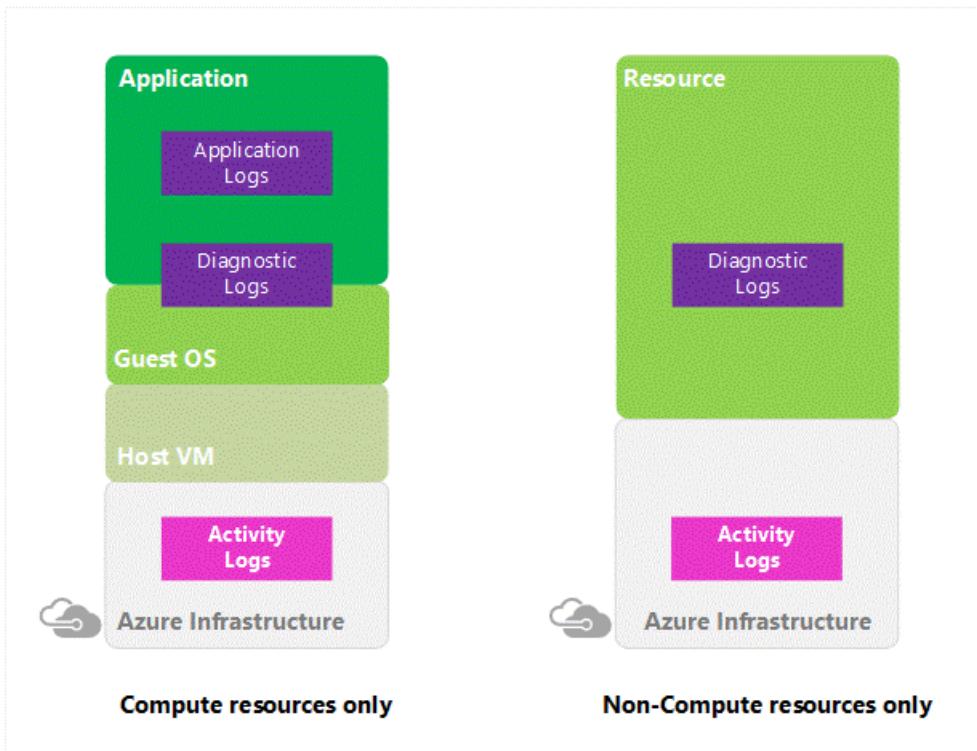
LOG CATEGORY	LOG TYPE	USAGE	INTEGRATION
Virtual machines and cloud services	Windows Event Log service and Linux Syslog	Captures system data and logging data on the virtual machines and transfers that data into a storage account of your choice.	Windows (using Windows Azure Diagnostics [WAD] storage) and Linux in Azure Monitor
Azure Storage Analytics	Storage logging, provides metrics data for a storage account	Provides insight into trace requests, analyzes usage trends, and diagnoses issues with your storage account.	REST API or the client library
Network Security Group (NSG) flow logs	JSON format, shows outbound and inbound flows on a per-rule basis	Displays information about ingress and egress IP traffic through a Network Security Group.	Azure Network Watcher
Application insight	Logs, exceptions, and custom diagnostics	Provides an application performance monitoring (APM) service for web developers on multiple platforms.	REST API, Power BI
Process data / security alerts	Azure Security Center alerts, Azure Log Analytics alerts	Provides security information and alerts.	REST APIs, JSON

Activity logs

[Azure activity logs](#) provide insight into the operations that were performed on resources in your subscription. Activity logs were previously known as "audit logs" or "operational logs," because they report [control-plane events](#) for your subscriptions.

Activity logs help you determine the "what, who, and when" for write operations (that is, PUT, POST, or DELETE). Activity logs also help you understand the status of the operation and other relevant properties. Activity logs do not include read (GET) operations.

In this article, PUT, POST, and DELETE refer to all the write operations that an activity log contains on the resources. For example, you can use the activity logs to find an error when you're troubleshooting issues or to monitor how a user in your organization modified a resource.



You can retrieve events from an activity log by using the Azure portal, [Azure CLI](#), PowerShell cmdlets, and [Azure Monitor REST API](#). Activity logs have 90-day data-retention period.

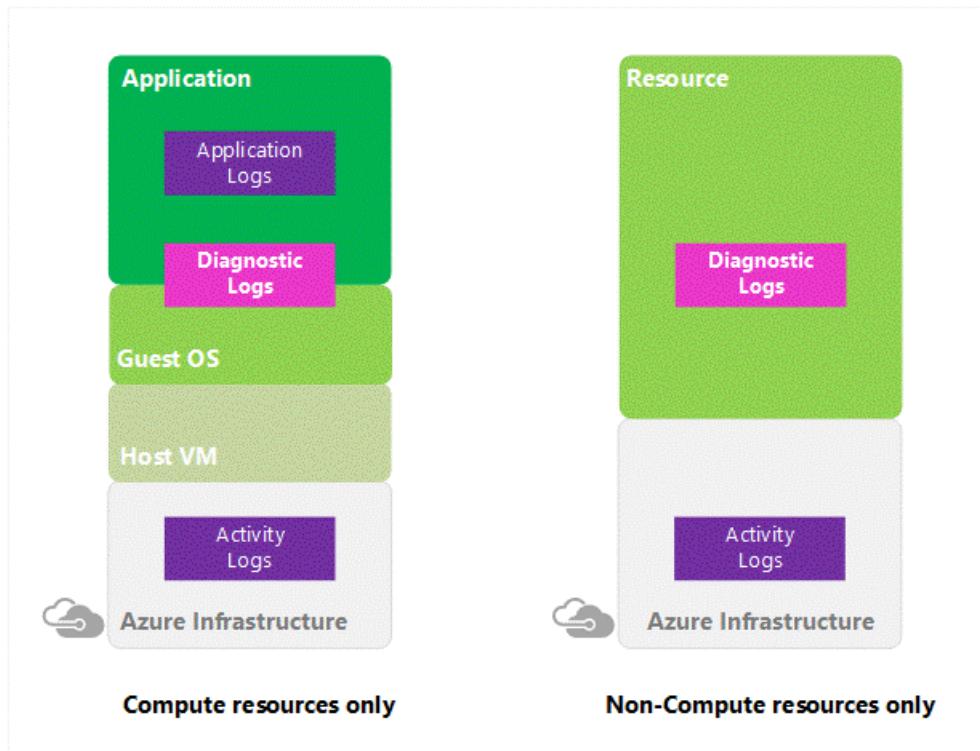
Integration scenarios for an activity log event:

- Create an email or webhook alert that's triggered by an activity log event.
- Stream it to an event hub for ingestion by a third-party service or custom analytics solution such as PowerBI.
- Analyze it in PowerBI by using the [PowerBI content pack](#).
- Save it to a storage account for archival or manual inspection. You can specify the retention time (in days) by using log profiles.
- Query and view it in the Azure portal.
- Query it via PowerShell cmdlet, Azure CLI, or REST API.
- Export the activity log with log profiles to [Log Analytics](#).

You can use a storage account or [event hub namespace](#) that is not in the same subscription as the one that's emitting the log. Whoever configures the setting must have the appropriate [role-based access control \(RBAC\)](#) access to both subscriptions.

Azure diagnostics logs

Azure diagnostics logs are emitted by a resource that provides rich, frequent data about the operation of that resource. The content of these logs varies by resource type. For example, [Windows event system logs](#) are a category of diagnostics logs for VMs, and [blob, table, and queue logs](#) are categories of diagnostics logs for storage accounts. Diagnostics logs differ from activity logs, which provide insight into the operations that were performed on resources in your subscription.



Azure diagnostics logs offer multiple configuration options, such as the Azure portal, PowerShell, Azure CLI, and the REST API.

Integration scenarios

- Save them to a [storage account](#) for auditing or manual inspection. You can specify the retention time (in days) by using the diagnostics settings.
- Stream them to [event hubs](#) for ingestion by a third-party service or custom analytics solution, such as [PowerBI](#).
- Analyze them with [Log Analytics](#).

Supported services, schema for diagnostics logs and supported log categories per resource type

Service	Schema and Documentation	Resource Type	Category
Azure Load Balancer	Log Analytics for Load Balancer (Preview)	Microsoft.Network/loadBalancers Microsoft.Network/loadBalancers	LoadBalancerAlertEvent LoadBalancerProbeHealthStatus
Network Security Groups	Log Analytics for Network Security Groups	Microsoft.Network/networks Microsoft.Network/networks Microsoft.Network/networks Microsoft.Network/networks	NetworkSecurityGroupEvent NetworkSecurityGroupRuleCounter
Azure Application Gateway	Diagnostics logging for Application Gateway	Microsoft.Network/applicationGateways Microsoft.Network/applicationGateways Microsoft.Network/applicationGateways Microsoft.Network/applicationGateways	ApplicationGatewayAccessLog ApplicationGatewayPerformanceLog ApplicationGatewayFirewallLog
Azure Key Vault	Key Vault logs	Microsoft.KeyVault/vaults	AuditEvent

Service	Schema and Documentation	Resource Type	Category
Azure Search	Enabling and using Search Traffic Analytics	Microsoft.Search/searchServices	OperationLogs
Azure Data Lake Store	Access diagnostics logs for Data Lake Store	Microsoft.DataLakeStore/accounts Microsoft.DataLakeStore/accounts	Audit Requests
Azure Data Lake Analytics	Access diagnostics logs for Data Lake Analytics	Microsoft.DataLakeAnalytics/accounts Microsoft.DataLakeAnalytics/accounts	Audit Requests
Azure Logic Apps	Logic Apps B2B custom tracking schema	Microsoft.Logic/workflows Microsoft.Logic/integrationAccounts	WorkflowRuntime IntegrationAccountTrackingEvents
Azure Batch	Azure Batch diagnostics logs	Microsoft.Batch/batchAccounts	ServiceLog
Azure Automation	Log Analytics for Azure Automation	Microsoft.Automation/automationAccounts Microsoft.Automation/automationAccounts	JobLogs JobStreams
Azure Event Hubs	Event Hubs diagnostics logs	Microsoft.EventHub/namespaces Microsoft.EventHub/namespaces	ArchiveLogs OperationalLogs
Azure Stream Analytics	Job diagnostics logs	Microsoft.StreamAnalytics/streamingjobs Microsoft.StreamAnalytics/streamingjobs	Execution Authoring
Azure Service Bus	Service Bus diagnostics logs	Microsoft.ServiceBus/namespaces	OperationalLogs

Azure Active Directory reporting

Azure Active Directory (Azure AD) includes security, activity, and audit reports for a user's directory. The [Azure AD audit report](#) helps you identify privileged actions that occurred in the user's Azure AD instance. Privileged actions include elevation changes (for example, role creation or password resets), changing policy configurations (for example, password policies), or changes to the directory configuration (for example, changes to domain federation settings).

The reports provide the audit record for the event name, the user who performed the action, the target resource affected by the change, and the date and time (in UTC). Users can retrieve the list of audit events for Azure AD via the [Azure portal](#), as described in [View your audit logs](#).

The included reports are listed in the following table:

Security Reports	Activity Reports	Audit Reports
Sign-ins from unknown sources	Application usage: summary	Directory audit report

SECURITY REPORTS	ACTIVITY REPORTS	AUDIT REPORTS
Sign-ins after multiple failures	Application usage: detailed	
Sign-ins from multiple geographies	Application dashboard	
Sign-ins from IP addresses with suspicious activity	Account provisioning errors	
Irregular sign-in activity	Individual user devices	
Sign-ins from possibly infected devices	Individual user activity	
Users with anomalous sign-in activity	Groups activity report	
	Password reset registration activity report	
	Password reset activity	

The data in these reports can be useful to your applications, such as Security Information and Event Management (SIEM) systems, audit, and business intelligence tools. The Azure AD reporting APIs provide programmatic access to the data through a set of REST-based APIs. You can call these [APIs](#) from various programming languages and tools.

Events in the Azure AD audit report are retained for 180 days.

NOTE

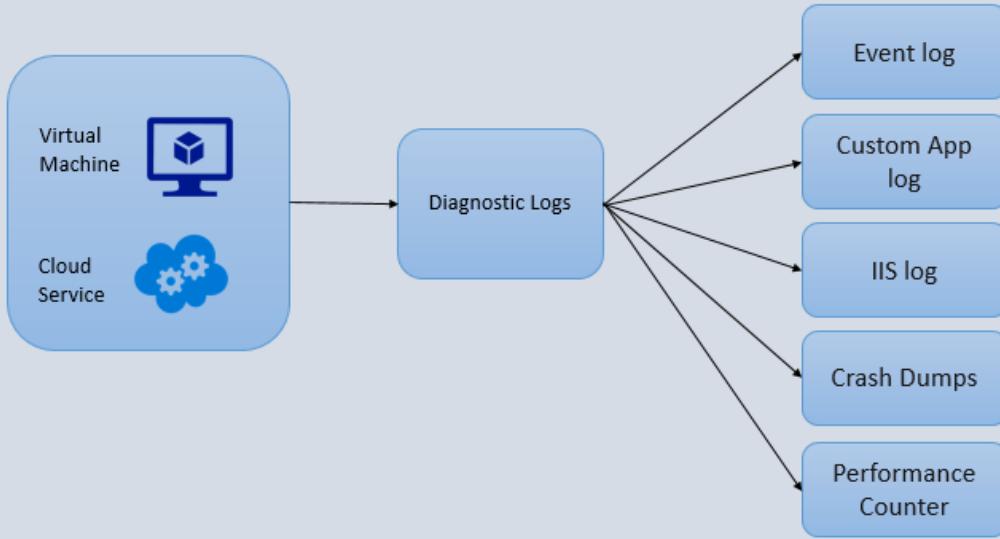
For more information about report retention, see [Azure AD report retention policies](#).

If you're interested in retaining your audit events longer, use the Reporting API to regularly pull[audit events](#) into a separate data store.

Virtual machine logs that use Azure Diagnostics

[Azure Diagnostics](#) is the capability within Azure that enables the collection of diagnostics data on a deployed application. You can use the diagnostics extension from any of several sources. Currently supported are [Azure cloud service web and worker roles](#).

Azure Cloud Service & VM Logs



Azure virtual machines that are running Microsoft Windows and Service Fabric

You can enable Azure Diagnostics on a virtual machine by doing any of the following:

- [Use Visual Studio to trace Azure virtual machines](#)
- [Set up Azure Diagnostics remotely on an Azure virtual machine](#)
- [Use PowerShell to set up diagnostics on Azure virtual machines](#)
- [Create a Windows virtual machine with monitoring and diagnostics by using an Azure Resource Manager template](#)

Storage Analytics

[Azure Storage Analytics](#) logs and provides metrics data for a storage account. You can use this data to trace requests, analyze usage trends, and diagnose issues with your storage account. Storage Analytics logging is available for the [Azure Blob](#), [Azure Queue](#), and [Azure Table storage services](#). Storage Analytics logs detailed information about successful and failed requests to a storage service.

You can use this information to monitor individual requests and to diagnose issues with a storage service. Requests are logged on a best-effort basis. Log entries are created only if there are requests made against the service endpoint. For example, if a storage account has activity in its blob endpoint but not in its table or queue endpoints, only logs that pertain to the Blob storage service are created.

To use Storage Analytics, enable it individually for each service you want to monitor. You can enable it in the [Azure portal](#). For more information, see [Monitor a storage account in the Azure portal](#). You can also enable Storage Analytics programmatically via the REST API or the client library. Use the Set Service Properties operation to enable Storage Analytics individually for each service.

The aggregated data is stored in a well-known blob (for logging) and in well-known tables (for metrics), which you can access by using the Blob storage service and Table storage service APIs.

Storage Analytics has a 20-terabyte (TB) limit on the amount of stored data that is independent of the total limit for your storage account. All logs are stored in [block blobs](#) in a container named \$logs, which is automatically created when you enable Storage Analytics for a storage account.

NOTE

- For more information about billing and data retention policies, see [Storage Analytics and billing](#).
- For more information about storage account limits, see [Azure Storage scalability and performance targets](#).

Storage Analytics logs the following types of authenticated and anonymous requests:

AUTHENTICATED	ANONYMOUS
Successful requests	Successful requests
Failed requests, including timeout, throttling, network, authorization, and other errors	Requests using a shared access signature, including failed and successful requests
Requests using a shared access signature, including failed and successful requests	Time-out errors for both client and server
Requests to analytics data	Failed GET requests with error code 304 (not modified)
Requests made by Storage Analytics itself, such as log creation or deletion, are not logged. A full list of the logged data is documented in Storage Analytics logged operations and status messages and Storage Analytics log format .	All other failed anonymous requests are not logged. A full list of the logged data is documented in Storage Analytics logged operations and status messages and Storage Analytics log format .

Azure networking logs

Network logging and monitoring in Azure is comprehensive and covers two broad categories:

- [Network Watcher](#): Scenario-based network monitoring is provided with the features in Network Watcher. This service includes packet capture, next hop, IP flow verify, security group view, NSG flow logs. Scenario level monitoring provides an end to end view of network resources in contrast to individual network resource monitoring.
- [Resource monitoring](#): Resource level monitoring comprises four features, diagnostics logs, metrics, troubleshooting, and resource health. All these features are built at the network resource level.

Network Watcher is a regional service that enables you to monitor and diagnose conditions at a network scenario level in, to, and from Azure. Network diagnostics and visualization tools available with Network Watcher help you

understand, diagnose, and gain insights to your network in Azure.

Network Security Group flow logging

NSG flow logs are a feature of Network Watcher that you can use to view information about ingress and egress IP traffic through an NSG. These flow logs are written in JSON format and show:

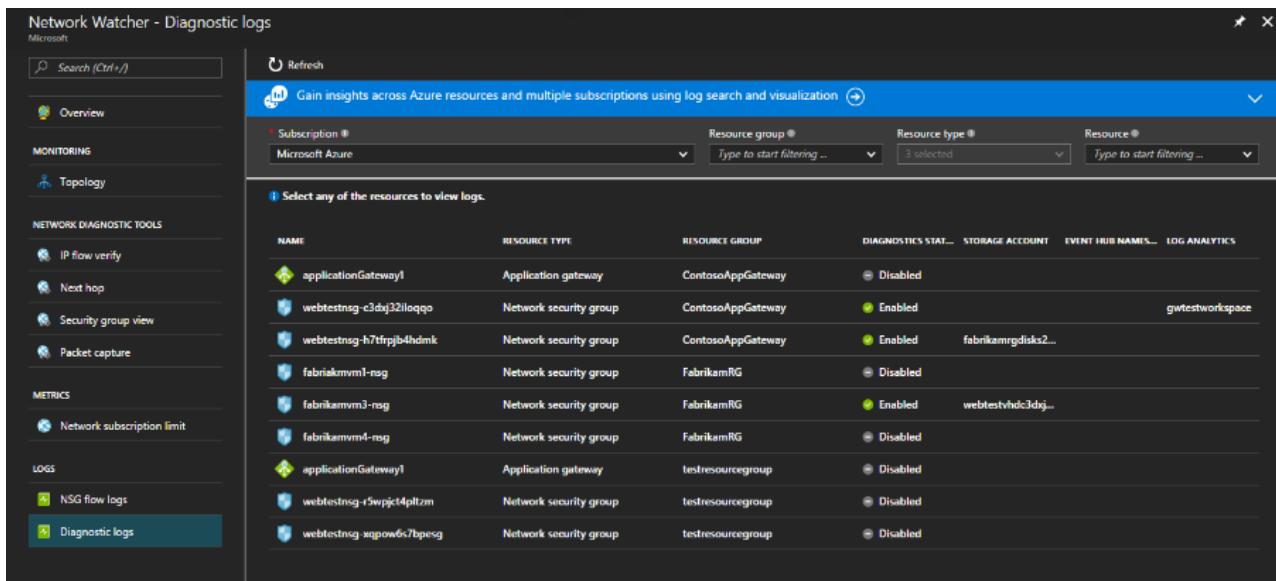
- Outbound and inbound flows on a per-rule basis.
- The NIC that the flow applies to.
- 5-tuple information about the flow: the source or destination IP, the source or destination port, and the protocol.
- Whether the traffic was allowed or denied.

Although flow logs target NSGs, they are not displayed in the same way as the other logs. Flow logs are stored only within a storage account.

The same retention policies that are seen on other logs apply to flow logs. Logs have a retention policy that you can set from 1 day to 365 days. If a retention policy is not set, the logs are maintained forever.

Diagnostics logs

Periodic and spontaneous events are created by network resources and logged in storage accounts, and sent to an event hub or Log Analytics. The logs provide insights into the health of a resource. They can be viewed in tools such as Power BI and Log Analytics. To learn how to view diagnostics logs, see [Log Analytics](#).



The screenshot shows the Network Watcher - Diagnostic logs interface. On the left, there's a navigation sidebar with sections like Overview, MONITORING, Topology, NETWORK DIAGNOSTIC TOOLS (IP flow verify, Next hop, Security group view, Packet capture), METRICS (Network subscription limit), LOGS (NSG flow logs, Diagnostic logs), and a search bar. The main area has a header with 'Gain insights across Azure resources and multiple subscriptions using log search and visualization' and filters for Subscription (Microsoft Azure), Resource group, Resource type, and Resource. Below this is a table titled 'Select any of the resources to view logs.' with columns: NAME, RESOURCE TYPE, RESOURCE GROUP, DIAGNOSTICS STATUS, STORAGE ACCOUNT, EVENT HUB NAMES, and LOG ANALYTICS. The table lists several resources: applicationGateway1 (Application gateway, ContosoAppGateway, Disabled), webtestnsg-c3d9j32lloqqo (Network security group, ContosoAppGateway, Enabled, fabrikamrgdisks2...), webtestnsg-h7frpjbj4hdmk (Network security group, ContosoAppGateway, Enabled, fabrikamrgdisks2...), fabrikamvmm1-nsg (Network security group, FabrikamRG, Disabled), fabrikamvmm3-nsg (Network security group, FabrikamRG, Enabled, webtestvhdc3d9j...), fabrikamvmm4-nsg (Network security group, FabrikamRG, Disabled), applicationGateway1 (Application gateway, testresourcegroup, Disabled), webtestnsg-r5wpjct4ptzm (Network security group, testresourcegroup, Disabled), and webtestnsg-xipow6s7bpsg (Network security group, testresourcegroup, Disabled). The 'Diagnostic logs' option in the sidebar is highlighted.

NAME	RESOURCE TYPE	RESOURCE GROUP	DIAGNOSTICS STATUS	STORAGE ACCOUNT	EVENT HUB NAMES	LOG ANALYTICS
applicationGateway1	Application gateway	ContosoAppGateway	Disabled			
webtestnsg-c3d9j32lloqqo	Network security group	ContosoAppGateway	Enabled			gatetestworkspace
webtestnsg-h7frpjbj4hdmk	Network security group	ContosoAppGateway	Enabled			
fabrikamvmm1-nsg	Network security group	FabrikamRG	Disabled			
fabrikamvmm3-nsg	Network security group	FabrikamRG	Enabled			
fabrikamvmm4-nsg	Network security group	FabrikamRG	Disabled			
applicationGateway1	Application gateway	testresourcegroup	Disabled			
webtestnsg-r5wpjct4ptzm	Network security group	testresourcegroup	Disabled			
webtestnsg-xipow6s7bpsg	Network security group	testresourcegroup	Disabled			

Diagnostics logs are available for [Load Balancer](#), [Network Security Groups](#), Routes, and [Application Gateway](#).

Network Watcher provides a diagnostics logs view. This view contains all networking resources that support diagnostics logging. From this view, you can enable and disable networking resources conveniently and quickly.

In addition to the previously mentioned logging capabilities, Network Watcher currently has the following capabilities:

- **Topology:** Provides a network-level view that shows the various interconnections and associations between network resources in a resource group.
- **Variable packet capture:** Captures packet data in and out of a virtual machine. Advanced filtering options and fine-tuning controls, such as time- and size-limitation settings, provide versatility. The packet data can be stored in a blob store or on the local disk in .cap file format.
- **IP flow verification:** Checks to see whether a packet is allowed or denied based on flow information 5-tuple packet parameters (that is, destination IP, source IP, destination port, source port, and protocol). If the

packet is denied by a security group, the rule and group that denied the packet is returned.

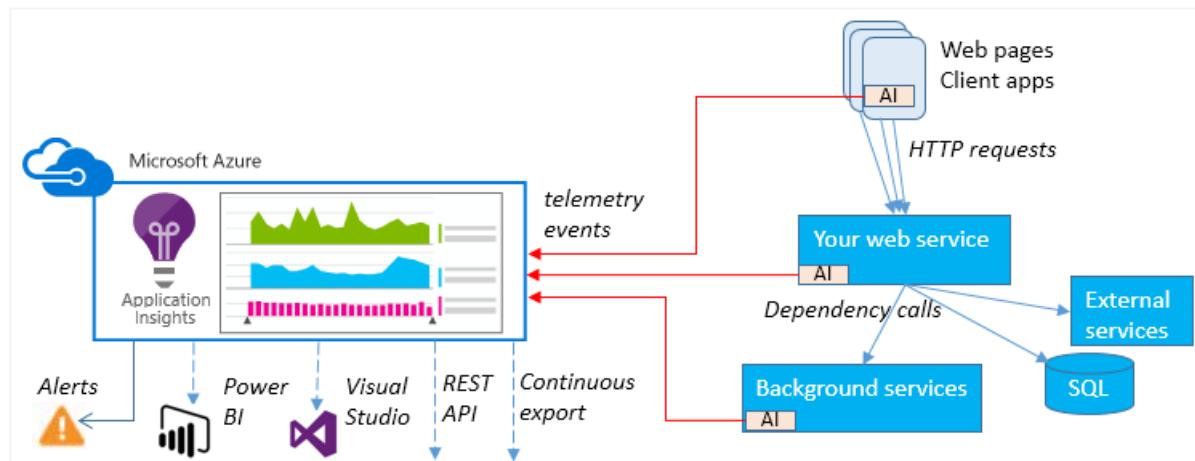
- [Next hop](#): Determines the next hop for packets being routed in the Azure network fabric, so that you can diagnose any misconfigured user-defined routes.
- [Security group view](#): Gets the effective and applied security rules that are applied on a VM.
- [Virtual network gateway and connection troubleshooting](#): Helps you troubleshoot virtual network gateways and connections.
- [Network subscription limits](#): Enables you to view network resource usage against limits.

Application Insights

[Azure Application Insights](#) is an extensible APM service for web developers on multiple platforms. Use it to monitor live web applications. It automatically detects performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app.

Application Insights is designed to help you continuously improve performance and usability.

It works for apps on a wide variety of platforms, including .NET, Node.js, and J2EE, whether they're hosted on-premises or in the cloud. It integrates with your DevOps process and has connection points with various development tools.



Application Insights is aimed at the development team, to help you understand how your app is performing and how it's being used. It monitors:

- **Request rates, response times, and failure rates**: Find out which pages are most popular, at what times of day, and where your users are. See which pages perform best. If your response times and failure rates go high when there are more requests, you might have a resourcing problem.
- **Dependency rates, response times, and failure rates**: Find out whether external services are slowing you down.
- **Exceptions**: Analyze the aggregated statistics, or pick specific instances and drill into the stack trace and related requests. Both server and browser exceptions are reported.
- **Page views and load performance**: Get reports from your users' browsers.
- **AJAX calls**: Get webpage rates, response times, and failure rates.
- **User and session counts**.
- **Performance counters**: Get data from your Windows or Linux server machines, such as CPU, memory, and network usage.
- **Host diagnostics**: Get data from Docker or Azure.

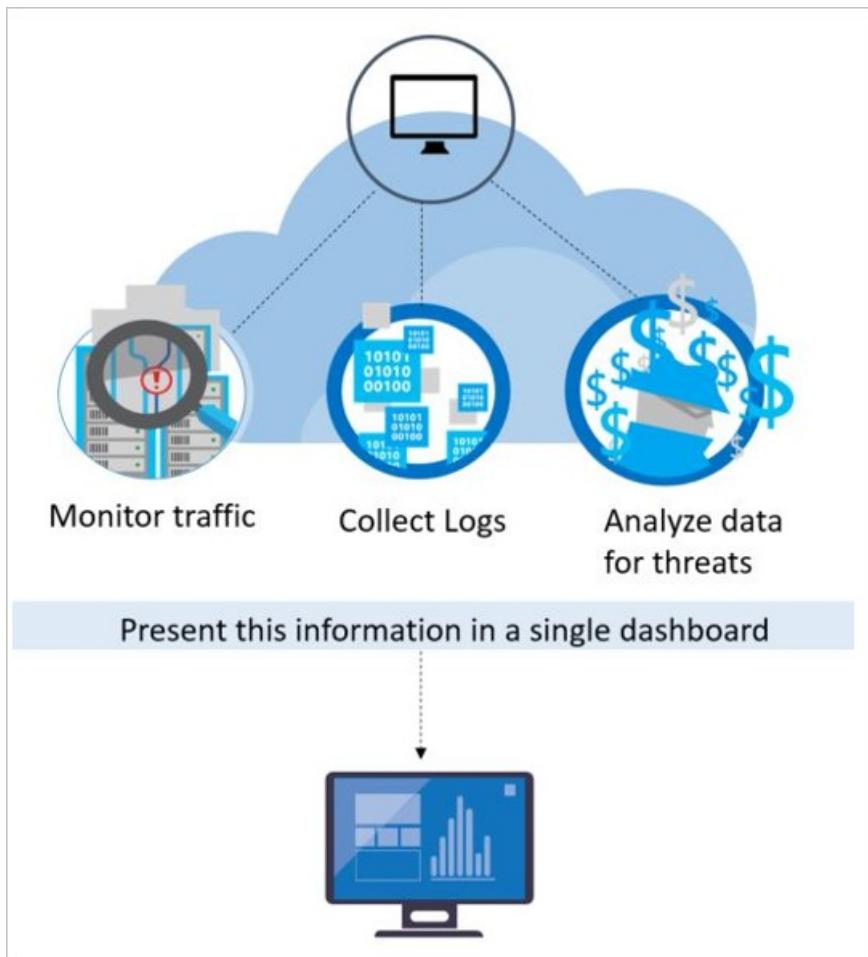
- **Diagnostics trace logs:** Get data from your app, so that you can correlate trace events with requests.
- **Custom events and metrics:** Get data that you write yourself in the client or server code, to track business events such as items sold or games won.

The following table lists and describes integration scenarios:

INTEGRATION SCENARIO	DESCRIPTION
Application map	The components of your app, with key metrics and alerts.
Diagnostics search for instance data	Search and filter events such as requests, exceptions, dependency calls, log traces, and page views.
Metrics Explorer for aggregated data	Explore, filter, and segment aggregated data such as rates of requests, failures, and exceptions; response times, page load times.
Dashboards	Mash up data from multiple resources and share with others. Great for multi-component applications, and for continuous display in the team room.
Live Metrics Stream	When you deploy a new build, watch these near-real-time performance indicators to make sure everything works as expected.
Analytics	Answer tough questions about your app's performance and usage by using this powerful query language.
Automatic and manual alerts	Automatic alerts adapt to your app's normal patterns of telemetry and are triggered when there's something outside the usual pattern. You can also set alerts on particular levels of custom or standard metrics.
Visual Studio	View performance data in the code. Go to code from stack traces.
Power BI	Integrate usage metrics with other business intelligence.
REST API	Write code to run queries over your metrics and raw data.
Continuous export	Bulk export of raw data to storage when it arrives.

Azure Security Center alerts

Azure Security Center threat detection works by automatically collecting security information from your Azure resources, the network, and connected partner solutions. It analyzes this information, often correlating information from multiple sources, to identify threats. Security alerts are prioritized in Security Center along with recommendations on how to remediate the threat. For more information, see [Azure Security Center](#).



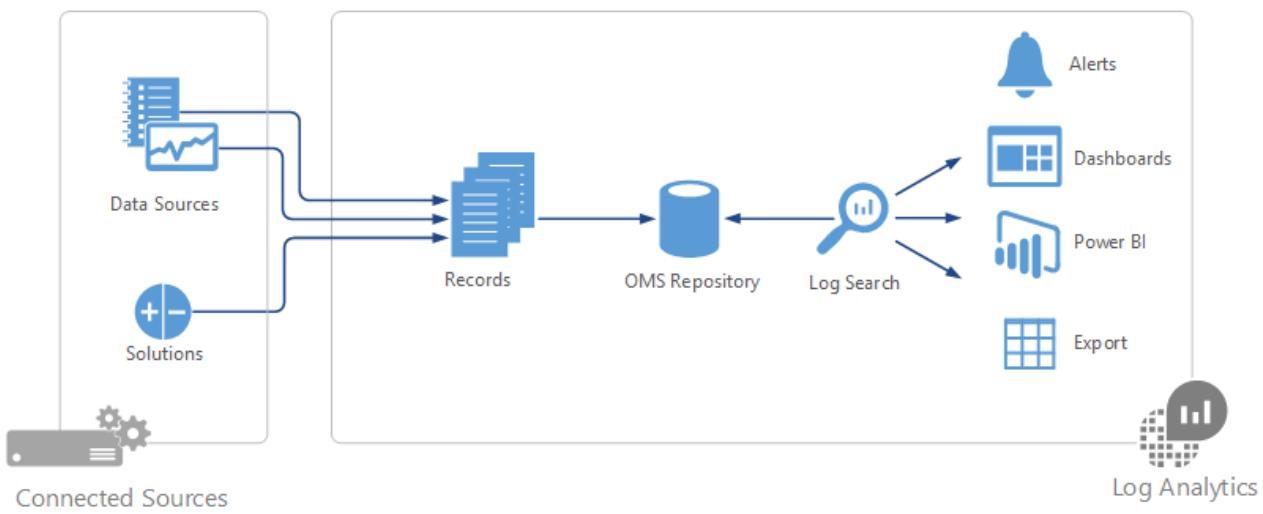
Security Center employs advanced security analytics, which go far beyond signature-based approaches. It applies breakthroughs in large data and [machine learning](#) technologies to evaluate events across the entire cloud fabric. In this way, it detects threats that would be impossible to identify by using manual approaches and predicting the evolution of attacks. These security analytics include:

- **Integrated threat intelligence:** Looks for known bad actors by applying global threat intelligence from Microsoft products and services, the Microsoft Digital Crimes Unit (DCU), the Microsoft Security Response Center (MSRC), and external feeds.
- **Behavioral analytics:** Applies known patterns to discover malicious behavior.
- **Anomaly detection:** Uses statistical profiling to build a historical baseline. It alerts on deviations from established baselines that conform to a potential attack vector.

Many security operations and incident response teams rely on a SIEM solution as the starting point for triaging and investigating security alerts. With Azure Log Integration, you can sync Security Center alerts and virtual machine security events, collected by Azure diagnostics and audit logs, with your Log Analytics or SIEM solution in near real time.

Log Analytics

Log Analytics is a service in Azure that helps you collect and analyze data that's generated by resources in your cloud and on-premises environments. It gives you real-time insights by using integrated search and custom dashboards to readily analyze millions of records across all your workloads and servers, regardless of their physical location.



At the center of Log Analytics is the Log Analytics workspace, which is hosted in Azure. Log Analytics collects data in the workspace from connected sources by configuring data sources and adding solutions to your subscription. Data sources and solutions each create different record types, each with its own set of properties. But sources and solutions can still be analyzed together in queries to the workspace. This capability allows you to use the same tools and methods to work with a variety of data collected by a variety of sources.

Connected sources are the computers and other resources that generate the data that's collected by Log Analytics. Sources can include agents that are installed on [Windows](#) and [Linux](#) computers that connect directly, or agents in [a connected System Center Operations Manager management group](#). Log Analytics can also collect data from an [Azure storage account](#).

[Data sources](#) are the various kinds of data that's collected from each connected source. Sources include events and [performance data](#) from [Windows](#) and Linux agents, in addition to sources such as [IIS logs](#) and [custom text logs](#). You configure each data source that you want to collect, and the configuration is automatically delivered to each connected source.

There are four ways to [collect logs and metrics for Azure services](#):

- Azure Diagnostics direct to Log Analytics (**Diagnostics** in the following table)
- Azure Diagnostics to Azure storage to Log Analytics (**Storage** in the following table)
- Connectors for Azure services (**Connector** in the following table)
- Scripts to collect and then post data into Log Analytics (blank cells in the following table and for services that are not listed)

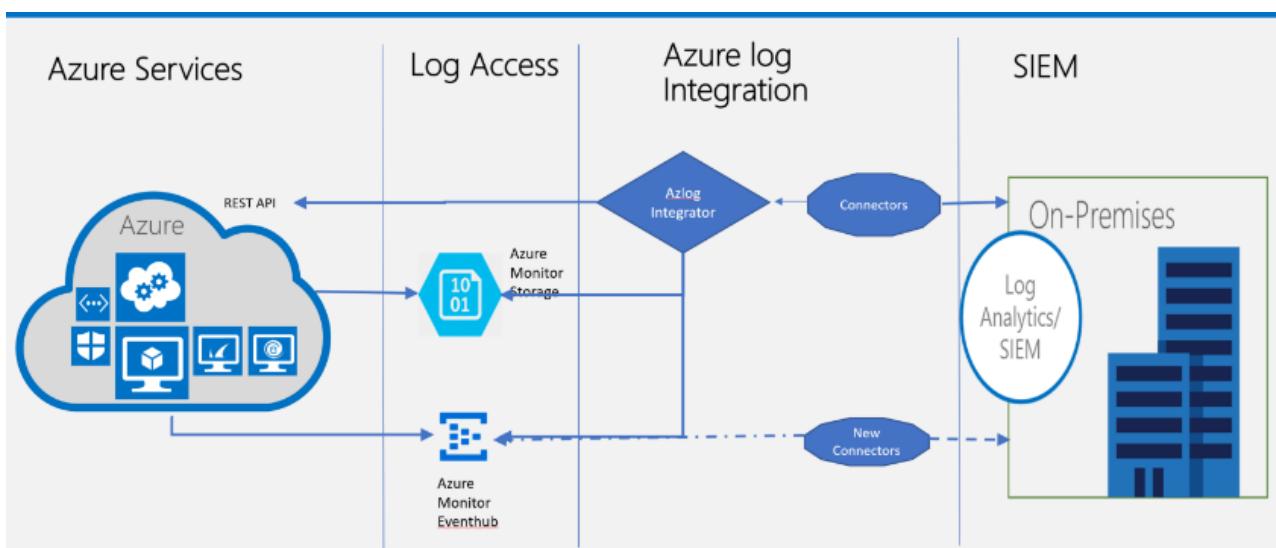
SERVICE	RESOURCE TYPE	LOGS	METRICS	SOLUTION
Azure Application Gateway	Microsoft.Network/applicationGateways	Diagnostics	Diagnostics	Azure Application Gateway Analytics
Application Insights		Connector	Connector	Application Insights Connector (Preview)
Azure Automation accounts	Microsoft.Automation/AutomationAccounts	Diagnostics		More information
Azure Batch accounts	Microsoft.Batch/batchAccounts	Diagnostics	Diagnostics	

Service	Resource type	Logs	Metrics	Solution
Classic cloud services		Storage		More information
Cognitive Services	Microsoft.CognitiveServices/accounts	Diagnostics		
Azure Data Lake Analytics	Microsoft.DataLakeAnalytics/accounts	Diagnostics		
Azure Data Lake Store	Microsoft.DataLakeStore/accounts	Diagnostics		
Azure Event Hub namespace	Microsoft.EventHub/namespaces	Diagnostics	Diagnostics	
Azure IoT Hub	Microsoft.Devices/IotHubs		Diagnostics	
Azure Key Vault	Microsoft.KeyVault/vaults	Diagnostics		Key Vault Analytics
Azure Load Balancer	Microsoft.Network/loadBalancers	Diagnostics		
Azure Logic Apps	Microsoft.Logic/workflows	Diagnostics	Diagnostics	
	Microsoft.Logic/integrationAccounts			
Network Security Groups	Microsoft.Network/networksecuritygroups	Diagnostics		Azure Network Security Group analytics
Recovery vaults	Microsoft.RecoveryServices/vaults			Azure Recovery Services Analytics (Preview)
Search services	Microsoft.Search/searchServices	Diagnostics	Diagnostics	
Service Bus namespace	Microsoft.ServiceBus/namespaces	Diagnostics	Diagnostics	Service Bus Analytics (Preview)
Service Fabric		Storage		Service Fabric Analytics (Preview)
SQL (v12)	Microsoft.Sql/servers/databases		Diagnostics	

Service	Resource Type	Logs	Metrics	Solution
	Microsoft.Sql/servers/elasticPools			
Storage			Script	Azure Storage Analytics (Preview)
Azure Virtual Machines	Microsoft.Compute/virtualMachines	Extension	Extension	
			Diagnostics	
Virtual machine scale sets	Microsoft.Compute/virtualMachines		Diagnostics	
	Microsoft.Compute/virtualMachineScaleSets/virtualMachines			
Web server farms	Microsoft.Web/serverfarms		Diagnostics	
Websites	Microsoft.Web/sites		Diagnostics	More information
	Microsoft.Web/sites/slots			

Log Integration with on-premises SIEM systems

With [Azure Log Integration](#), you can integrate raw logs from your Azure resources with your on-premises SIEM system (Security information and event management system).



Log Integration collects Azure diagnostics from your Windows virtual machines, Azure activity logs, Azure Security Center alerts, and Azure resource provider logs. This integration provides a unified dashboard for all your assets, whether they're on-premises or in the cloud, so that you can aggregate, correlate, analyze, and alert for security

events.

Log Integration currently supports the integration of Azure activity logs, Windows event logs from Windows virtual machines with your Azure subscription, Azure Security Center alerts, Azure diagnostics logs, and Azure AD audit logs.

LOG TYPE	LOG ANALYTICS SUPPORTING JSON (SPLUNK, ARCSIGHT, AND IBM QRADAR)
Azure AD audit logs	Yes
Activity logs	Yes
Security Center alerts	Yes
Diagnostics logs (resource logs)	Yes
VM logs	Yes, via forwarded events and not through JSON

[Get started with Azure Log Integration](#): This tutorial walks you through installing Azure Log Integration and integrating logs from Azure storage, Azure activity logs, Azure Security Center alerts, and Azure AD audit logs.

Integration scenarios for SIEM:

- [Partner configuration steps](#): This blog post shows you how to configure Azure Log Integration to work with partner solutions Splunk, HP ArcSight, and IBM QRadar.
- [Azure Log Integration FAQ](#): This article answers questions about Azure Log Integration.
- [Integrating Security Center alerts with Azure Log Integration](#): This article discusses how to sync Security Center alerts, virtual machine security events collected by Azure diagnostics logs, and Azure audit logs with your Log Analytics or SIEM solution.

Next steps

- [Auditing and logging](#): Protect data by maintaining visibility and responding quickly to timely security alerts.
- [Security logging and audit-log collection within Azure](#): Enforce these settings to ensure that your Azure instances are collecting the correct security and audit logs.
- [Configure audit settings for a site collection](#): If you're a site collection administrator, retrieve the history of individual users' actions and the history of actions taken during a particular date range.
- [Search the audit log in the Office 365 Security & Compliance Center](#): Use the Office 365 Security & Compliance Center to search the unified audit log and view user and administrator activity in your Office 365 organization.

Azure network security

9/5/2018 • 2 minutes to read • [Edit Online](#)

Abstract

Azure network services maximize flexibility, availability, resiliency, security, and integrity by design. This white paper provides details on the networking functions of Azure. It also describes how customers can use the native security features in Azure to help protect their information assets. The intended audiences for this white paper include:

- Technical managers, network administrators, and developers who are looking for security solutions that are available and supported in Azure.
- SMEs or business process executives who want a high-level overview of the Azure technologies and services that relate to network security in the Azure public cloud.

[Download the white paper](#)

Azure Functions and serverless platform security

12/12/2018 • 2 minutes to read • [Edit Online](#)

Abstract

Most enterprises need a significant amount of resources and time to manage servers, which adds cost. If enterprises can use fewer resources to manage servers, they can focus on building great applications.

Serverless computing helps you do just that, because the infrastructure that you need to run and scale your apps is managed for you. Serverless computing is the abstraction of servers, infrastructure, and operating systems.

Serverless computing is driven by the reaction to events and triggers, which are all taking place in near real-time—in the cloud.

As a fully managed service, server management and capacity planning are invisible to the developer. The serverless framework helps you develop and deploy serverless applications by using Azure Functions. It's a command-line interface (CLI) that offers structure and automation to help you build sophisticated, event-driven, serverless architectures composed of functions and events. An Azure function is an independent unit of deployment, like a microservice. It's merely code, deployed in the cloud, that is most often written to perform a single job.

Despite the benefits, serverless security has its own risk factors to deal with. The serverless approach doesn't introduce new security concerns, but it requires having an approach to existing security concerns. This white paper focuses on these security matters:

- Benefits of a serverless platform
- Security issues in serverless computing
- Critical security issues and mitigations in the context of Azure
- Securing the Microsoft serverless platform

[Download the white paper](#)

Container security in Microsoft Azure

9/5/2018 • 2 minutes to read • [Edit Online](#)

Abstract

Container technology is causing a structural change in the cloud-computing world. Containers make it possible to run multiple instances of an application on a single instance of an operating system, thereby using resources more efficiently. Containers give organizations consistency and flexibility. They enable continuous deployment because the application can be developed on a desktop, tested in a virtual machine, and then deployed for production in the cloud. Containers provide agility, streamlined operations, scalability, and reduced costs due to resource optimization.

Because container technology is relatively new, many IT professionals have security concerns about the lack of visibility and usage in a production environment. Development teams are often unaware of security best practices. This white paper can help security operations teams and developers in selecting approaches to secure container development and deployments on the Microsoft Azure platform.

This paper describes containers, container deployment and management, and native platform services. It also describes runtime security issues that arise with the use of containers on the Azure platform. In figures and examples, this paper focuses on Docker as the container model and Kubernetes as the container orchestrator. Most of the security recommendations also apply to other container models from Microsoft partners on the Azure platform.

[Download the white paper](#)

Azure Operational Security

9/5/2018 • 2 minutes to read • [Edit Online](#)

Abstract

Microsoft Azure operational security refers to the services, controls, and features available to users for protecting their data, applications, and other assets in Azure. Azure operational security is built on a framework that incorporates the knowledge gained through various capabilities that are unique to Microsoft, including the Microsoft Security Development Lifecycle (SDL), the Microsoft Security Response Center program, and deep awareness of the cybersecurity threat landscape. This white paper outlines how you can approach operational security by using Azure. It covers several Azure services, including:

- Azure Log Analytics
- Azure Backup
- Azure Security Center
- Azure Monitor
- Azure Network Watcher
- Azure Storage Analytics
- Azure Active Directory

[Download the white paper](#)

Isolation in the Azure Public Cloud

12/12/2018 • 22 minutes to read • [Edit Online](#)

Introduction

Overview

To assist current and prospective Azure customers understand and utilize the various security-related capabilities available in and surrounding the Azure platform, Microsoft has developed a series of White Papers, Security Overviews, Best Practices, and Checklists. The topics range in terms of breadth and depth and are updated periodically. This document is part of that series as summarized in the Abstract section following.

Azure Platform

Azure is an open and flexible cloud service platform that supports the broadest selection of operating systems, programming languages, frameworks, tools, databases, and devices. For example, you can:

- Run Linux containers with Docker integration;
- Build apps with JavaScript, Python, .NET, PHP, Java, and Node.js; and
- Build back-ends for iOS, Android, and Windows devices.

Microsoft Azure supports the same technologies millions of developers and IT professionals already rely on and trust.

When you build on, or migrate IT assets to, a public cloud service provider, you are relying on that organization's abilities to protect your applications and data with the services and the controls they provide to manage the security of your cloud-based assets.

Azure's infrastructure is designed from the facility to applications for hosting millions of customers simultaneously, and it provides a trustworthy foundation upon which businesses can meet their security needs. In addition, Azure provides you with a wide array of configurable security options and the ability to control them so that you can customize security to meet the unique requirements of your deployments. This document helps you meet these requirements.

Abstract

Microsoft Azure allows you to run applications and virtual machines (VMs) on shared physical infrastructure. One of the prime economic motivations to running applications in a cloud environment is the ability to distribute the cost of shared resources among multiple customers. This practice of multi-tenancy improves efficiency by multiplexing resources among disparate customers at low costs. Unfortunately, it also introduces the risk of sharing physical servers and other infrastructure resources to run your sensitive applications and VMs that may belong to an arbitrary and potentially malicious user.

This article outlines how Microsoft Azure provides isolation against both malicious and non-malicious users and serves as a guide for architecting cloud solutions by offering various isolation choices to architects. This white paper focuses on the technology of Azure platform and customer-facing security controls, and does not attempt to address SLAs, pricing models, and DevOps practice considerations.

Tenant Level Isolation

One of the primary benefits of cloud computing is concept of a shared, common infrastructure across numerous customers simultaneously, leading to economies of scale. This concept is called multi-tenancy. Microsoft works continuously to ensure that the multi-tenant architecture of Microsoft Cloud Azure supports security, confidentiality, privacy, integrity, and availability standards.

In the cloud-enabled workplace, a tenant can be defined as a client or organization that owns and manages a specific instance of that cloud service. With the identity platform provided by Microsoft Azure, a tenant is simply a dedicated instance of Azure Active Directory (Azure AD) that your organization receives and owns when it signs up for a Microsoft cloud service.

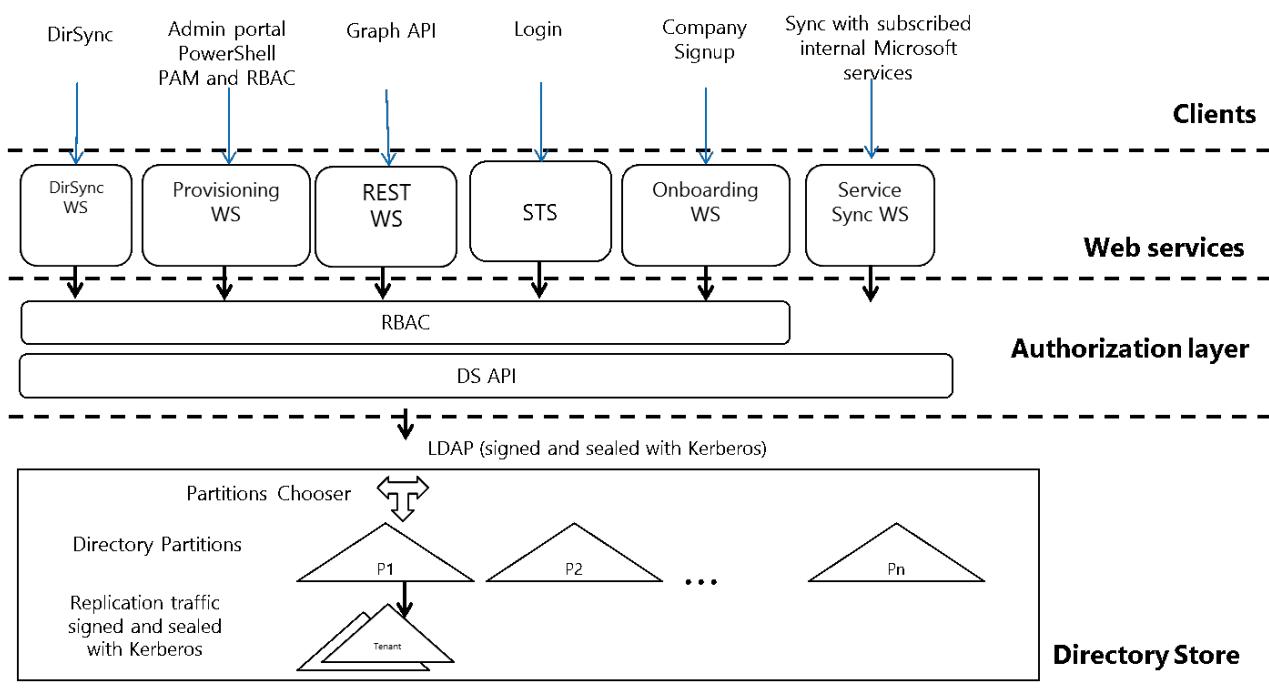
Each Azure AD directory is distinct and separate from other Azure AD directories. Just like a corporate office building is a secure asset specific to only your organization, an Azure AD directory was also designed to be a secure asset for use by only your organization. The Azure AD architecture isolates customer data and identity information from co-mingling. This means that users and administrators of one Azure AD directory cannot accidentally or maliciously access data in another directory.

Azure Tenancy

Azure tenancy (Azure Subscription) refers to a “customer/billing” relationship and a unique [tenant](#) in [Azure Active Directory](#). Tenant level isolation in Microsoft Azure is achieved using Azure Active Directory and [role-based controls](#) offered by it. Each Azure subscription is associated with one Azure Active Directory (AD) directory.

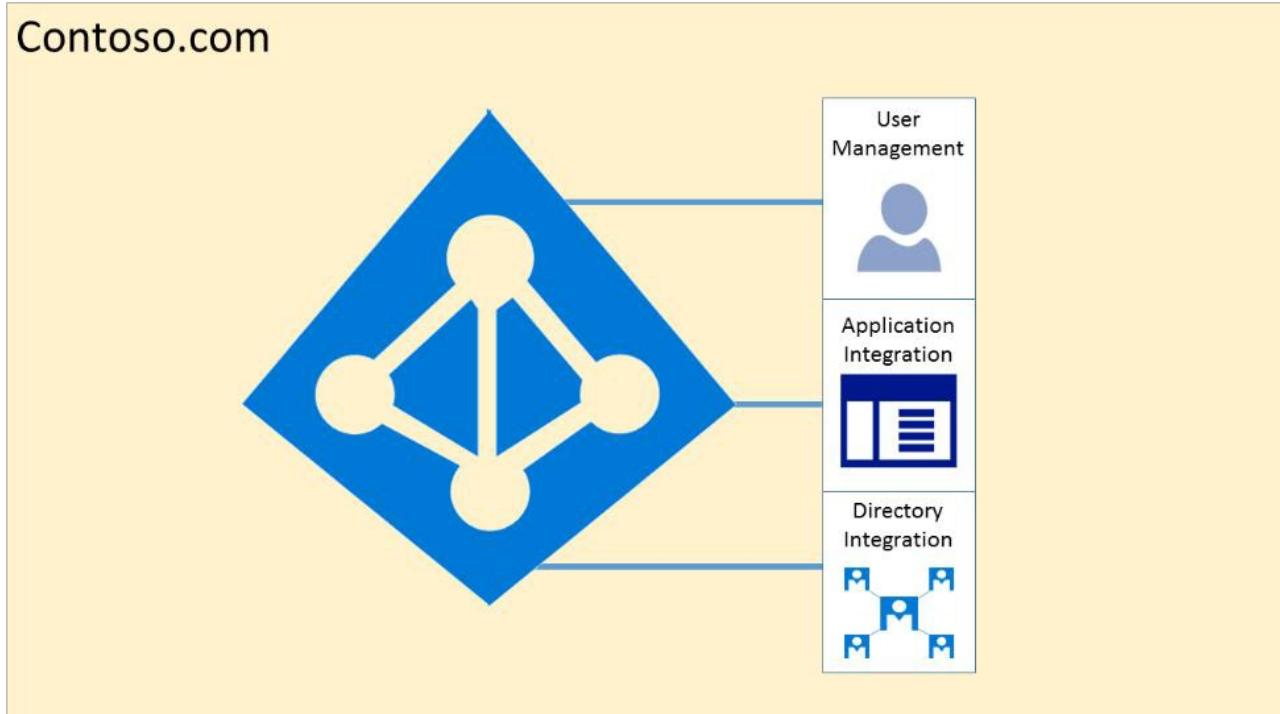
Users, groups, and applications from that directory can manage resources in the Azure subscription. You can assign these access rights using the Azure portal, Azure command-line tools, and Azure Management APIs. An Azure AD tenant is logically isolated using security boundaries so that no customer can access or compromise co-tenants, either maliciously or accidentally. Azure AD runs on “bare metal” servers isolated on a segregated network segment, where host-level packet filtering and Windows Firewall block unwanted connections and traffic.

- Access to data in Azure AD requires user authentication via a security token service (STS). Information on the user’s existence, enabled state, and role is used by the authorization system to determine whether the requested access to the target tenant is authorized for this user in this session.



- Tenants are discrete containers and there is no relationship between these.
- No access across tenants unless tenant admin grants it through federation or provisioning user accounts from other tenants.
- Physical access to servers that comprise the Azure AD service, and direct access to Azure AD’s back-end systems, is restricted.
- Azure AD users have no access to physical assets or locations, and therefore it is not possible for them to bypass the logical RBAC policy checks stated following.

For diagnostics and maintenance needs, an operational model that employs a just-in-time privilege elevation system is required and used. Azure AD Privileged Identity Management (PIM) introduces the concept of an eligible admin. [Eligible admins](#) should be users that need privileged access now and then, but not every day. The role is inactive until the user needs access, then they complete an activation process and become an active admin for a predetermined amount of time.



Azure Active Directory hosts each tenant in its own protected container, with policies and permissions to and within the container solely owned and managed by the tenant.

The concept of tenant containers is deeply ingrained in the directory service at all layers, from portals all the way to persistent storage.

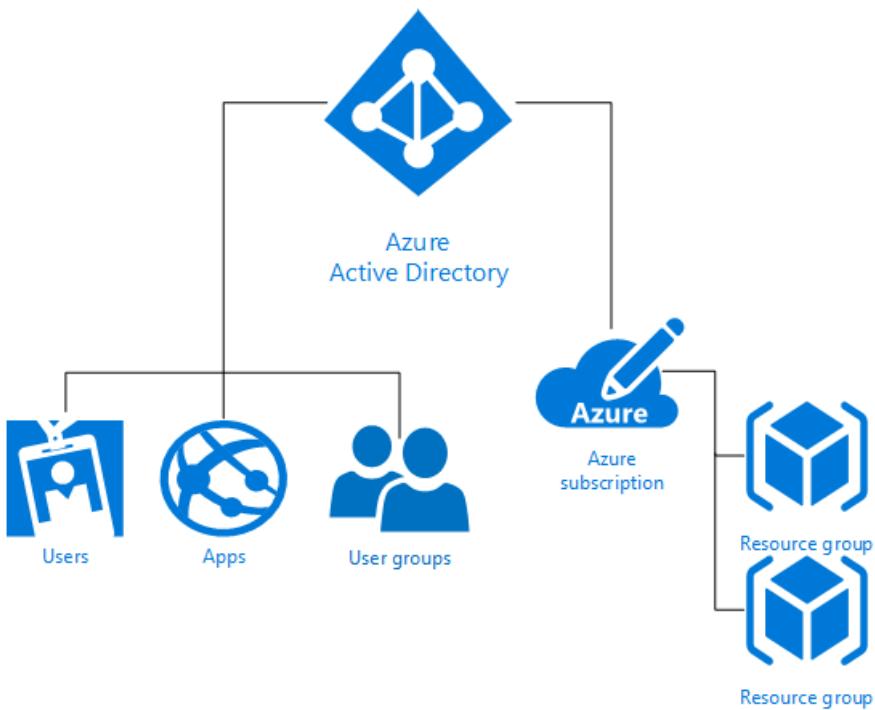
Even when metadata from multiple Azure Active Directory tenants is stored on the same physical disk, there is no relationship between the containers other than what is defined by the directory service, which in turn is dictated by the tenant administrator.

Azure Role-Based Access Control (RBAC)

[Azure Role-Based Access Control \(RBAC\)](#) helps you to share various components available within an Azure subscription by providing fine-grained access management for Azure. Azure RBAC enables you to segregate duties within your organization and grant access based on what users need to perform their jobs. Instead of giving everybody unrestricted permissions in Azure subscription or resources, you can allow only certain actions.

Azure RBAC has three basic roles that apply to all resource types:

- **Owner** has full access to all resources including the right to delegate access to others.
- **Contributor** can create and manage all types of Azure resources but can't grant access to others.
- **Reader** can view existing Azure resources.



The rest of the RBAC roles in Azure allow management of specific Azure resources. For example, the Virtual Machine Contributor role allows the user to create and manage virtual machines. It does not give them access to the Azure Virtual Network or the subnet that the virtual machine connects to.

[RBAC built-in roles](#) list the roles available in Azure. It specifies the operations and scope that each built-in role grants to users. If you're looking to define your own roles for even more control, see how to build [Custom roles in Azure RBAC](#).

Some other capabilities for Azure Active Directory include:

- Azure AD enables SSO to SaaS applications, regardless of where they are hosted. Some applications are federated with Azure AD, and others use password SSO. Federated applications can also support user provisioning and [password vaulting](#).
- Access to data in [Azure Storage](#) is controlled via authentication. Each storage account has a primary key ([storage account key](#), or SAK) and a secondary secret key (the shared access signature, or SAS).
- Azure AD provides Identity as a Service through federation by using [Active Directory Federation Services](#), synchronization, and replication with on-premises directories.
- [Azure Multi-Factor Authentication](#) is the multi-factor authentication service that requires users to verify sign-ins by using a mobile app, phone call, or text message. It can be used with Azure AD to help secure on-premises resources with the Azure Multi-Factor Authentication server, and also with custom applications and directories using the SDK.
- [Azure AD Domain Services](#) lets you join Azure virtual machines to an Active Directory domain without deploying domain controllers. You can sign in to these virtual machines with your corporate Active Directory credentials and administer domain-joined virtual machines by using Group Policy to enforce security baselines on all your Azure virtual machines.
- [Azure Active Directory B2C](#) provides a highly available global-identity management service for consumer-facing applications that scales to hundreds of millions of identities. It can be integrated across mobile and web platforms. Your consumers can sign in to all your applications through customizable experiences by using their existing social accounts or by creating credentials.

Isolation from Microsoft Administrators & Data Deletion

Microsoft takes strong measures to protect your data from inappropriate access or use by unauthorized persons.

These operational processes and controls are backed by the [Online Services Terms](#), which offer contractual commitments that govern access to your data.

- Microsoft engineers do not have default access to your data in the cloud. Instead, they are granted access, under management oversight, only when necessary. That access is carefully controlled and logged, and revoked when it is no longer needed.
- Microsoft may hire other companies to provide limited services on its behalf. Subcontractors may access customer data only to deliver the services for which, we have hired them to provide, and they are prohibited from using it for any other purpose. Further, they are contractually bound to maintain the confidentiality of our customers' information.

Business services with audited certifications such as ISO/IEC 27001 are regularly verified by Microsoft and accredited audit firms, which perform sample audits to attest that access, only for legitimate business purposes. You can always access your own customer data at any time and for any reason.

If you delete any data, Microsoft Azure deletes the data, including any cached or backup copies. For in-scope services, that deletion will occur within 90 days after the end of the retention period. (In-scope services are defined in the Data Processing Terms section of our [Online Services Terms](#).)

If a disk drive used for storage suffers a hardware failure, it is securely [erased or destroyed](#) before Microsoft returns it to the manufacturer for replacement or repair. The data on the drive is overwritten to ensure that the data cannot be recovered by any means.

Compute Isolation

Microsoft Azure provides various cloud-based computing services that include a wide selection of compute instances & services that can scale up and down automatically to meet the needs of your application or enterprise. These compute instance and service offer isolation at multiple levels to secure data without sacrificing the flexibility in configuration that customers demand.

Isolated Virtual Machine Sizes

Azure Compute offers virtual machine sizes that are isolated to a specific hardware type and dedicated to a single customer. These virtual machine sizes are best suited for workloads that require a high degree of isolation from other customers for workloads involving elements like compliance and regulatory requirements. Customers can also choose to further subdivide the resources of these Isolated virtual machines by using [Azure support for nested virtual machines](#).

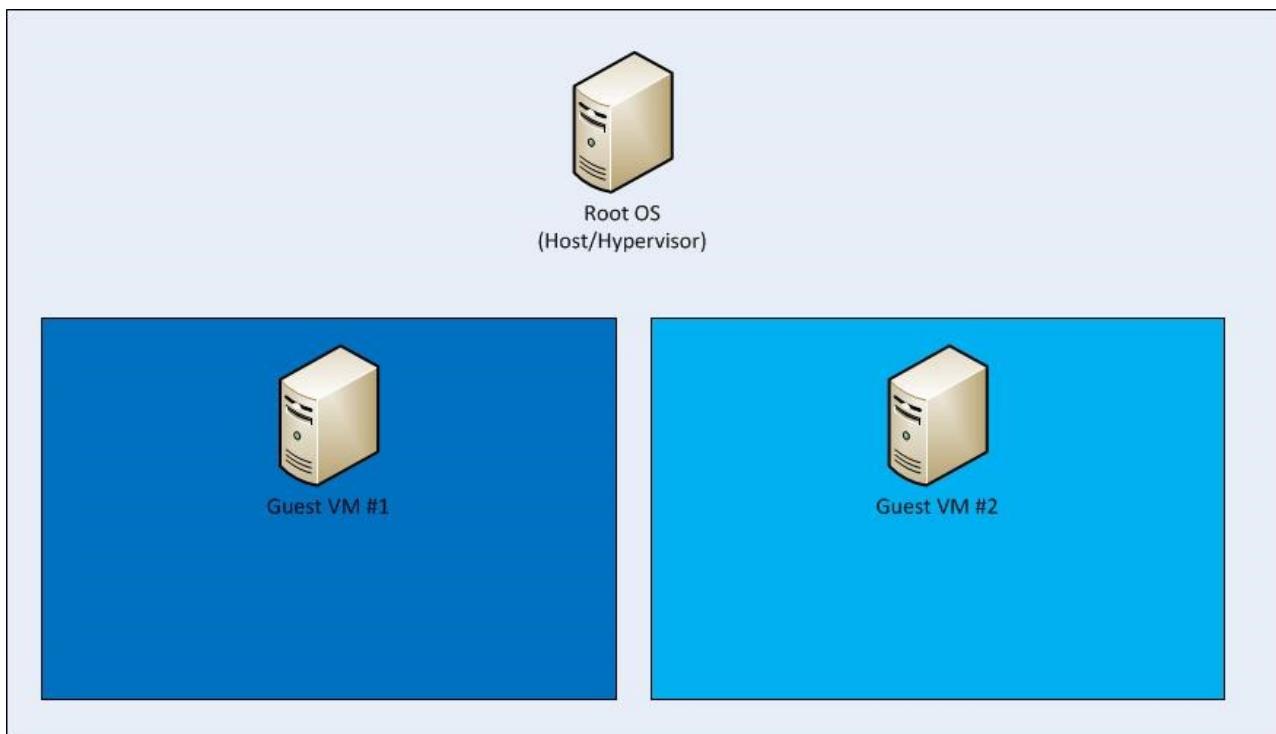
Utilizing an isolated size guarantees that your virtual machine will be the only one running on that specific server instance. The current Isolated virtual machine offerings include:

- Standard_E64is_v3
- Standard_E64i_v3
- Standard_M128ms
- Standard_GS5
- Standard_G5
- Standard_DS15_v2
- Standard_D15_v2

You can learn more about each Isolated size available [here](#).

Hyper-V & Root OS Isolation Between Root VM & Guest VMs

Azure's compute platform is based on machine virtualization—meaning that all customer code executes in a Hyper-V virtual machine. On each Azure node (or network endpoint), there is a Hypervisor that runs directly over the hardware and divides a node into a variable number of Guest Virtual Machines (VMs).



Each node also has one special Root VM, which runs the Host OS. A critical boundary is the isolation of the root VM from the guest VMs and the guest VMs from one another, managed by the hypervisor and the root OS. The hypervisor/root OS pairing leverages Microsoft's decades of operating system security experience, and more recent learning from Microsoft's Hyper-V, to provide strong isolation of guest VMs.

The Azure platform uses a virtualized environment. User instances operate as standalone virtual machines that do not have access to a physical host server.

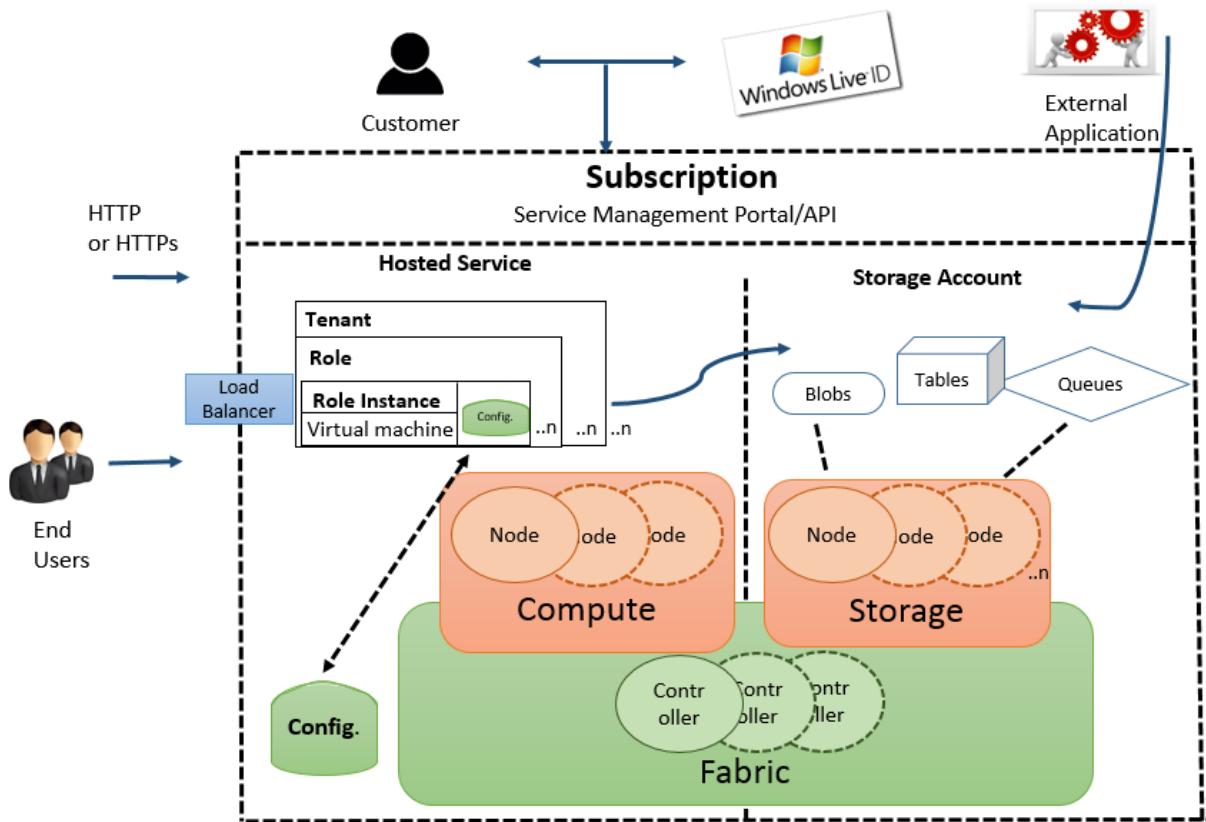
The Azure hypervisor acts like a micro-kernel and passes all hardware access requests from guest virtual machines to the host for processing by using a shared-memory interface called VMBus. This prevents users from obtaining raw read/write/execute access to the system and mitigates the risk of sharing system resources.

Advanced VM placement algorithm & protection from side channel attacks

Any cross-VM attack involves two steps: placing an adversary-controlled VM on the same host as one of the victim VMs, and then breaching the isolation boundary to either steal sensitive victim information or affect its performance for greed or vandalism. Microsoft Azure provides protection at both steps by using an advanced VM placement algorithm and protection from all known side channel attacks including noisy neighbor VMs.

The Azure Fabric Controller

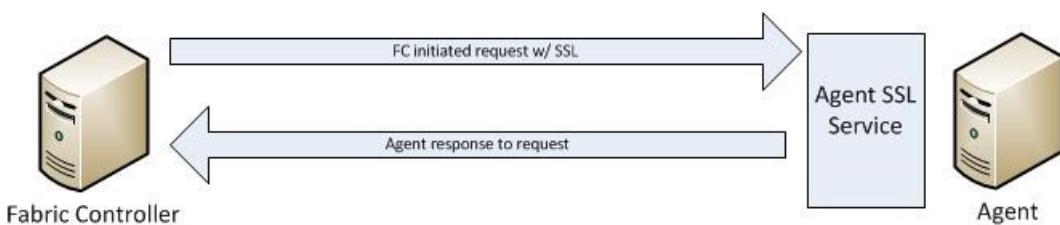
The Azure Fabric Controller is responsible for allocating infrastructure resources to tenant workloads, and it manages unidirectional communications from the host to virtual machines. The VM placing algorithm of the Azure fabric controller is highly sophisticated and nearly impossible to predict at physical host level.



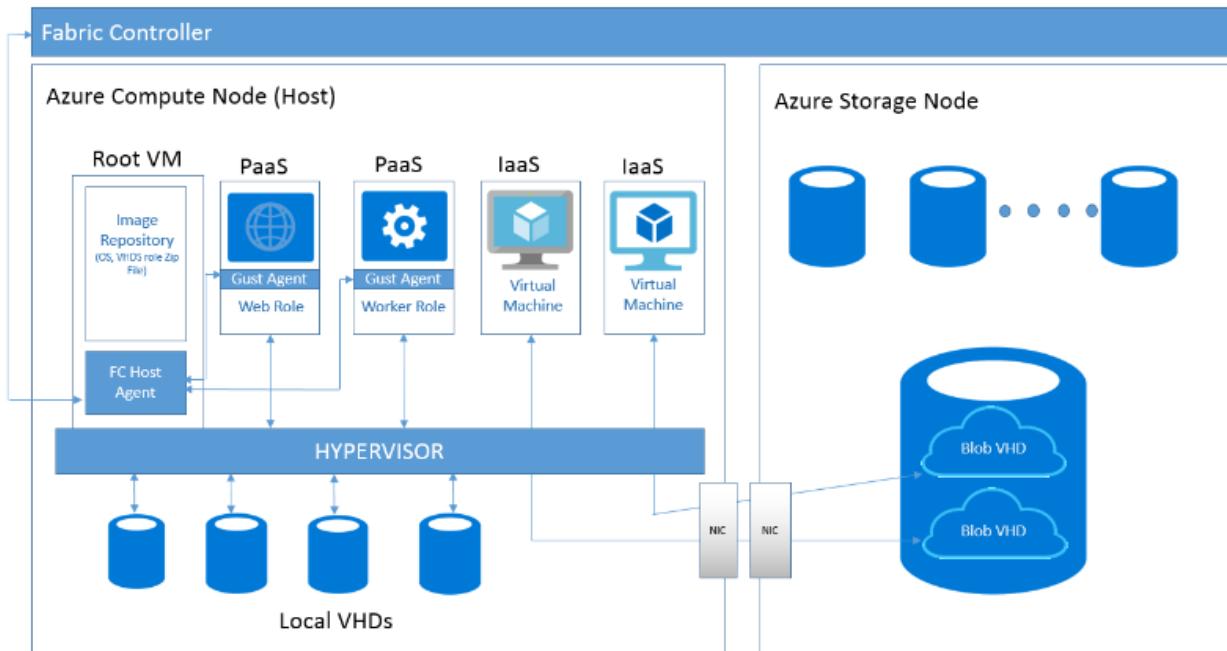
The Azure hypervisor enforces memory and process separation between virtual machines, and it securely routes network traffic to guest OS tenants. This eliminates possibility of and side channel attack at VM level.

In Azure, the root VM is special: it runs a hardened operating system called the root OS that hosts a fabric agent (FA). FAs are used in turn to manage guest agents (GA) within guest OSes on customer VMs. FAs also manage storage nodes.

The collection of Azure hypervisor, root OS/FA, and customer VMs/GAs comprises a compute node. FAs are managed by a fabric controller (FC), which exists outside of compute and storage nodes (compute and storage clusters are managed by separate FCs). If a customer updates their application's configuration file while it's running, the FC communicates with the FA, which then contacts GAs, which notify the application of the configuration change. In the event of a hardware failure, the FC will automatically find available hardware and restart the VM there.



Communication from a Fabric Controller to an agent is unidirectional. The agent implements an SSL-protected service that only responds to requests from the controller. It cannot initiate connections to the controller or other privileged internal nodes. The FC treats all responses as if they were untrusted.



Isolation extends from the Root VM from Guest VMs, and the Guest VMs from one another. Compute nodes are also isolated from storage nodes for increased protection.

The hypervisor and the host OS provide network packet - filters to help assure that untrusted virtual machines cannot generate spoofed traffic or receive traffic not addressed to them, direct traffic to protected infrastructure endpoints, or send/receive inappropriate broadcast traffic.

Additional Rules Configured by Fabric Controller Agent to Isolate VM

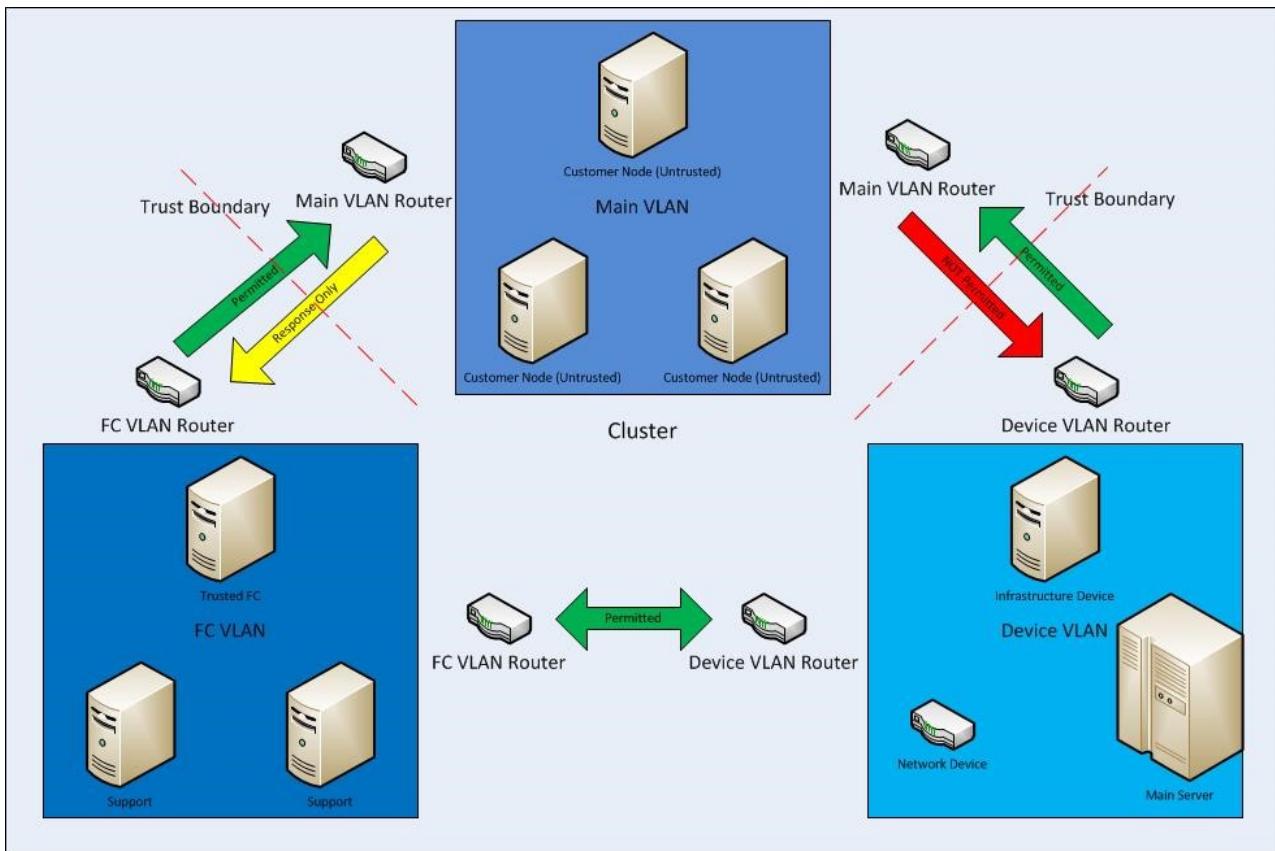
By default, all traffic is blocked when a virtual machine is created, and then the fabric controller agent configures the packet filter to add rules and exceptions to allow authorized traffic.

There are two categories of rules that are programmed:

- **Machine configuration or infrastructure rules:** By default, all communication is blocked. There are exceptions to allow a virtual machine to send and receive DHCP and DNS traffic. Virtual machines can also send traffic to the “public” internet and send traffic to other virtual machines within the same Azure Virtual Network and the OS activation server. The virtual machines’ list of allowed outgoing destinations does not include Azure router subnets, Azure management, and other Microsoft properties.
- **Role configuration file:** This defines the inbound Access Control Lists (ACLs) based on the tenant’s service model.

VLAN Isolation

There are three VLANs in each cluster:



- The main VLAN – interconnects untrusted customer nodes
- The FC VLAN – contains trusted FCs and supporting systems
- The device VLAN – contains trusted network and other infrastructure devices

Communication is permitted from the FC VLAN to the main VLAN, but cannot be initiated from the main VLAN to the FC VLAN. Communication is also blocked from the main VLAN to the device VLAN. This assures that even if a node running customer code is compromised, it cannot attack nodes on either the FC or device VLANs.

Storage Isolation

Logical Isolation Between Compute and Storage

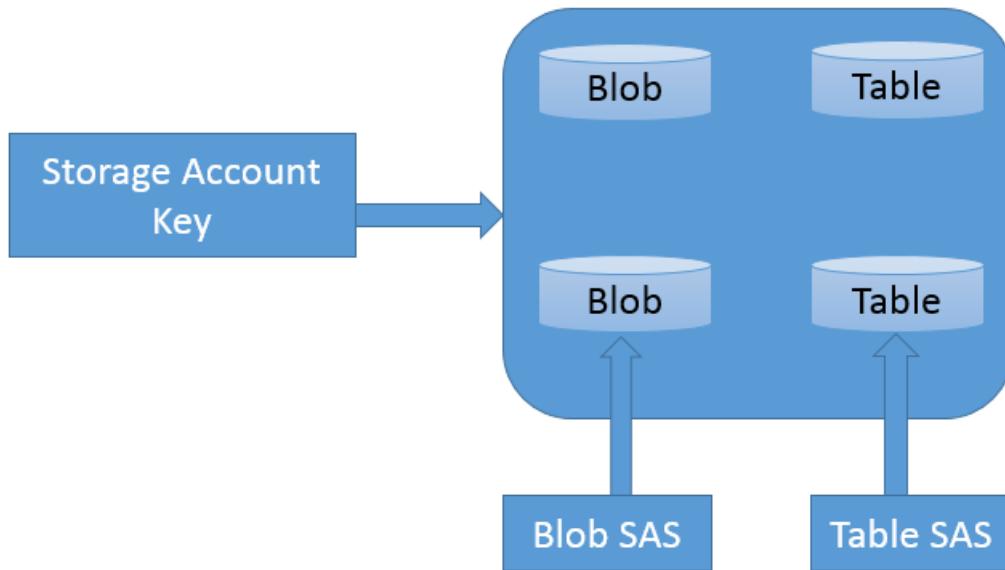
As part of its fundamental design, Microsoft Azure separates VM-based computation from storage. This separation enables computation and storage to scale independently, making it easier to provide multi-tenancy and isolation.

Therefore, Azure Storage runs on separate hardware with no network connectivity to Azure Compute except logically. [This](#) means that when a virtual disk is created, disk space is not allocated for its entire capacity. Instead, a table is created that maps addresses on the virtual disk to areas on the physical disk and that table is initially empty.

The first time a customer writes data on the virtual disk, space on the physical disk is allocated, and a pointer to it is placed in the table.

Isolation Using Storage Access control

Access Control in Azure Storage has a simple access control model. Each Azure subscription can create one or more Storage Accounts. Each Storage Account has a single secret key that is used to control access to all data in that Storage Account.



Access to Azure Storage data (including Tables) can be controlled through a [SAS \(Shared Access Signature\)](#) token, which grants scoped access. The SAS is created through a query template (URL), signed with the [SAK \(Storage Account Key\)](#). That [signed URL](#) can be given to another process (that is, delegated), which can then fill in the details of the query and make the request of the storage service. A SAS enables you to grant time-based access to clients without revealing the storage account's secret key.

The SAS means that we can grant a client limited permissions, to objects in our storage account for a specified period of time and with a specified set of permissions. We can grant these limited permissions without having to share your account access keys.

IP Level Storage Isolation

You can establish firewalls and define an IP address range for your trusted clients. With an IP address range, only clients that have an IP address within the defined range can connect to [Azure Storage](#).

IP storage data can be protected from unauthorized users via a networking mechanism that is used to allocate a dedicated or dedicated tunnel of traffic to IP storage.

Encryption

Azure offers the following types of Encryption to protect data:

- Encryption in transit
- Encryption at rest

Encryption in Transit

Encryption in transit is a mechanism of protecting data when it is transmitted across networks. With Azure Storage, you can secure data using:

- [Transport-level encryption](#), such as HTTPS when you transfer data into or out of Azure Storage.
- [Wire encryption](#), such as SMB 3.0 encryption for Azure File shares.
- [Client-side encryption](#), to encrypt the data before it is transferred into storage and to decrypt the data after it is transferred out of storage.

Encryption at Rest

For many organizations, [data encryption at rest](#) is a mandatory step towards data privacy, compliance, and data sovereignty. There are three Azure features that provide encryption of data that is "at rest":

- [Storage Service Encryption](#) allows you to request that the storage service automatically encrypt data when writing it to Azure Storage.
- [Client-side Encryption](#) also provides the feature of encryption at rest.
- [Azure Disk Encryption](#) allows you to encrypt the OS disks and data disks used by an IaaS virtual machine.

Azure Disk Encryption

[Azure Disk Encryption](#) for virtual machines (VMs) helps you address organizational security and compliance requirements by encrypting your VM disks (including boot and data disks) with keys and policies you control in [Azure Key Vault](#).

The Disk Encryption solution for Windows is based on [Microsoft BitLocker Drive Encryption](#), and the Linux solution is based on [dm-crypt](#).

The solution supports the following scenarios for IaaS VMs when they are enabled in Microsoft Azure:

- Integration with Azure Key Vault
- Standard tier VMs: A, D, DS, G, GS, and so forth, series IaaS VMs
- Enabling encryption on Windows and Linux IaaS VMs
- Disabling encryption on OS and data drives for Windows IaaS VMs
- Disabling encryption on data drives for Linux IaaS VMs
- Enabling encryption on IaaS VMs that are running Windows client OS
- Enabling encryption on volumes with mount paths
- Enabling encryption on Linux VMs that are configured with disk striping (RAID) by using [mdadm](#)
- Enabling encryption on Linux VMs by using [LVM\(Logical Volume Manager\)](#) for data disks
- Enabling encryption on Windows VMs that are configured by using storage spaces
- All Azure public regions are supported

The solution does not support the following scenarios, features, and technology in the release:

- Basic tier IaaS VMs
- Disabling encryption on an OS drive for Linux IaaS VMs
- IaaS VMs that are created by using the classic VM creation method
- Integration with your on-premises Key Management Service
- Azure Files (shared file system), Network File System (NFS), dynamic volumes, and Windows VMs that are configured with software-based RAID systems

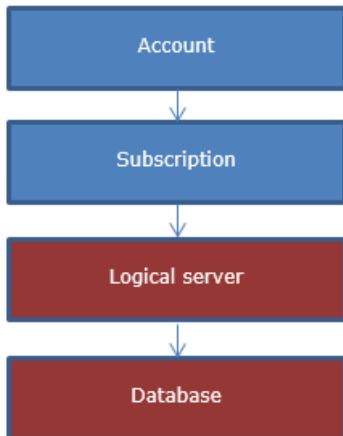
SQL Azure Database Isolation

SQL Database is a relational database service in the Microsoft cloud based on the market-leading Microsoft SQL Server engine and capable of handling mission-critical workloads. SQL Database offers predictable data isolation at account level, geography / region based and based on networking— all with near-zero administration.

SQL Azure Application Model

[Microsoft SQL Azure](#) Database is a cloud-based relational database service built on SQL Server technologies. It provides a highly available, scalable, multi-tenant database service hosted by Microsoft in cloud.

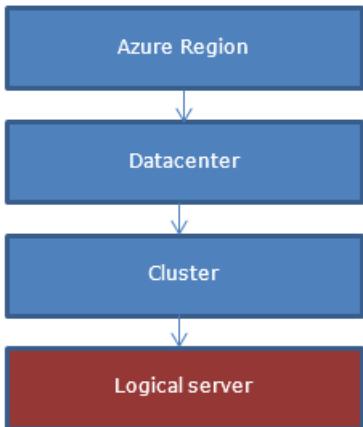
From an application perspective SQL Azure provides the following hierarchy: Each level has one-to-many containment of levels below.



The account and subscription are Microsoft Azure platform concepts to associate billing and management.

Logical servers and databases are SQL Azure-specific concepts and are managed by using SQL Azure, provided OData and TSQL interfaces or via SQL Azure portal that integrated into Azure portal.

SQL Azure servers are not physical or VM instances, instead they are collections of databases, sharing management and security policies, which are stored in so called "logical master" database.



Logical master databases include:

- SQL logins used to connect to the server
- Firewall rules

Billing and usage-related information for SQL Azure databases from the same logical server are not guaranteed to be on the same physical instance in SQL Azure cluster, instead applications must provide the target database name when connecting.

From a customer perspective, a logical server is created in a geo-graphical region while the actual creation of the server happens in one of the clusters in the region.

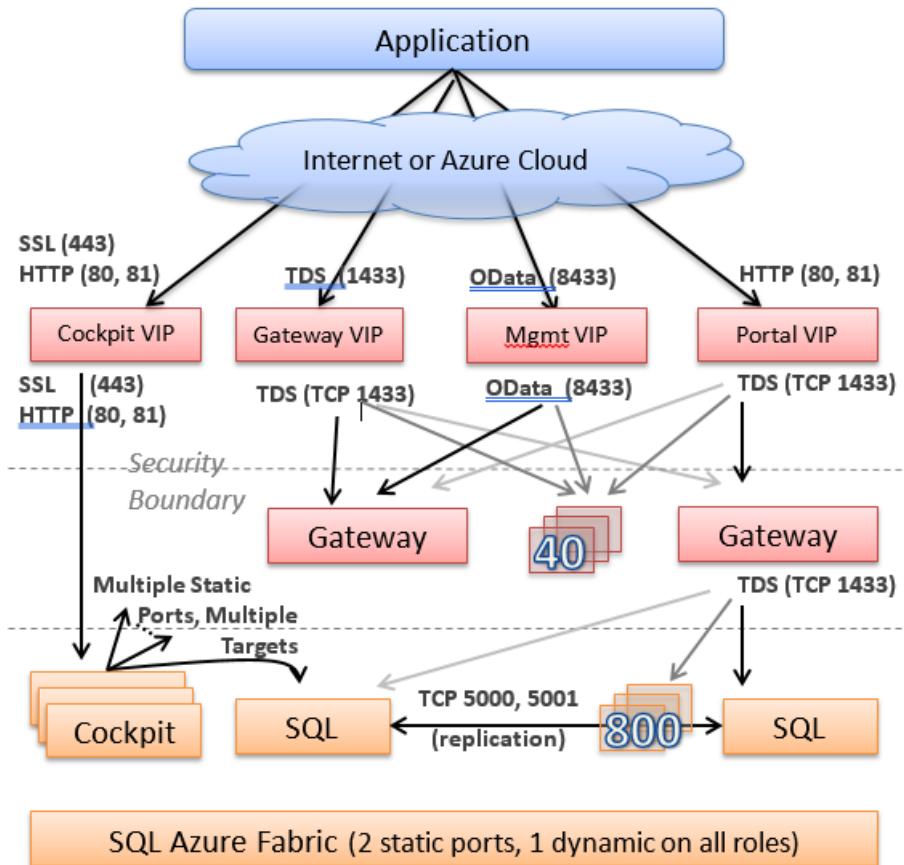
Isolation through Network Topology

When a logical server is created and its DNS name is registered, the DNS name points to the so called "Gateway VIP" address in the specific data center where the server was placed.

Behind the VIP (virtual IP address), we have a collection of stateless gateway services. In general, gateways get

involved when there is coordination needed between multiple data sources (master database, user database, etc.).
 Gateway services implement the following:

- **TDS connection proxying.** This includes locating user database in the backend cluster, implementing the login sequence and then forwarding the TDS packets to the backend and back.
- **Database management.** This includes implementing a collection of workflows to do CREATE/ALTER/DROP database operations. The database operations can be invoked by either sniffing TDS packets or explicit OData APIs.
- CREATE/ALTER/DROP login/user operations
- Logical server management operations via OData API



The tier behind the gateways is called "back-end". This is where all the data is stored in a highly available fashion. Each piece of data is said to belong to a "partition" or "failover unit", each of them having at least three replicas. Replicas are stored and replicated by SQL Server engine and managed by a failover system often referred to as "fabric".

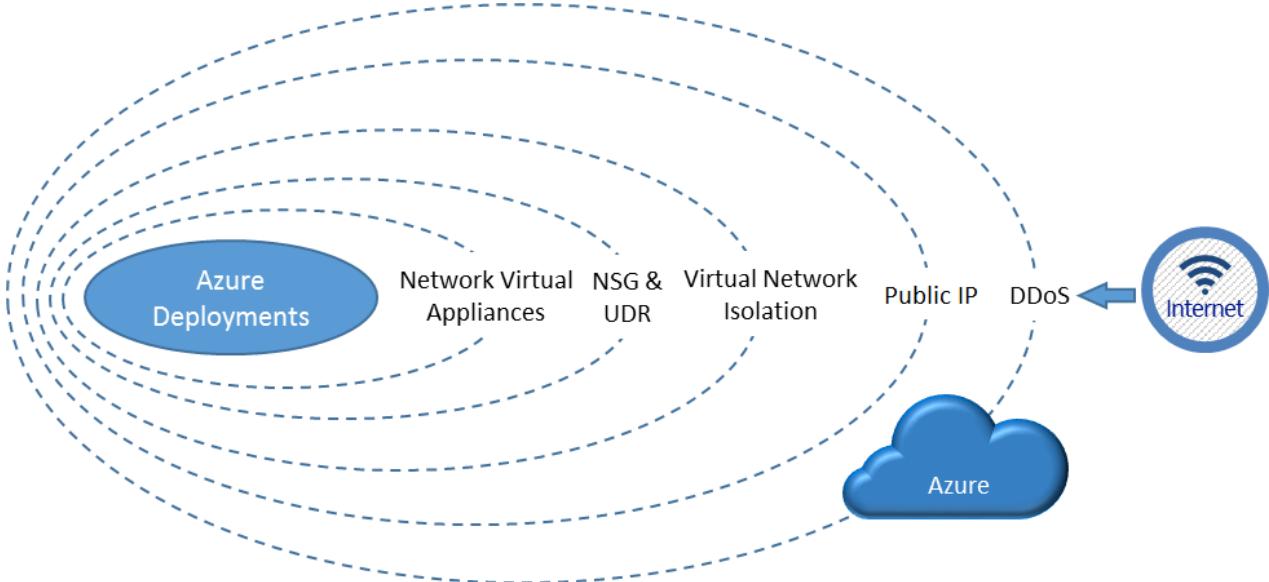
Generally, the back-end system does not communicate outbound to other systems as a security precaution. This is reserved to the systems in the front-end (gateway) tier. The gateway tier machines have limited privileges on the back-end machines to minimize the attack surface as a defense-in-depth mechanism.

Isolation by Machine Function and Access

SQL Azure (is composed of services running on different machine functions. SQL Azure is divided into "backend" Cloud Database and "front-end" (Gateway/Management) environments, with the general principle of traffic only going into back-end and not out. The front-end environment can communicate to the outside world of other services and in general, has only limited permissions in the back-end (enough to call the entry points it needs to invoke).

Networking Isolation

Azure deployment has multiple layers of network isolation. The following diagram shows various layers of network isolation Azure provides to customers. These layers are both native in the Azure platform itself and customer-defined features. Inbound from the Internet, Azure DDoS provides isolation against large-scale attacks against Azure. The next layer of isolation is customer-defined public IP addresses (endpoints), which are used to determine which traffic can pass through the cloud service to the virtual network. Native Azure virtual network isolation ensures complete isolation from all other networks, and that traffic only flows through user configured paths and methods. These paths and methods are the next layer, where NSGs, UDR, and network virtual appliances can be used to create isolation boundaries to protect the application deployments in the protected network.



Traffic isolation: A [virtual network](#) is the traffic isolation boundary on the Azure platform. Virtual machines (VMs) in one virtual network cannot communicate directly to VMs in a different virtual network, even if both virtual networks are created by the same customer. Isolation is a critical property that ensures customer VMs and communication remains private within a virtual network.

[Subnet](#) offers an additional layer of isolation within a virtual network based on IP range. IP addresses in the virtual network, you can divide a virtual network into multiple subnets for organization and security. VMs and PaaS role instances deployed to subnets (same or different) within a VNet can communicate with each other without any extra configuration. You can also configure [network security group \(NSGs\)](#) to allow or deny network traffic to a VM instance based on rules configured in access control list (ACL) of NSG. NSGs can be associated with either subnets or individual VM instances within that subnet. When an NSG is associated with a subnet, the ACL rules apply to all the VM instances in that subnet.

Next Steps

- [Network Isolation Options for Machines in Windows Azure Virtual Networks](#)

This includes the classic front-end and back-end scenario where machines in a particular back-end network or subnetwork may only allow certain clients or other computers to connect to a particular endpoint based on a whitelist of IP addresses.

- [Compute Isolation](#)

Microsoft Azure provides a variety of cloud-based computing services that include a wide selection of compute instances & services that can scale up and down automatically to meet the needs of your application or enterprise.

- [Storage Isolation](#)

Microsoft Azure separates customer VM-based computation from storage. This separation enables computation and storage to scale independently, making it easier to provide multi-tenancy and isolation. Therefore, Azure Storage runs on separate hardware with no network connectivity to Azure Compute except logically. All requests

run over HTTP or HTTPS based on customer's choice.

Azure security technical capabilities

11/7/2018 • 31 minutes to read • [Edit Online](#)

To assist current and prospective Azure customers understand and utilize the various security-related capabilities available in and surrounding the Azure Platform, Microsoft has developed a series of White Papers, Security Overviews, Best Practices, and Checklists. The topics range in terms of breadth and depth and are updated periodically. This document is part of that series as summarized in the Abstract section below. Further information on this Azure Security series can be found at (URL).

Azure platform

Microsoft Azure is a cloud platform comprised of infrastructure and application services, with integrated data services and advanced analytics, and developer tools and services, hosted within Microsoft's public cloud data centers. Customers use Azure for many different capacities and scenarios, from basic compute, networking, and storage, to mobile and web app services, to full cloud scenarios like Internet of Things, and can be used with open source technologies, and deployed as hybrid cloud or hosted within a customer's datacenter. Azure provides cloud technology as building blocks to help companies save costs, innovate quickly, and manage systems proactively. When you build on, or migrate IT assets to a cloud provider, you are relying on that organization's abilities to protect your applications and data with the services and the controls they provide to manage the security of your cloud-based assets.

Microsoft Azure is the only cloud computing provider that offers a secure, consistent application platform and infrastructure-as-a-service for teams to work within their different cloud skillsets and levels of project complexity, with integrated data services and analytics that uncover intelligence from data wherever it exists, across both Microsoft and non-Microsoft platforms, open frameworks and tools, providing choice for integrating cloud with on-premises as well deploying Azure cloud services within on-premises datacenters. As part of the Microsoft Trusted Cloud, customers rely on Azure for industry-leading security, reliability, compliance, privacy, and the vast network of people, partners, and processes to support organizations in the cloud.

With Microsoft Azure, you can:

- Accelerate innovation with the cloud.
- Power business decisions & apps with insights.
- Build freely and deploy anywhere.
- Protect their business.

Scope

The focal point of this whitepaper concerns security features and functionality supporting Microsoft Azure's core components, namely [Microsoft Azure Storage](#), [Microsoft Azure SQL Database](#), [Microsoft Azure's virtual machine model](#), and the tools and infrastructure that manage it all. This white paper focus on Microsoft Azure technical capabilities available to you as customers to fulfil their role in protecting the security and privacy of their data.

The importance of understanding this shared responsibility model is essential for customers who are moving to the cloud. Cloud providers offer considerable advantages for security and compliance efforts, but these advantages do not absolve the customer from protecting their users, applications, and service offerings.

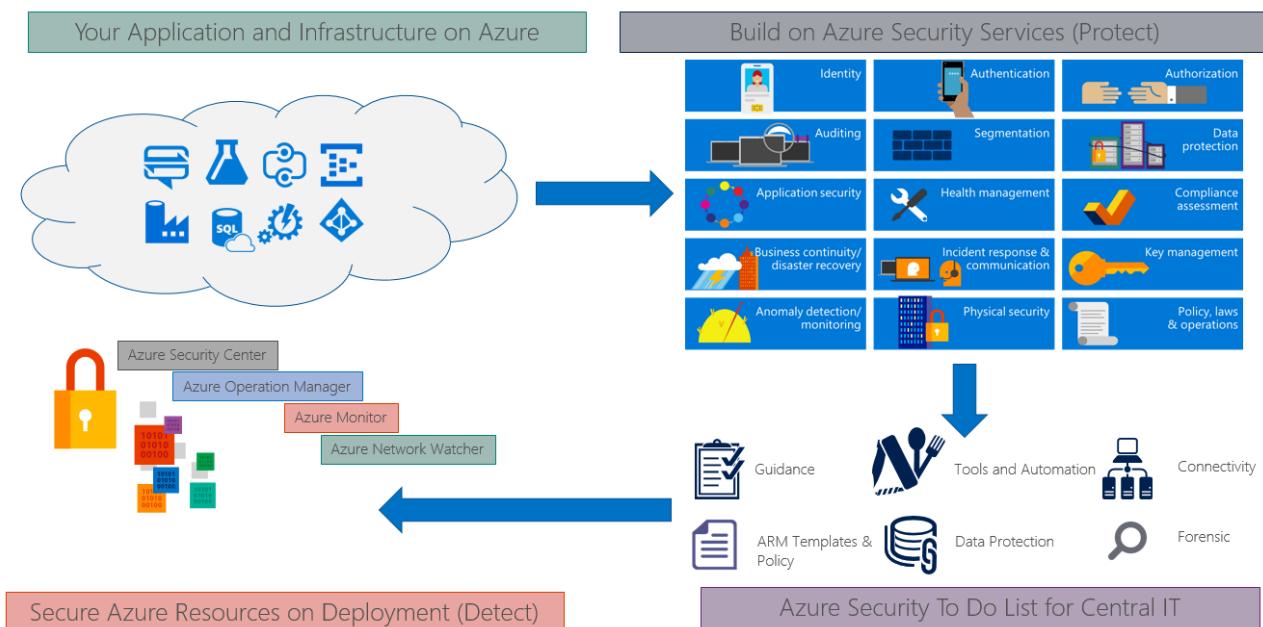
For IaaS solutions, the customer is responsible or has a shared responsibility for securing and managing the operating system, network configuration, applications, identity, clients, and data. PaaS solutions build on IaaS

deployments, the customer is still responsible or has a shared responsibility for securing and managing applications, identity, clients, and data. For SaaS solutions, Nonetheless, the customer continues to be accountable. They must ensure that data is classified correctly, and they share a responsibility to manage their users and end-point devices.

This document does not provide detailed coverage of any of the related Microsoft Azure platform components such as Azure Web Sites, Azure Active Directory, HDInsight, Media Services, and other services that are layered atop the core components. Although a minimum level of general information is provided, readers are assumed familiar with Azure basic concepts as described in other references provided by Microsoft and included in links provided in this white paper.

Available security technical capabilities to fulfil user (Customer) responsibility - Big picture

Microsoft Azure provides services that can help customers meet the security, privacy, and compliance needs. The Following picture helps explain various Azure services available for users to build a secure and compliant application infrastructure based on industry standards.



Manage and control identity and user access (Protect)

Azure helps you protect business and personal information by enabling you to manage user identities and credentials and control access.

Azure Active Directory

Microsoft identity and access management solutions help IT protect access to applications and resources across the corporate datacenter and into the cloud, enabling additional levels of validation such as multi-factor authentication and conditional access policies. Monitoring suspicious activity through advanced security reporting, auditing and alerting helps mitigate potential security issues. [Azure Active Directory Premium](#) provides single sign-on to thousands of cloud (SaaS) apps and access to web apps you run on-premises.

Security benefits of Azure Active Directory (Azure AD) include the ability to:

- Create and manage a single identity for each user across your hybrid enterprise, keeping users, groups, and devices in sync.
- Provide single sign-on access to your applications including thousands of pre-integrated SaaS apps.
- Enable application access security by enforcing rules-based Multi-Factor Authentication for both on-

premises and cloud applications.

- Provision secure remote access to on-premises web applications through Azure AD Application Proxy.

The [Azure Active Directory portal](#) is available a part of the Azure portal. From this dashboard, you can get an overview of the state of your organization, and easily dive into managing the directory, users, or application access.

The screenshot shows the Azure Active Directory admin center dashboard for 'woodgrove WOODGROVEONLINE.COM'. The dashboard includes a company logo, a 'Welcome to the Azure AD admin center' section with a 'Sync with Windows Server AD' button, a 'Users and groups' section showing user profiles, a 'Users Sign-ins' chart from April 16 to May 7, and a 'Quick tasks' sidebar with links like 'Add a user', 'Azure portal', and 'Company branding'.

The following are core Azure Identity management capabilities:

- Single sign-on
- Multi-factor authentication
- Security monitoring, alerts, and machine learning-based reports
- Consumer identity and access management
- Device registration
- Privileged identity management
- Identity protection

Single sign-on

[Single sign-on \(SSO\)](#) means being able to access all the applications and resources that you need to do business, by signing in only once using a single user account. Once signed in, you can access all the applications you need without being required to authenticate (for example, type a password) a second time.

Many organizations rely upon software as a service (SaaS) applications such as Office 365, Box and Salesforce for end-user productivity. Historically, IT staff needed to individually create and update user accounts in each SaaS application, and users had to remember a password for each SaaS application.

Azure AD extends on-premises Active Directory into the cloud, enabling users to use their primary organizational account to not only sign in to their domain-joined devices and company resources, but also all the web and SaaS applications needed for their job.

Not only do users not have to manage multiple sets of usernames and passwords, application access can be automatically provisioned or de-provisioned based on organizational groups and their status as an employee.

Azure AD introduces [security and access governance controls](#) that enable you to centrally manage users' access across SaaS applications.

Multi-factor authentication

Azure Multi-factor Authentication (MFA) is a method of authentication that requires the use of more than one verification method and adds a critical second layer of security to user sign-ins and transactions. [MFA helps safeguard](#) access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication via a range of verification options—phone call, text message, or mobile app notification or verification code and third-party OAuth tokens.

Security monitoring, alerts, and machine learning-based reports

Security monitoring and alerts and machine learning-based reports that identify inconsistent access patterns can help you protect your business. You can use Azure Active Directory's access and usage reports to gain visibility into the integrity and security of your organization's directory. With this information, a directory admin can better determine where possible security risks may lie so that they can adequately plan to mitigate those risks.

In the Azure portal or through the [Azure Active Directory portal](#), [reports](#) are categorized in the following ways:

- Anomaly reports – contain sign in events that we found to be anomalous. Our goal is to make you aware of such activity and enable you to be able to decide about whether an event is suspicious.
- Integrated Application reports – provide insights into how cloud applications are being used in your organization. Azure Active Directory offers integration with thousands of cloud applications.
- Error reports – indicate errors that may occur when provisioning accounts to external applications.
- User-specific reports – display device/sign in activity data for a specific user.
- Activity logs – contain a record of all audited events within the last 24 hours, last 7 days, or last 30 days, and group activity changes, and password reset and registration activity.

Consumer identity and access management

Azure Active Directory B2C is a highly available, global, identity management service for consumer-facing applications that scales to hundreds of millions of identities. It can be integrated across mobile and web platforms. Your consumers can log on to all your applications through customizable experiences by using their existing social accounts or by creating new credentials.

In the past, application developers who wanted to [sign up and sign in consumers](#) into their applications would have written their own code. And they would have used on-premises databases or systems to store usernames and passwords. Azure Active Directory B2C offers your organization a better way to integrate consumer identity management into applications with the help of a secure, standards-based platform, and a large set of extensible policies.

When you use Azure Active Directory B2C, your consumers can sign up for your applications by using their existing social accounts (Facebook, Google, Amazon, LinkedIn) or by creating new credentials (email address and password, or username and password).

Device registration

Azure AD device registration is the foundation for device-based [conditional access](#) scenarios. When a device is registered, Azure AD device registration provides the device with an identity that is used to authenticate the device when the user signs in. The authenticated device, and the attributes of the device, can then be used to enforce conditional access policies for applications that are hosted in the cloud and on-premises.

When combined with a [mobile device management \(MDM\)](#) solution such as Intune, the device attributes in Azure Active Directory are updated with additional information about the device. This allows you to create conditional access rules that enforce access from devices to meet your standards for security and compliance.

Privileged identity management

Azure Active Directory (AD) Privileged Identity Management lets you manage, control, and monitor your

privileged identities and access to resources in Azure AD as well as other Microsoft online services like Office 365 or Microsoft Intune.

Sometimes users need to carry out privileged operations in Azure or Office 365 resources, or other SaaS apps. This often means organizations have to give them permanent privileged access in Azure AD. This is a growing security risk for cloud-hosted resources because organizations can't sufficiently monitor what those users are doing with their admin privileges. Additionally, if a user account with privileged access is compromised, that one breach could impact their overall cloud security. Azure AD Privileged Identity Management helps to resolve this risk.

Azure AD Privileged Identity Management lets you:

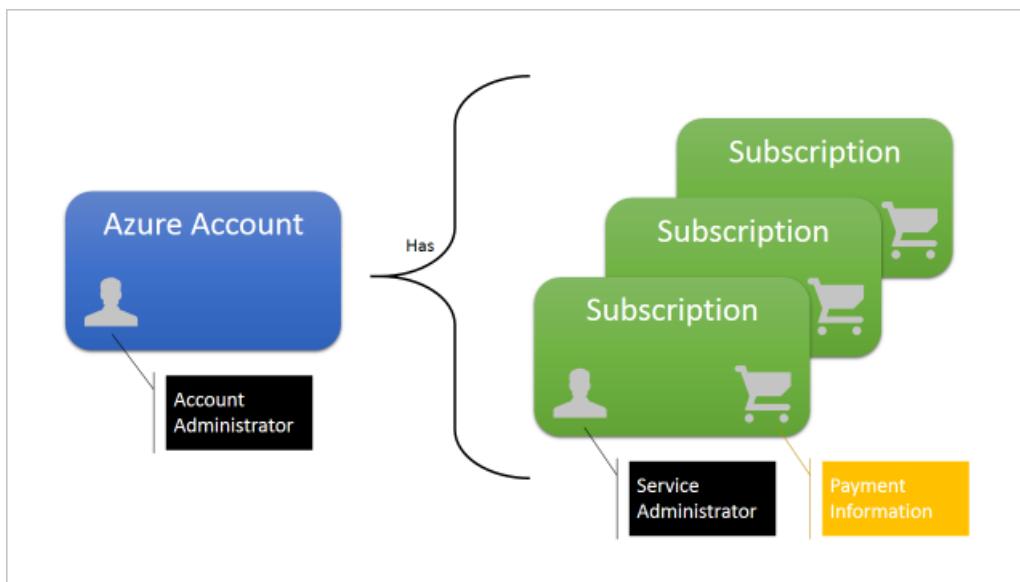
- See which users are Azure AD admins
- Enable on-demand, "just in time" administrative access to Microsoft Online Services like Office 365 and Intune
- Get reports about administrator access history and changes in administrator assignments
- Get alerts about access to a privileged role

Identity protection

[Azure AD Identity Protection](#) is a security service that provides a consolidated view into risk events and potential vulnerabilities affecting your organization's identities. Identity Protection uses existing Azure Active Directory's anomaly detection capabilities (available through Azure AD's Anomalous Activity Reports), and introduces new risk event types that can detect anomalies in real-time.

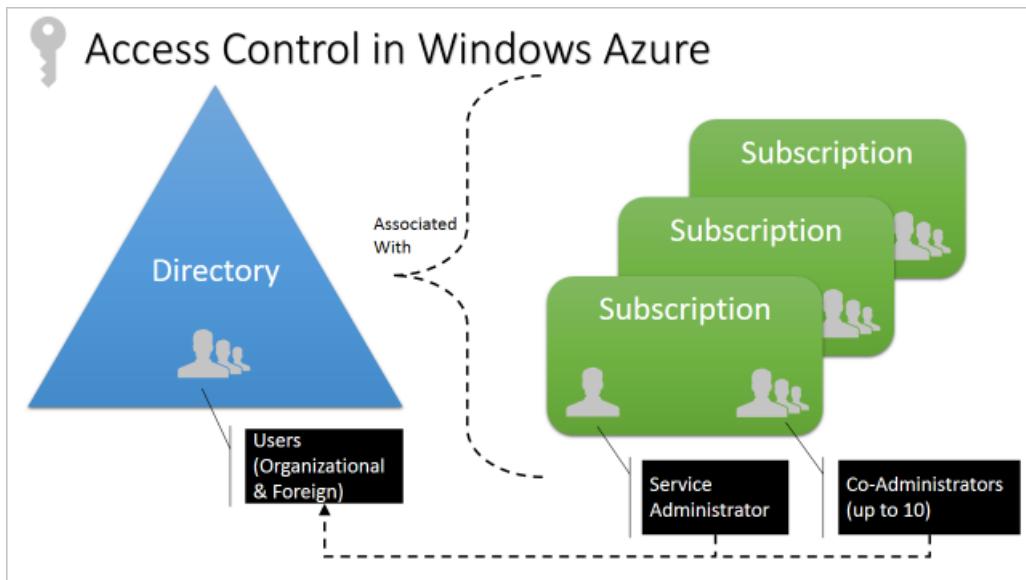
Secured resource access in Azure

Access control in Azure starts from a billing perspective. The owner of an Azure account, accessed by visiting the [Azure Account Center](#), is the Account Administrator (AA). Subscriptions are a container for billing, but they also act as a security boundary: each subscription has a Service Administrator (SA) who can add, remove, and modify Azure resources in that subscription by using the Azure portal. The default SA of a new subscription is the AA, but the AA can change the SA in the Azure Account Center.

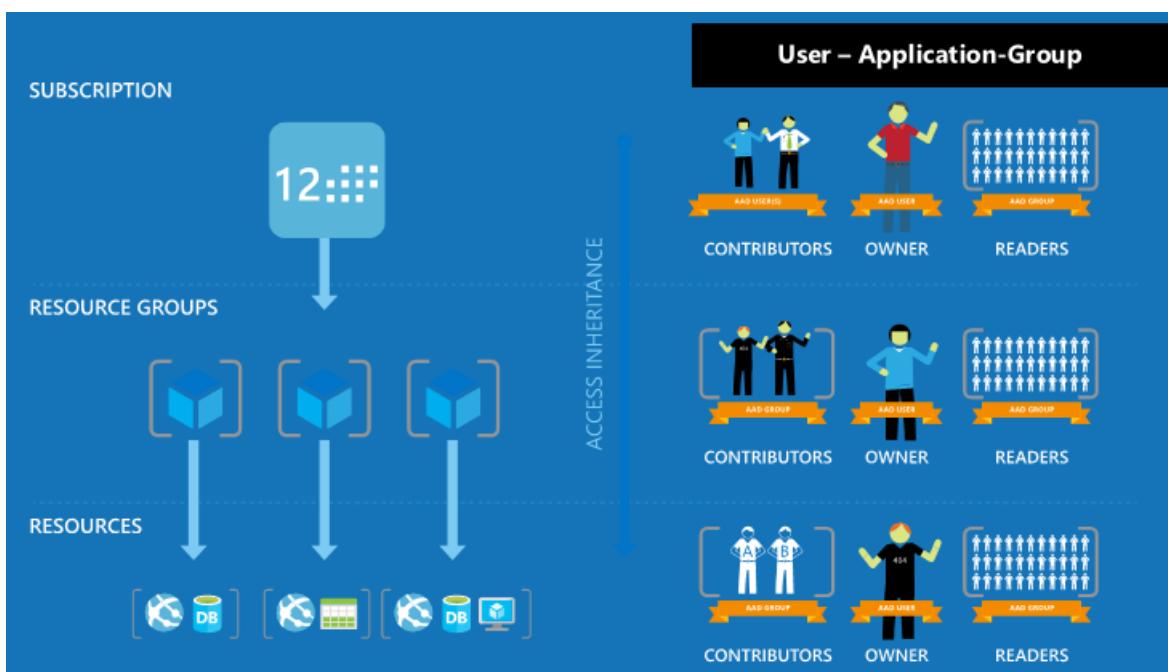


Subscriptions also have an association with a directory. The directory defines a set of users. These can be users from the work or school that created the directory, or they can be external users (that is, Microsoft Accounts). Subscriptions are accessible by a subset of those directory users who have been assigned as either Service Administrator (SA) or Co-Administrator (CA); the only exception is that, for legacy reasons, Microsoft Accounts (formerly Windows Live ID) can be assigned as SA or CA without being present in the directory.

Security-oriented companies should focus on giving employees the exact permissions they need. Too many permissions can expose an account to attackers. Too few permissions mean that employees can't get their work done efficiently. [Azure Role-Based Access Control \(RBAC\)](#) helps address this problem by offering fine-grained access management for Azure.



Using RBAC, you can segregate duties within your team and grant only the amount of access to users that they need to perform their jobs. Instead of giving everybody unrestricted permissions in your Azure subscription or resources, you can allow only certain actions. For example, use RBAC to let one employee manage virtual machines in a subscription, while another can manage SQL databases within the same subscription.



Azure data security and encryption (protect)

One of the keys to data protection in the cloud is accounting for the possible states in which your data may occur, and what controls are available for that state. For Azure data security and encryption best practices the recommendations be around the following data's states.

- At-rest: This includes all information storage objects, containers, and types that exist statically on physical media, be it magnetic or optical disk.
- In-Transit: When data is being transferred between components, locations or programs, such as over the network, across a service bus (from on-premises to cloud and vice-versa, including hybrid connections such

as ExpressRoute), or during an input/output process, it is thought of as being in-motion.

Encryption at rest

To achieve encryption at rest, do each of the following:

Support at least one of the recommended encryption models detailed in the following table to encrypt data.

ENCRYPTION MODELS			
Server Encryption	Server Encryption	Server Encryption	Client Encryption
Server-Side Encryption using Service Managed Keys	Server-side encryption using Customer-Managed Keys in Azure Key Vault	Server-side encryption using on-prem customer managed keys	
<ul style="list-style-type: none">• Azure Resource Providers perform the encryption and decryption operations• Microsoft manages the keys• Full cloud functionality	<ul style="list-style-type: none">• Azure Resource Providers perform the encryption and decryption operations• Customer controls keys via Azure Key Vault• Full cloud functionality	<ul style="list-style-type: none">• Azure Resource Providers perform the encryption and decryption operations• Customer controls keys On-Prem• Full cloud functionality	<ul style="list-style-type: none">• Azure services cannot see decrypted data• Customers keep keys on-premises (or in other secure stores). Keys are not available to Azure services• Reduced cloud functionality

Enabling encryption at rest

Identify All Locations Your Stores Data

The goal of Encryption at Rest is to encrypt all data. Doing so eliminates the possibility of missing important data or all persisted locations. Enumerate all data stored by your application.

NOTE

Not just "application data" or "PII" but any data relating to application including account metadata (subscription mappings, contract info, PII).

Consider what stores you are using to store data. For example:

- External storage (for example, SQL Azure, Document DB, HDInsights, Data Lake, etc.)
- Temporary storage (any local cache that includes tenant data)
- In-memory cache (could be put into the page file.)

Leverage the existing encryption at rest support in Azure

For each store you use, leverage the existing Encryption at Rest support.

- Azure Storage: See [Azure Storage Service Encryption for Data at Rest](#),
- SQL Azure: See [Transparent Data Encryption \(TDE\)](#), [SQL Always Encrypted](#)
- VM & Local disk storage ([Azure Disk Encryption](#))

For VM and Local disk storage use Azure Disk Encryption where supported:

IaaS

Services with IaaS VMs (Windows or Linux) should use [Azure Disk Encryption](#) to encrypt volumes containing customer data.

PaaS v2

Services running on PaaS v2 using Service Fabric can use Azure disk encryption for Virtual Machine Scale Set

[VMSS] to encrypt their PaaS v2 VMs.

PaaS v1

Azure Disk Encryption currently is not supported on PaaS v1. Therefore, you must use application level encryption to encrypt persisted data at rest. This includes, but is not limited to, application data, temporary files, logs, and crash dumps.

Most services should attempt to leverage the encryption of a storage resource provider. Some services have to do explicit encryption, for example, any persisted key material (Certificates, root / master keys) must be stored in Key Vault.

If you support service-side encryption with customer-managed keys there needs to be a way for the customer to get the key to us. The supported and recommended way to do that by integrating with Azure Key Vault (AKV). In this case customers can add and manage their keys in Azure Key Vault. A customer can learn how to use AKV via [Getting Started with Key Vault](#).

To integrate with Azure Key Vault, you'd add code to request a key from AKV when needed for decryption.

- See [Azure Key Vault – Step by Step](#) for info on how to integrate with AKV.

If you support customer managed keys, you need to provide a UX for the customer to specify which Key Vault (or Key Vault URI) to use.

As Encryption at Rest involves the encryption of host, infrastructure and tenant data, the loss of the keys due to system failure or malicious activity could mean all the encrypted data is lost. It is therefore critical that your Encryption at Rest solution has a comprehensive disaster recovery story resilient to system failures and malicious activity.

Services that implement Encryption at Rest are usually still susceptible to the encryption keys or data being left unencrypted on the host drive (for example, in the page file of the host OS.) Therefore, services must ensure the host volume for their services is encrypted. To facilitate this Compute team has enabled the deployment of Host Encryption, which uses [BitLocker](#) NKP and extensions to the DCM service and agent to encrypt the host volume.

Most services are implemented on standard Azure VMs. Such services should get [Host Encryption](#) automatically when Compute enables it. For services running in Compute managed clusters host encryption is enabled automatically as Windows Server 2016 is rolled out.

Encryption in-transit

Protecting data in transit should be essential part of your data protection strategy. Since data is moving back and forth from many locations, the general recommendation is that you always use SSL/TLS protocols to exchange data across different locations. In some circumstances, you may want to isolate the entire communication channel between your on-premises and cloud infrastructure by using a virtual private network (VPN).

For data moving between your on-premises infrastructure and Azure, you should consider appropriate safeguards such as HTTPS or VPN.

For organizations that need to secure access from multiple workstations located on-premises to Azure, use [Azure site-to-site VPN](#).

For organizations that need to secure access from one workstation located on-premises to Azure, use [Point-to-Site VPN](#).

Larger data sets can be moved over a dedicated high-speed WAN link such as [ExpressRoute](#). If you choose to use ExpressRoute, you can also encrypt the data at the application-level using [SSL/TLS](#) or other protocols for added protection.

If you are interacting with Azure Storage through the Azure Portal, all transactions occur via HTTPS. [Storage REST API](#) over HTTPS can also be used to interact with [Azure Storage](#) and [Azure SQL Database](#).

Organizations that fail to protect data in transit are more susceptible for [man-in-the-middle attacks](#), [eavesdropping](#), and session hijacking. These attacks can be the first step in gaining access to confidential data.

You can learn more about Azure VPN option by reading the article [Planning and design for VPN Gateway](#).

Enforce file level data encryption

[Azure RMS](#) uses encryption, identity, and authorization policies to help secure your files and email. Azure RMS works across multiple devices — phones, tablets, and PCs by protecting both within your organization and outside your organization. This capability is possible because Azure RMS adds a level of protection that remains with the data, even when it leaves your organization's boundaries.

When you use Azure RMS to protect your files, you are using industry-standard cryptography with full support of [FIPS 140-2](#). When you leverage Azure RMS for data protection, you have the assurance that the protection stays with the file, even if it is copied to storage that is not under the control of IT, such as a cloud storage service. The same occurs for files shared via e-mail, the file is protected as an attachment to an email message, with instructions how to open the protected attachment. When planning for Azure RMS adoption we recommend the following:

- Install the [RMS sharing app](#). This app integrates with Office applications by installing an Office add-in so that users can easily protect files directly.
- Configure applications and services to support Azure RMS
- Create [custom templates](#) that reflect your business requirements. For example: a template for top secret data that should be applied in all top secret related emails.

Organizations that are weak on [data classification](#) and file protection may be more susceptible to data leakage. Without proper file protection, organizations won't be able to obtain business insights, monitor for abuse and prevent malicious access to files.

NOTE

You can learn more about Azure RMS by reading the article [Getting Started with Azure Rights Management](#).

Secure your application (protect)

While Azure is responsible for securing the infrastructure and platform that your application runs on, it is your responsibility to secure your application itself. In other words, you need to develop, deploy, and manage your application code and content in a secure way. Without this, your application code or content can still be vulnerable to threats.

Web application firewall (WAF)

[Web application firewall \(WAF\)](#) is a feature of [Application Gateway](#) that provides centralized protection of your web applications from common exploits and vulnerabilities.

Web application firewall is based on rules from the [OWASP core rule sets](#) 3.0 or 2.2.9. Web applications are increasingly targets of malicious attacks that exploit common known vulnerabilities. Common among these exploits are SQL injection attacks, cross site scripting attacks to name a few. Preventing such attacks in application code can be challenging and may require rigorous maintenance, patching and monitoring at multiple layers of the application topology. A centralized web application firewall helps make security management much simpler and gives better assurance to application administrators against threats or intrusions. A WAF solution can also react to a security threat faster by patching a known vulnerability at a central location versus securing each of individual web applications. Existing application gateways can be converted to a web application firewall enabled application gateway easily.

Some of the common web vulnerabilities which web application firewall protects against includes:

- SQL injection protection
- Cross site scripting protection
- Common Web Attacks Protection such as command injection, HTTP request smuggling, HTTP response splitting, and remote file inclusion attack
- Protection against HTTP protocol violations
- Protection against HTTP protocol anomalies such as missing host user-agent and accept headers
- Prevention against bots, crawlers, and scanners
- Detection of common application misconfigurations (that is, Apache, IIS, etc.)

NOTE

For a more detailed list of rules and their protections see the following [Core rule sets](#):

Azure also provides several easy-to-use features to help secure both inbound and outbound traffic for your app. Azure also helps customers secure their application code by providing externally provided functionality to scan your web application for vulnerabilities.

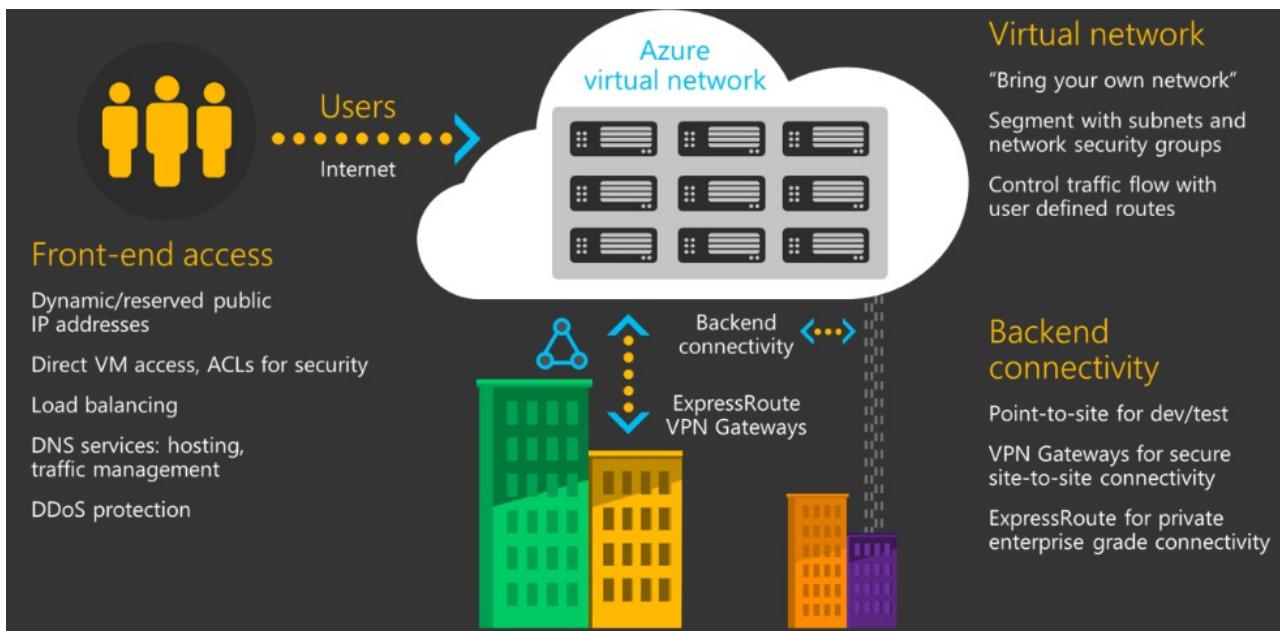
- [Setup Azure Active Directory authentication for your app](#)
- [Secure traffic to your app by enabling Transport Layer Security \(TLS/SSL\) - HTTPS](#)
 - Force all incoming traffic over HTTPS connection
 - Enable Strict Transport Security (HSTS)
- [Restrict access to your app by client's IP address](#)
- [Restrict access to your app by client's behavior - request frequency and concurrency](#)
- [Scan your web app code for vulnerabilities using Tinfoil Security Scanning](#)
- [Configure TLS mutual authentication to require client certificates to connect to your web app](#)
- [Configure a client certificate for use from your app to securely connect to external resources](#)
- [Remove standard server headers to avoid tools from fingerprinting your app](#)
- [Securely connect your app with resources in a private network using Point-To-Site VPN](#)
- [Securely connect your app with resources in a private network using Hybrid Connections](#)

Azure App Service uses the same Antimalware solution used by Azure Cloud Services and Virtual Machines. To learn more about this refer to our [Antimalware documentation](#).

Secure your network (protect)

Microsoft Azure includes a robust networking infrastructure to support your application and service connectivity requirements. Network connectivity is possible between resources located in Azure, between on-premises and Azure hosted resources, and to and from the Internet and Azure.

The [Azure network infrastructure](#) enables you to securely connect Azure resources to each other with [virtual networks \(VNets\)](#). A VNet is a representation of your own network in the cloud. A VNet is a logical isolation of the Azure cloud network dedicated to your subscription. You can connect VNets to your on-premises networks.



If you need basic network level access control (based on IP address and the TCP or UDP protocols), then you can use [Network Security Groups](#). A Network Security Group (NSG) is a basic stateful packet filtering firewall and it enables you to control access based on a [5-tuple](#).

Azure networking supports the ability to customize the routing behavior for network traffic on your Azure Virtual Networks. You can do this by configuring [User-Defined Routes](#) in Azure.

[Forced tunneling](#) is a mechanism you can use to ensure that your services are not allowed to initiate a connection to devices on the Internet.

Azure supports dedicated WAN link connectivity to your on-premises network and an Azure Virtual Network with [ExpressRoute](#). The link between Azure and your site uses a dedicated connection that does not go over the public Internet. If your Azure application is running in multiple datacenters, you can use [Azure Traffic Manager](#) to route requests from users intelligently across instances of the application. You can also route traffic to services not running in Azure if they are accessible from the Internet.

Virtual machine security (protect)

[Azure Virtual Machines](#) lets you deploy a wide range of computing solutions in an agile way. With support for Microsoft Windows, Linux, Microsoft SQL Server, Oracle, IBM, SAP, and Azure BizTalk Services, you can deploy any workload and any language on nearly any operating system.

With Azure, you can use [antimalware software](#) from security vendors such as Microsoft, Symantec, Trend Micro, and Kaspersky to protect your virtual machines from malicious files, adware, and other threats.

Microsoft Antimalware for Azure Cloud Services and Virtual Machines is a real-time protection capability that helps identify and remove viruses, spyware, and other malicious software. Microsoft Antimalware provides configurable alerts when known malicious or unwanted software attempts to install itself or run on your Azure systems.

[Azure Backup](#) is a scalable solution that protects your application data with zero capital investment and minimal operating costs. Application errors can corrupt your data, and human errors can introduce bugs into your applications. With Azure Backup, your virtual machines running Windows and Linux are protected.

[Azure Site Recovery](#) helps orchestrate replication, failover, and recovery of workloads and apps so that they are available from a secondary location if your primary location goes down.

Ensure compliance: Cloud services due diligence checklist (protect)

Microsoft developed [the Cloud Services Due Diligence Checklist](#) to help organizations exercise due diligence as they consider a move to the cloud. It provides a structure for an organization of any size and type—private businesses and public-sector organizations, including government at all levels and nonprofits—to identify their own performance, service, data management, and governance objectives and requirements. This allows them to compare the offerings of different cloud service providers, ultimately forming the basis for a cloud service agreement.

The checklist provides a framework that aligns clause-by-clause with a new international standard for cloud service agreements, ISO/IEC 19086. This standard offers a unified set of considerations for organizations to help them make decisions about cloud adoption, and create a common ground for comparing cloud service offerings.

The checklist promotes a thoroughly vetted move to the cloud, providing structured guidance and a consistent, repeatable approach for choosing a cloud service provider.

Cloud adoption is no longer simply a technology decision. Because checklist requirements touch on every aspect of an organization, they serve to convene all key internal decision-makers—the CIO and CISO as well as legal, risk management, procurement, and compliance professionals. This increases the efficiency of the decision-making process and ground decisions in sound reasoning, thereby reducing the likelihood of unforeseen roadblocks to adoption.

In addition, the checklist:

- Exposes key discussion topics for decision-makers at the beginning of the cloud adoption process.
- Supports thorough business discussions about regulations and the organization's own objectives for privacy, personally identifiable information (PII), and data security.
- Helps organizations identify any potential issues that could affect a cloud project.
- Provides a consistent set of questions, with the same terms, definitions, metrics, and deliverables for each provider, to simplify the process of comparing offerings from different cloud service providers.

Azure infrastructure and application security validation (detect)

[Azure Operational Security](#) refers to the services, controls, and features available to users for protecting their data, applications, and other assets in Microsoft Azure.

Insight & Analytics

- Quickly diagnose root cause across the full stack of modern applications and underlying infrastructure
- Monitor and alert on key metrics and KPIs in real time to rapidly identify problems
- Collect, process and analyze petabytes of data
- Create and share data insights across your company in minutes
- Integrate with and extend the value of existing monitoring tools

Protection & Recovery

- Protection of Cloud Assets (DR/Backup for IAAS, Backup of SQL PaaS)
- Enhanced Capacity Planning and Monitoring with Log Analytics
- Enterprise coverage with Linux distros, SQL AG, Encryption at rest
- Faster, Cheaper, Compact Backup Storage ([Xcool](#), De-dup, [ReFS](#))
- Centralized hybrid backup monitoring and reporting in Azure
- Workload protection for public, hybrid, and private cloud
- Enterprise grade VMware VM Backup

Automation & Control

- Trigger immediate action in response to issues automatically or on-demand
- Maintain the state of IT resources and resolve configuration drifts
- Keep IT systems updated with minimal downtime
- Track and manage changes with ease

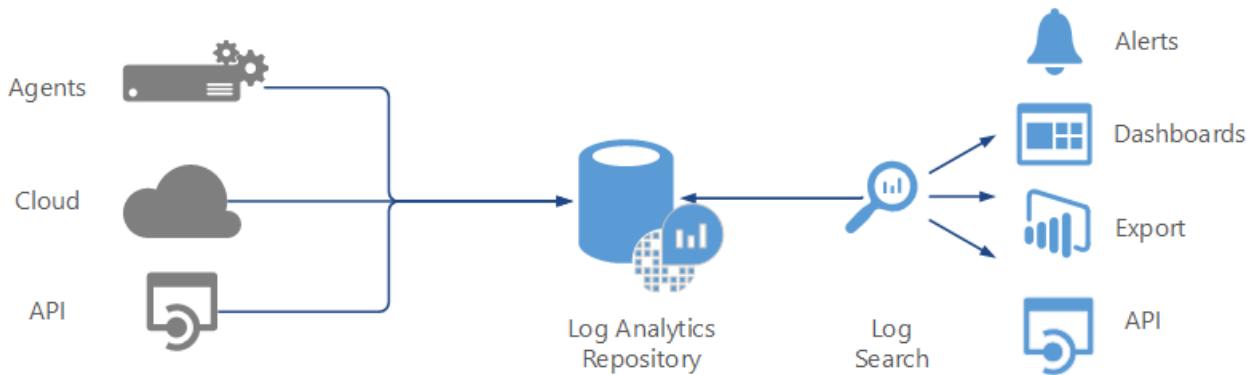
Security & Compliance

- Collection of security data from virtually any source
- Insight into security status (antimalware, system updates)
- Correlations to detect malicious activities and search for rapid investigation
- Integrates operational and security management
- Threat detection using advanced analytics

Azure Operational Security is built on a framework that incorporates the knowledge gained through a various capabilities that are unique to Microsoft, including the Microsoft Security Development Lifecycle (SDL), the Microsoft Security Response Centre program, and deep awareness of the cybersecurity threat landscape.

Microsoft Azure Log Analytics

[Log Analytics](#) is the IT management solution for the hybrid cloud. Used alone or to extend your existing System Center deployment, Log Analytics gives you the maximum flexibility and control for cloud-based management of your infrastructure.



With Log Analytics, you can manage any instance in any cloud, including on-premises, Azure, AWS, Windows Server, Linux, VMware, and OpenStack, at a lower cost than competitive solutions. Built for the cloud-first world, Log Analytics offers a new approach to managing your enterprise that is the fastest, most cost-effective way to meet new business challenges and accommodate new workloads, applications and cloud environments.

Log analytics

[Log Analytics](#) provides monitoring services by collecting data from managed resources into a central repository. This data could include events, performance data, or custom data provided through the API. Once collected, the data is available for alerting, analysis, and export.

The screenshot shows the Azure Log Analytics dashboard. On the left, there's an 'Overview' section with a red-bordered box around 'Recommendations' showing 20 Total. Below it are sections for 'Prevention' (Compute, Networking, Storage & data, Applications) and 'Detection' (Security alerts, Most attacked resources). On the right, a separate window titled 'Recommendations' lists various configuration tasks with columns for Description, Resource, State, and Severity.

DESCRIPTION	RESOURCE	STATE	SEVERITY
Enable VM Agent	3 virtual mac...	Open	High
Install Endpoint Protection	8 virtual mac...	Open	High
Add a web application firewall	2 web applic...	Open	High
Add a Next Generation Firewall	6 endpoints	Open	High
Finalize Internet facing endpoint protec...	VM3-RDP-M...	Open	High
Enable Network Security Groups on sub...	3 subnets	Open	High
Enable Network Security Groups on virt...	vm1classic	Open	High
Route traffic through NGFW only	vm3	Open	High
Enable Auditing & Threat detection on...	sqlserver1as...	Open	High
Remediate vulnerabilities (by Qualys)	2 virtual mac...	Open	High
Enable Auditing & Threat detection on...	2 SQL datab...	Open	High
Apply a Just-In-Time network access co...	7 virtual mac...	Open	High
Apply system updates	3 virtual mac...	Open	High
Apply disk encryption	12 virtual ma...	Open	High
Enable encryption for Azure Storage Ac...	19 storage a...	Open	High
Restrict access through Internet facing...	6 virtual mac...	Open	Medium
Add a vulnerability assessment solution	8 virtual mac...	Open	Medium

This method allows you to consolidate data from a variety of sources, so you can combine data from your Azure services with your existing on-premises environment. It also clearly separates the collection of the data from the action taken on that data so that all actions are available to all kinds of data.

Azure Security Center

[Azure Security Center](#) helps you prevent, detect, and respond to threats with increased visibility into and control over the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Security Center analyzes the security state of your Azure resources to identify potential security vulnerabilities. A list of recommendations guides you through the process of configuring needed controls.

Examples include:

- Provisioning antimalware to help identify and remove malicious software
- Configuring network security groups and rules to control traffic to VMs
- Provisioning of web application firewalls to help defend against attacks that target your web applications
- Deploying missing system updates
- Addressing OS configurations that do not match the recommended baselines

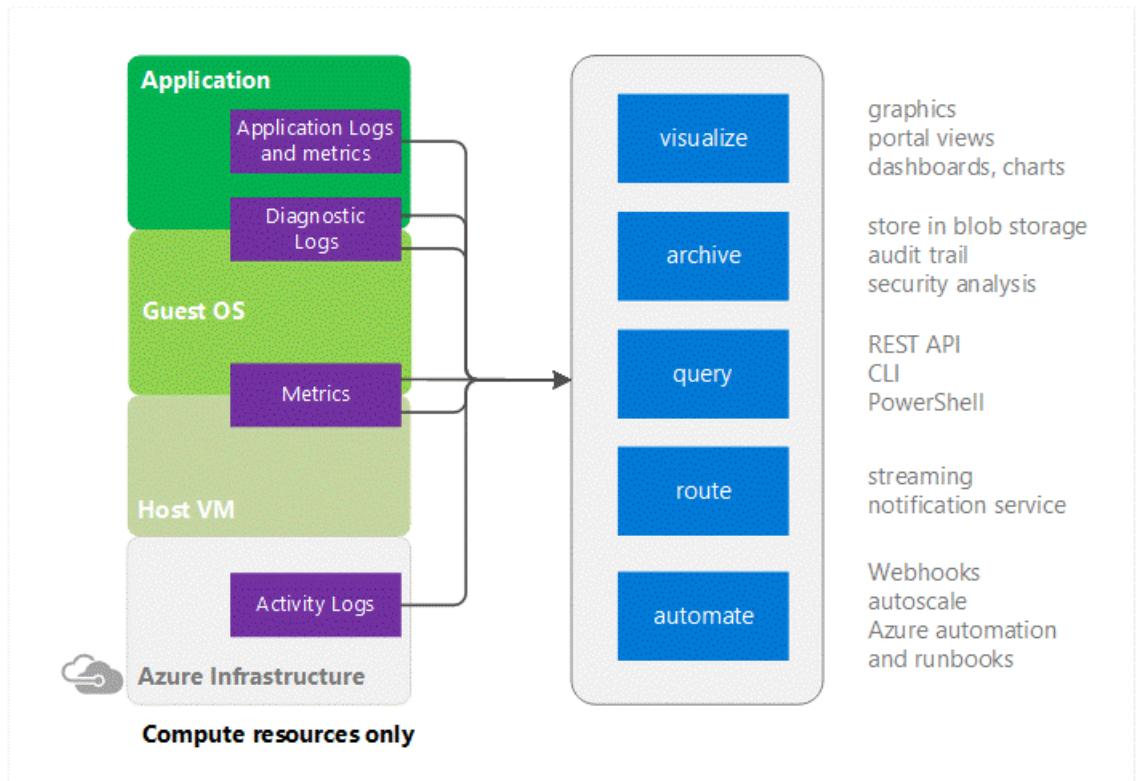
Security Center automatically collects, analyzes, and integrates log data from your Azure resources, the network, and partner solutions like antimalware programs and firewalls. When threats are detected, a security alert is created. Examples include detection of:

- Compromised VMs communicating with known malicious IP addresses
- Advanced malware detected by using Windows error reporting
- Brute force attacks against VMs
- Security alerts from integrated antimalware programs and firewalls

Azure monitor

[Azure Monitor](#) provides pointers to information on specific types of resources. It offers visualization, query, routing, alerting, auto scale, and automation on data both from the Azure infrastructure (Activity Log) and each individual Azure resource (Diagnostic Logs).

Cloud applications are complex with many moving parts. Monitoring provides data to ensure that your application stays up and running in a healthy state. It also helps you to stave off potential problems or troubleshoot past ones.



In addition, you can use monitoring data to gain deep insights about your application. That knowledge can help you to improve application performance or maintainability, or automate actions that would otherwise require manual intervention.

Auditing your network security is vital for detecting network vulnerabilities and ensuring compliance with your IT security and regulatory governance model. With Security Group view, you can retrieve the configured Network Security Group and security rules, as well as the effective security rules. With the list of rules applied, you can determine the ports that are open and ss network vulnerability.

Network watcher

[Network Watcher](#) is a regional service that enables you to monitor and diagnose conditions at a network level in, to, and from Azure. Network diagnostic and visualization tools available with Network Watcher help you understand, diagnose, and gain insights to your network in Azure. This service includes packet capture, next hop, IP flow verify, security group view, NSG flow logs. Scenario level monitoring provides an end to end view of network resources in contrast to individual network resource monitoring.

Storage analytics

[Storage Analytics](#) can store metrics that include aggregated transaction statistics and capacity data about requests to a storage service. Transactions are reported at both the API operation level as well as at the storage service level, and capacity is reported at the storage service level. Metrics data can be used to analyze storage service usage, diagnose issues with requests made against the storage service, and to improve the performance of applications that use a service.

Application Insights

[Application Insights](#) is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Use it to monitor your live web application. It will automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and J2EE, hosted on-premises or in the cloud. It integrates with your devOps process, and has connection points to a various development tools.

It monitors:

- **Request rates, response times, and failure rates** - Find out which pages are most popular, at what times of day, and where your users are. See which pages perform best. If your response times and failure rates go high when there are more requests, then perhaps you have a resourcing problem.
- **Dependency rates, response times, and failure rates** - Find out whether external services are slowing you down.
- **Exceptions** - Analyze the aggregated statistics, or pick specific instances and drill into the stack trace and related requests. Both server and browser exceptions are reported.
- **Page views and load performance** - reported by your users' browsers.
- **AJAX calls from web pages** - rates, response times, and failure rates.
- **User and session counts.**
- **Performance counters** from your Windows or Linux server machines, such as CPU, memory, and network usage.
- **Host diagnostics** from Docker or Azure.
- **Diagnostic trace logs** from your app - so that you can correlate trace events with requests.
- **Custom events and metrics** that you write yourself in the client or server code, to track business events such as items sold, or games won.

The infrastructure for your application is typically made up of many components – maybe a virtual machine, storage account, and virtual network, or a web app, database, database server, and 3rd party services. You do not see these components as separate entities, instead you see them as related and interdependent parts of a single entity. You want to deploy, manage, and monitor them as a group. [Azure Resource Manager](#) enables you to work with the resources in your solution as a group.

You can deploy, update, or delete all the resources for your solution in a single, coordinated operation. You use a template for deployment and that template can work for different environments such as testing, staging, and

production. Resource Manager provides security, auditing, and tagging features to help you manage your resources after deployment.

The benefits of using Resource Manager

Resource Manager provides several benefits:

- You can deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually.
- You can repeatedly deploy your solution throughout the development lifecycle and have confidence your resources are deployed in a consistent state.
- You can manage your infrastructure through declarative templates rather than scripts.
- You can define the dependencies between resources, so they are deployed in the correct order.
- You can apply access control to all services in your resource group because Role-Based Access Control (RBAC) is natively integrated into the management platform.
- You can apply tags to resources to logically organize all the resources in your subscription.
- You can clarify your organization's billing by viewing costs for a group of resources sharing the same tag.

NOTE

Resource Manager provides a new way to deploy and manage your solutions. If you used the earlier deployment model and want to learn about the changes, see [Understanding Resource Manager Deployment and classic deployment](#).

Next steps

Find out more about security by reading some of our in-depth security topics:

- [Auditing and logging](#)
- [Cybercrime](#)
- [Design and operational security](#)
- [Encryption](#)
- [Identity and access management](#)
- [Network security](#)
- [Threat management](#)

Develop secure applications on Azure

11/16/2018 • 2 minutes to read • [Edit Online](#)

Abstract

This paper is a general guide to the security questions and controls you should consider at each phase of the software development lifecycle when developing applications for the cloud. Implementing these concepts before you release your product can help you build more secure software. The recommendations presented in this paper come from our experience with Azure security and the experiences of our customers.

This paper is intended to be a resource for software designers, developers, and testers at all levels who build and deploy secure Azure solutions.

[Download the white paper](#)

Azure encryption overview

9/24/2018 • 11 minutes to read • [Edit Online](#)

This article provides an overview of how encryption is used in Microsoft Azure. It covers the major areas of encryption, including encryption at rest, encryption in flight, and key management with Azure Key Vault. Each section includes links to more detailed information.

Encryption of data at rest

Data at rest includes information that resides in persistent storage on physical media, in any digital format. The media can include files on magnetic or optical media, archived data, and data backups. Microsoft Azure offers a variety of data storage solutions to meet different needs, including file, disk, blob, and table storage. Microsoft also provides encryption to protect [Azure SQL Database](#), [Azure Cosmos DB](#), and Azure Data Lake.

Data encryption at rest is available for services across the software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) cloud models. This article summarizes and provides resources to help you use the Azure encryption options.

For a more detailed discussion of how data at rest is encrypted in Azure, see [Azure Data Encryption-at-Rest](#).

Azure encryption models

Azure supports various encryption models, including server-side encryption that uses service-managed keys, customer-managed keys in Key Vault, or customer-managed keys on customer-controlled hardware. With client-side encryption, you can manage and store keys on-premises or in another secure location.

Client-side encryption

Client-side encryption is performed outside of Azure. It includes:

- Data encrypted by an application that's running in the customer's datacenter or by a service application.
- Data that is already encrypted when it is received by Azure.

With client-side encryption, cloud service providers don't have access to the encryption keys and cannot decrypt this data. You maintain complete control of the keys.

Server-side encryption

The three server-side encryption models offer different key management characteristics, which you can choose according to your requirements:

- **Service-managed keys:** Provides a combination of control and convenience with low overhead.
- **Customer-managed keys:** Gives you control over the keys, including Bring Your Own Keys (BYOK) support, or allows you to generate new ones.
- **Service-managed keys in customer-controlled hardware:** Enables you to manage keys in your proprietary repository, outside of Microsoft control. This characteristic is called Host Your Own Key (HYOK). However, configuration is complex, and most Azure services don't support this model.

Azure disk encryption

You can protect Windows and Linux virtual machines by using [Azure disk encryption](#), which uses [Windows BitLocker](#) technology and Linux [DM-Crypt](#) to protect both operating system disks and data disks with full volume encryption.

Encryption keys and secrets are safeguarded in your [Azure Key Vault subscription](#). By using the Azure Backup service, you can back up and restore encrypted virtual machines (VMs) that use Key Encryption Key (KEK) configuration.

Azure Storage Service Encryption

Data at rest in Azure Blob storage and Azure file shares can be encrypted in both server-side and client-side scenarios.

[Azure Storage Service Encryption \(SSE\)](#) can automatically encrypt data before it is stored, and it automatically decrypts the data when you retrieve it. The process is completely transparent to users. Storage Service Encryption uses 256-bit [Advanced Encryption Standard \(AES\) encryption](#), which is one of the strongest block ciphers available. AES handles encryption, decryption, and key management transparently.

Client-side encryption of Azure blobs

You can perform client-side encryption of Azure blobs in various ways.

You can use the Azure Storage Client Library for .NET NuGet package to encrypt data within your client applications prior to uploading it to your Azure storage.

To learn more about and download the Azure Storage Client Library for .NET NuGet package, see [Windows Azure Storage 8.3.0](#).

When you use client-side encryption with Key Vault, your data is encrypted using a one-time symmetric Content Encryption Key (CEK) that is generated by the Azure Storage client SDK. The CEK is encrypted using a Key Encryption Key (KEK), which can be either a symmetric key or an asymmetric key pair. You can manage it locally or store it in Key Vault. The encrypted data is then uploaded to Azure Storage.

To learn more about client-side encryption with Key Vault and get started with how-to instructions, see [Tutorial: Encrypt and decrypt blobs in Azure Storage by using Key Vault](#).

Finally, you can also use the Azure Storage Client Library for Java to perform client-side encryption before you upload data to Azure Storage, and to decrypt the data when you download it to the client. This library also supports integration with [Key Vault](#) for storage account key management.

Encryption of data at rest with Azure SQL Database

[Azure SQL Database](#) is a general-purpose relational database service in Azure that supports structures such as relational data, JSON, spatial, and XML. SQL Database supports both server-side encryption via the Transparent Data Encryption (TDE) feature and client-side encryption via the Always Encrypted feature.

Transparent Data Encryption

TDE is used to encrypt [SQL Server](#), [Azure SQL Database](#), and [Azure SQL Data Warehouse](#) data files in real time, using a Database Encryption Key (DEK), which is stored in the database boot record for availability during recovery.

TDE protects data and log files, using AES and Triple Data Encryption Standard (3DES) encryption algorithms. Encryption of the database file is performed at the page level. The pages in an encrypted database are encrypted before they are written to disk and are decrypted when they're read into memory. TDE is now enabled by default on newly created Azure SQL databases.

Always Encrypted feature

With the [Always Encrypted](#) feature in Azure SQL you can encrypt data within client applications prior to storing it in Azure SQL Database. You can also enable delegation of on-premises database administration to third parties and maintain separation between those who own and can view the data and those who manage it but should not have access to it.

Cell-level or column-level encryption

With Azure SQL Database, you can apply symmetric encryption to a column of data by using Transact-SQL. This approach is called [cell-level encryption or column-level encryption \(CLE\)](#), because you can use it to encrypt specific

columns or even specific cells of data with different encryption keys. Doing so gives you more granular encryption capability than TDE, which encrypts data in pages.

CLE has built-in functions that you can use to encrypt data by using either symmetric or asymmetric keys, the public key of a certificate, or a passphrase using 3DES.

Cosmos DB database encryption

[Azure Cosmos DB](#) is Microsoft's globally distributed, multi-model database. User data that's stored in Cosmos DB in non-volatile storage (solid-state drives) is encrypted by default. There are no controls to turn it on or off.

Encryption at rest is implemented by using a number of security technologies, including secure key storage systems, encrypted networks, and cryptographic APIs. Encryption keys are managed by Microsoft and are rotated per Microsoft internal guidelines.

At-rest encryption in Data Lake

[Azure Data Lake](#) is an enterprise-wide repository of every type of data collected in a single place prior to any formal definition of requirements or schema. Data Lake Store supports "on by default," transparent encryption of data at rest, which is set up during the creation of your account. By default, Azure Data Lake Store manages the keys for you, but you have the option to manage them yourself.

Three types of keys are used in encrypting and decrypting data: the Master Encryption Key (MEK), Data Encryption Key (DEK), and Block Encryption Key (BEK). The MEK is used to encrypt the DEK, which is stored on persistent media, and the BEK is derived from the DEK and the data block. If you are managing your own keys, you can rotate the MEK.

Encryption of data in transit

Azure offers many mechanisms for keeping data private as it moves from one location to another.

TLS/SSL encryption in Azure

Microsoft uses the [Transport Layer Security](#) (TLS) protocol to protect data when it's traveling between the cloud services and customers. Microsoft datacenters negotiate a TLS connection with client systems that connect to Azure services. TLS provides strong authentication, message privacy, and integrity (enabling detection of message tampering, interception, and forgery), interoperability, algorithm flexibility, and ease of deployment and use.

[Perfect Forward Secrecy](#) (PFS) protects connections between customers' client systems and Microsoft cloud services by unique keys. Connections also use RSA-based 2,048-bit encryption key lengths. This combination makes it difficult for someone to intercept and access data that is in transit.

Azure Storage transactions

When you interact with Azure Storage through the Azure portal, all transactions take place over HTTPS. You can also use the Storage REST API over HTTPS to interact with Azure Storage. You can enforce the use of HTTPS when you call the REST APIs to access objects in storage accounts by enabling the secure transfer that's required for the storage account.

Shared Access Signatures ([SAS](#)), which can be used to delegate access to Azure Storage objects, include an option to specify that only the HTTPS protocol can be used when you use Shared Access Signatures. This approach ensures that anybody who sends links with SAS tokens uses the proper protocol.

[SMB 3.0](#), which used to access Azure Files shares, supports encryption, and it's available in Windows Server 2012 R2, Windows 8, Windows 8.1, and Windows 10. It allows cross-region access and even access on the desktop.

Client-side encryption encrypts the data before it's sent to your Azure Storage instance, so that it's encrypted as it travels across the network.

SMB encryption over Azure virtual networks

By using [SMB 3.0](#) in VMs that are running Windows Server 2012 or later, you can make data transfers secure by

encrypting data in transit over Azure Virtual Networks. By encrypting data, you help protect against tampering and eavesdropping attacks. Administrators can enable SMB encryption for the entire server, or just specific shares.

By default, after SMB encryption is turned on for a share or server, only SMB 3.0 clients are allowed to access the encrypted shares.

In-transit encryption in VMs

Data in transit to, from, and between VMs that are running Windows is encrypted in a number of ways, depending on the nature of the connection.

RDP sessions

You can connect and sign in to a VM by using the [Remote Desktop Protocol \(RDP\)](#) from a Windows client computer, or from a Mac with an RDP client installed. Data in transit over the network in RDP sessions can be protected by TLS.

You can also use Remote Desktop to connect to a Linux VM in Azure.

Secure access to Linux VMs with SSH

For remote management, you can use [Secure Shell \(SSH\)](#) to connect to Linux VMs running in Azure. SSH is an encrypted connection protocol that allows secure sign-ins over unsecured connections. It is the default connection protocol for Linux VMs hosted in Azure. By using SSH keys for authentication, you eliminate the need for passwords to sign in. SSH uses a public/private key pair (asymmetric encryption) for authentication.

Azure VPN encryption

You can connect to Azure through a virtual private network that creates a secure tunnel to protect the privacy of the data being sent across the network.

Azure VPN gateways

You can use an [Azure VPN gateway](#) to send encrypted traffic between your virtual network and your on-premises location across a public connection, or to send traffic between virtual networks.

Site-to-site VPNs use [IPsec](#) for transport encryption. Azure VPN gateways use a set of default proposals. You can configure Azure VPN gateways to use a custom IPsec/IKE policy with specific cryptographic algorithms and key strengths, rather than the Azure default policy sets.

Point-to-site VPNs

Point-to-site VPNs allow individual client computers access to an Azure virtual network. [The Secure Socket Tunneling Protocol \(SSTP\)](#) is used to create the VPN tunnel. It can traverse firewalls (the tunnel appears as an HTTPS connection). You can use your own internal public key infrastructure (PKI) root certificate authority (CA) for point-to-site connectivity.

You can configure a point-to-site VPN connection to a virtual network by using the Azure portal with certificate authentication or PowerShell.

To learn more about point-to-site VPN connections to Azure virtual networks, see:

[Configure a point-to-site connection to a virtual network by using certification authentication: Azure portal](#)

[Configure a point-to-site connection to a virtual network by using certificate authentication: PowerShell](#)

Site-to-site VPNs

You can use a site-to-site VPN gateway connection to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires an on-premises VPN device that has an external-facing public IP address assigned to it.

You can configure a site-to-site VPN connection to a virtual network by using the Azure portal, PowerShell, or Azure CLI.

For more information, see:

[Create a site-to-site connection in the Azure portal](#)

[Create a site-to-site connection in PowerShell](#)

[Create a virtual network with a site-to-site VPN connection by using CLI](#)

In-transit encryption in Data Lake

Data in transit (also known as data in motion) is also always encrypted in Data Lake Store. In addition to encrypting data prior to storing it in persistent media, the data is also always secured in transit by using HTTPS. HTTPS is the only protocol that is supported for the Data Lake Store REST interfaces.

To learn more about encryption of data in transit in Data Lake, see [Encryption of data in Data Lake Store](#).

Key management with Key Vault

Without proper protection and management of the keys, encryption is rendered useless. Key Vault is the Microsoft-recommended solution for managing and controlling access to encryption keys used by cloud services.

Permissions to access keys can be assigned to services or to users through Azure Active Directory accounts.

Key Vault relieves organizations of the need to configure, patch, and maintain hardware security modules (HSMs) and key management software. When you use Key Vault, you maintain control. Microsoft never sees your keys, and applications don't have direct access to them. You can also import or generate keys in HSMs.

Next steps

- [Azure security overview](#)
- [Azure network security overview](#)
- [Azure database security overview](#)
- [Azure virtual machines security overview](#)
- [Data encryption at rest](#)
- [Data security and encryption best practices](#)

Azure database security overview

11/28/2018 • 13 minutes to read • [Edit Online](#)

Security is a top concern for managing databases, and it has always been a priority for Azure SQL Database. Azure SQL Database supports connection security with firewall rules and connection encryption. It supports authentication with username and password and Azure Active Directory (Azure AD) authentication, which uses identities managed by Azure Active Directory. Authorization uses role-based access control.

Azure SQL Database supports encryption by performing real-time encryption and decryption of databases, associated backups, and transaction log files at rest without requiring changes to the application.

Microsoft provides additional ways to encrypt enterprise data:

- Cell-level encryption is available to encrypt specific columns or even cells of data with different encryption keys.
- If you need a hardware security module or central management of your encryption key hierarchy, consider using Azure Key Vault with SQL Server in an Azure virtual machine (VM).
- Always Encrypted (currently in preview) makes encryption transparent to applications. It also allows clients to encrypt sensitive data inside client applications without sharing the encryption keys with SQL Database.

Azure SQL Database Auditing enables enterprises to record events to an audit log in Azure Storage. SQL Database Auditing also integrates with Microsoft Power BI to facilitate drill-down reports and analyses.

Azure SQL databases can be tightly secured to satisfy most regulatory or security requirements, including HIPAA, ISO 27001/27002, and PCI DSS Level 1. A current list of security compliance certifications is available at the [Microsoft Azure Trust Center site](#).

This article walks through the basics of securing Microsoft Azure SQL databases for structured, tabular, and relational data. In particular, this article will get you started with resources for protecting data, controlling access, and proactive monitoring.

Protection of data

SQL Database helps secure your data by providing encryption:

- For data in motion through [Transport Layer Security \(TLS\)](#).
- For data at rest through [transparent data encryption](#).
- For data in use through [Always Encrypted](#).

For other ways to encrypt your data, consider:

- [Cell-level encryption](#) to encrypt specific columns or even cells of data with different encryption keys.
- [Azure Key Vault with SQL Server in an Azure VM](#), if you need a hardware security module or central management of your encryption key hierarchy.

Encryption in motion

A common problem for all client/server applications is the need for privacy as data moves over public and private networks. If data moving over a network is not encrypted, there's a chance that it can be captured and stolen by unauthorized users. When you're dealing with database services, make sure that data is encrypted between the database client and server. Also make sure that data is encrypted between database servers that communicate with each other and with middle-tier applications.

One problem when you administer a network is securing data that's being sent between applications across an untrusted network. You can use [TLS/SSL](#) to authenticate servers and clients, and then use it to encrypt messages between the authenticated parties.

In the authentication process, a TLS/SSL client sends a message to a TLS/SSL server. The server responds with the information that the server needs to authenticate itself. The client and server perform an additional exchange of session keys, and the authentication dialog ends. When authentication is completed, SSL-secured communication can begin between the server and the client through the symmetric encryption keys that are established during the authentication process.

All connections to Azure SQL Database require encryption (TLS/SSL) at all times while data is "in transit" to and from the database. SQL Database uses TLS/SSL to authenticate servers and clients and then use it to encrypt messages between the authenticated parties.

In your application's connection string, you must specify parameters to encrypt the connection and not to trust the server certificate. (This is done for you if you copy your connection string out of the Azure portal.) Otherwise, the connection will not verify the identity of the server and will be susceptible to "man-in-the-middle" attacks. For the ADO.NET driver, for instance, these connection string parameters are `Encrypt=True` and `TrustServerCertificate=False`.

Encryption at rest

You can take several precautions to help secure the database. For example, design a secure system, encrypt confidential assets, and build a firewall around the database servers. But in a scenario where the physical media (such as drives or backup tapes) are stolen, a malicious party can just restore or attach the database and browse the data.

One solution is to encrypt the sensitive data in the database and protect the keys that are used to encrypt the data with a certificate. This solution prevents anyone without the keys from using the data, but this kind of protection must be planned.

To solve this problem, SQL Server and SQL Database support [transparent data encryption](#). Transparent data encryption encrypts SQL Server and SQL Database data files, known as encryption data at rest.

Transparent data encryption helps protect against the threat of malicious activity. It performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application.

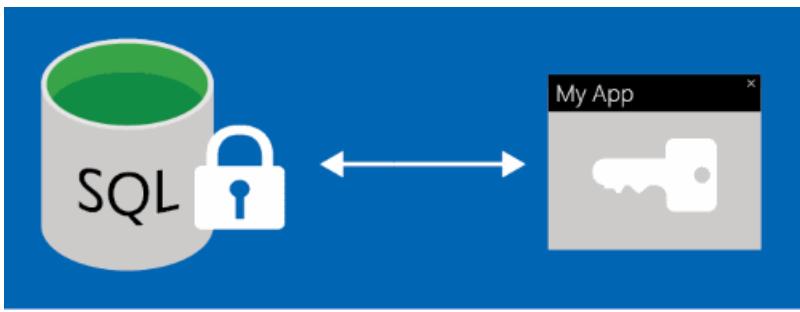
Transparent data encryption encrypts the storage of an entire database by using a symmetric key called the database encryption key. In SQL Database, the database encryption key is protected by a built-in server certificate. The built-in server certificate is unique for each SQL Database server.

If a database is in a Geo-DR relationship, it's protected by a different key on each server. If two databases are connected to the same server, they share the same built-in certificate. Microsoft automatically rotates these certificates at least every 90 days.

For more information, see [Transparent data encryption](#).

Encryption in use (client)

Most data breaches involve the theft of critical data such as credit card numbers or personally identifiable information. Databases can be treasure troves of sensitive information. They can contain customers' personal data (like national identification numbers), confidential competitive information, and intellectual property. Lost or stolen data, especially customer data, can result in brand damage, competitive disadvantage, and serious fines--even lawsuits.



[Always Encrypted](#) is a feature designed to protect sensitive data stored in Azure SQL Database or SQL Server databases. Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to the database engine (SQL Database or SQL Server).

Always Encrypted provides a separation between people who own the data (and can view it) and people who manage the data (but should have no access). It helps ensure that on-premises database administrators, cloud database operators, or other high-privileged but unauthorized users cannot access the encrypted data.

In addition, Always Encrypted makes encryption transparent to applications. An Always Encrypted-enabled driver is installed on the client computer so that it can automatically encrypt and decrypt sensitive data in the client application. The driver encrypts the data in sensitive columns before passing the data to the database engine. The driver automatically rewrites queries so that the semantics to the application are preserved. Similarly, the driver transparently decrypts data, stored in encrypted database columns, contained in query results.

Access control

To provide security, SQL Database controls access by using:

- Firewall rules that limit connectivity by IP address.
- Authentication mechanisms that require users to prove their identity.
- Authorization mechanisms that limit users to specific actions and data.

Database access

Data protection begins with controlling access to your data. The datacenter that hosts your data manages physical access. You can configure a firewall to manage security at the network layer. You also control access by configuring logins for authentication and defining permissions for server and database roles.

Firewall and firewall rules

[Azure SQL Database](#) provides a relational database service for Azure and other internet-based applications. To help protect your data, firewalls prevent all access to your database server until you specify which computers have permission. The firewall grants access to databases based on the originating IP address of each request. For more information, see [Overview of Azure SQL Database firewall rules](#).

The Azure SQL Database service is available only through TCP port 1433. To access a SQL database from your computer, ensure that your client computer firewall allows outgoing TCP communication on TCP port 1433. If inbound connections are not needed for other applications, block them on TCP port 1433.

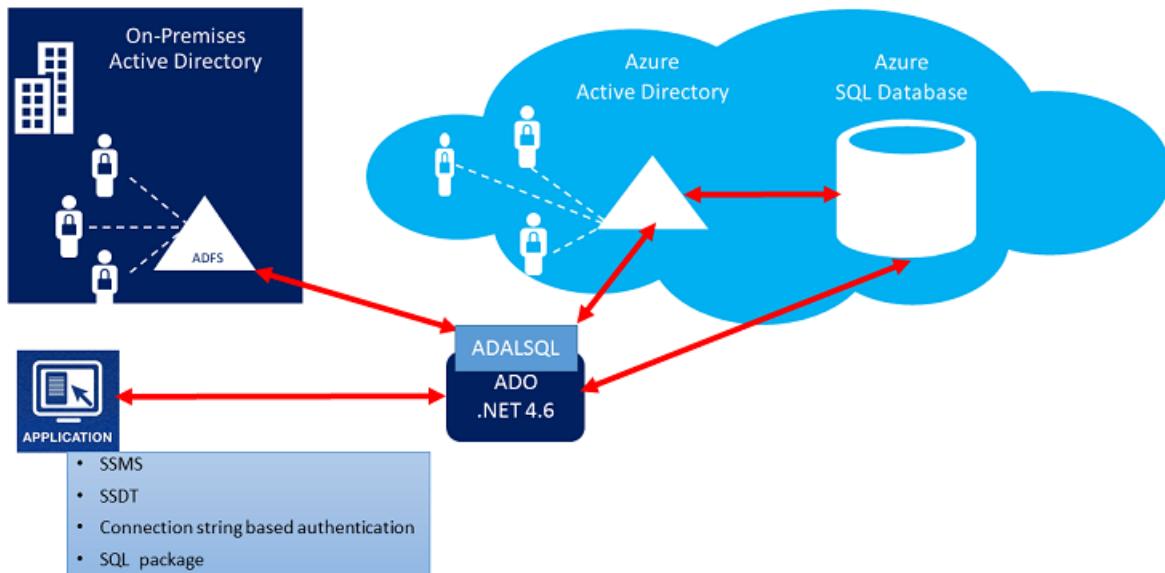
Authentication

Authentication refers to how you prove your identity when connecting to the database. SQL Database supports two types of authentication:

- **SQL Server authentication:** A single login account is created when a logical SQL instance is created, called the SQL Database Subscriber Account. This account connects by using [SQL Server authentication](#) (username and password). This account is an administrator on the logical server instance and on all user databases attached to that instance. The permissions of the subscriber account cannot be restricted. Only one of these accounts can exist.
- **Azure Active Directory authentication:** [Azure AD authentication](#) is a mechanism of connecting to Azure

SQL Database and Azure SQL Data Warehouse by using identities in Azure AD. You can use it to centrally manage identities of database users.

Azure AD Authentication with SQL DB



Advantages of Azure AD authentication include:

- It provides an alternative to SQL Server authentication.
- It helps stop the proliferation of user identities across database servers and allows password rotation in a single place.
- You can manage database permissions by using external (Azure AD) groups.
- It can eliminate storing passwords by enabling integrated Windows authentication and other forms of authentication that Azure AD supports.

Authorization

[Authorization](#) refers to what a user can do within an Azure SQL database. It's controlled by your user account's database [role memberships](#) and [object-level permissions](#). Authorization is the process of determining which securable resources a principal can access, and which operations are allowed for those resources.

Application access

Dynamic data masking

A service representative at a call center might identify callers by several digits of their social security number or credit card number. But those data items should not be fully exposed to the service representative.

You can define a masking rule that masks all but the last four digits of a social security number or credit card number in the result set of any query.



As another example, an appropriate data mask can be defined to protect personally identifiable information. A developer can then query production environments for troubleshooting purposes without violating compliance regulations.

[SQL Database dynamic data masking](#) limits sensitive data exposure by masking it to non-privileged users.

Dynamic data masking is supported for the V12 version of Azure SQL Database.

[Dynamic data masking](#) helps prevent unauthorized access to sensitive data by enabling you to designate how much of the sensitive data to reveal with minimal impact on the application layer. It's a policy-based security feature that hides the sensitive data in the result set of a query over designated database fields, while the data in the database is not changed.

NOTE

Dynamic data masking can be configured by the Azure Database admin, server admin, or security officer roles.

Row-Level Security

Another common security requirement for multitenant databases is [Row-Level Security](#). You can use this feature to control access to rows in a database table based on the characteristics of the user who's executing a query. (Example characteristics are group membership and execution context.)



The access restriction logic is located in the database tier rather than away from the data in another application tier. The database system applies the access restrictions every time that data access is attempted from any tier. This makes your security system more reliable and robust by reducing the surface area of your security system.

Row-Level Security introduces predicate-based access control. It features a flexible, centralized evaluation that can take into consideration metadata or any other criteria the administrator determines as appropriate. The predicate is

used as a criterion to determine whether or not the user has the appropriate access to the data based on user attributes. You can implement label-based access control by using predicate-based access control.

Proactive monitoring

SQL Database helps secure your data by providing *auditing* and *threat detection* capabilities.

Auditing

[Azure SQL Database auditing](#) increases your ability to gain insight into events and changes that occur within the database. Examples are updates and queries against the data.

SQL Database auditing tracks database events and writes them to an audit log in your Azure storage account. Auditing can help you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that might indicate business concerns or suspected security violations. Auditing enables and facilitates adherence to compliance standards but doesn't guarantee compliance.

You can use SQL Database auditing to:

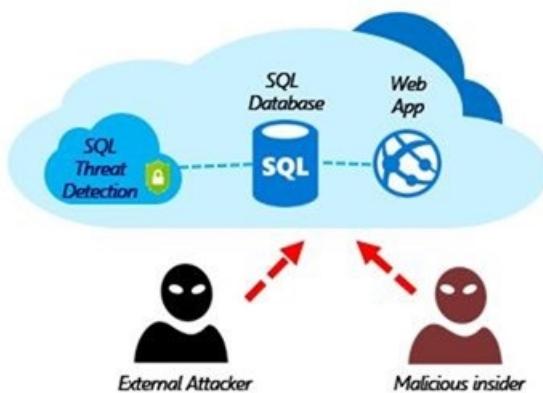
- **Retain** an audit trail of selected events. You can define categories of database actions to be audited.
- **Report** on database activity. You can use pre-configured reports and a dashboard to get started quickly with activity and event reporting.
- **Analyze** reports. You can find suspicious events, unusual activity, and trends.

There are two auditing methods:

- **Blob auditing:** Logs are written to Azure Blob storage. This is a newer auditing method. It provides higher performance, supports higher granularity object-level auditing, and is more cost effective.
- **Table auditing:** Logs are written to Azure Table storage.

Threat detection

[Advanced Threat Protection for Azure SQL Database](#) detects suspicious activities that indicate potential security threats. You can use threat detection to respond to suspicious events in the database, such as SQL injections, as they occur. It provides alerts and allows the use of Azure SQL Database auditing to explore the suspicious events.



SQL Advanced Threat Protection (ATP) provides a set of advanced SQL security capabilities, including Data Discovery & Classification, Vulnerability Assessment, and Threat Detection.

- [Data Discovery & Classification](#)
- [Vulnerability Assessment](#)
- [Threat Detection](#)

[Azure Database for PostgreSQL Advanced Threat Protection](#) provides a new layer of security, which enables you to detect and respond to potential threats as they occur by providing security alerts on anomalous activities. Users receive an alert upon suspicious database activities, and potential vulnerabilities, as well as anomalous database

access and queries patterns. Advanced Threat Protection for Azure Database for PostgreSQL integrates alerts with Azure Security Center. Type of alerts include:

- Access from unusual location
- Access from unusual Azure data center
- Access from unfamiliar principal
- Access from a potentially harmful application
- Brute force Azure database for PostgreSQL credentials

[Azure Database for MySQL Advanced Threat Protection](#) provides protection similar to PostgreSQL Advanced Protection.

Centralized security management

[Azure Security Center](#) helps you prevent, detect, and respond to threats. It provides integrated security monitoring and policy management across your Azure subscriptions. It helps detect threats that might otherwise go unnoticed, and it works with a broad ecosystem of security solutions.

[Security Center](#) helps you safeguard data in SQL Database by providing visibility into the security of all your servers and databases. With Security Center, you can:

- Define policies for SQL Database encryption and auditing.
- Monitor the security of SQL Database resources across all your subscriptions.
- Quickly identify and remediate security issues.
- Integrate alerts from [Azure SQL Database threat detection](#).

Security Center supports role-based access.

SQL Information Protection

[SQL Information Protection](#) automatically discovers and classifies potentially sensitive data, provides a labeling mechanism for persistently tagging sensitive data with classification attributes, and provides a detailed dashboard showing the classification state of the database.

In addition, it calculates the result set sensitivity of SQL queries, so that queries that extract sensitive data can be explicitly audited, and the data can be protected. For more details on SQL Information Protection, see [Azure SQL Database Data Discovery and Classification](#).

You can configure [SQL Information Protection policies](#) in Azure Security Center.

Azure Marketplace

The Azure Marketplace is an online applications and services marketplace that enables start-ups and independent software vendors (ISVs) to offer their solutions to Azure customers around the world. The Azure Marketplace combines Microsoft Azure partner ecosystems into a unified platform to better serve customers and partners. You can [run a search](#) to view database security products available in the Azure Marketplace.

Next steps

- [Secure your Azure SQL database](#)
- [Azure Security Center and Azure SQL Database service](#)
- [SQL Database threat detection](#)
- [Improve SQL database performance](#)

Azure database security best practices

11/7/2018 • 10 minutes to read • [Edit Online](#)

Security is a top concern for managing databases, and it has always been a priority for [Azure SQL Database](#). Your databases can be tightly secured to help satisfy most regulatory or security requirements, including HIPAA, ISO 27001/27002, and PCI DSS Level 1. A current list of security compliance certifications is available at the [Microsoft Trust Center site](#). You also can choose to place your databases in specific Azure datacenters based on regulatory requirements.

In this article, we discuss a collection of Azure database security best practices. These best practices are derived from our experience with Azure database security and the experiences of customers like yourself.

For each best practice, we explain:

- What the best practice is
- Why you want to enable that best practice
- What might be the result if you fail to enable the best practice
- How you can learn to enable the best practice

This Azure Database Security Best Practices article is based on a consensus opinion and Azure platform capabilities and feature sets as they exist at the time this article was written. Opinions and technologies change over time and this article will be updated on a regular basis to reflect those changes.

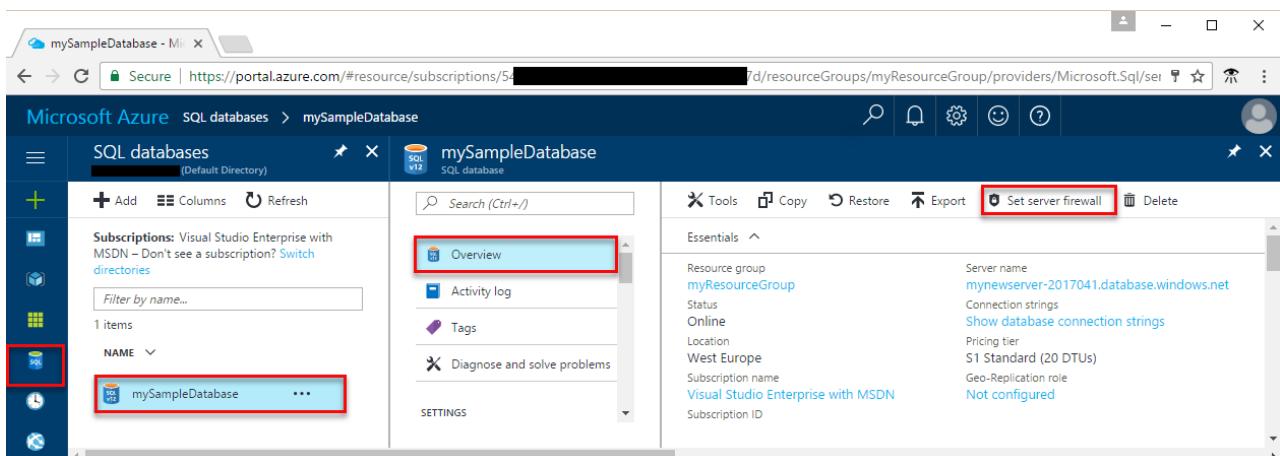
Use firewall rules to restrict database access

Microsoft Azure SQL Database provides a relational database service for Azure and other internet-based applications. To provide access security, SQL Database controls access with:

- Firewall rules that limit connectivity by IP address.
- Authentication mechanisms that require users to prove their identity.
- Authorization mechanisms that limit users to specific actions and data.

Firewalls prevent all access to your database server until you specify which computers have permission. The firewall grants access to databases based on the originating IP address of each request.

The following figure shows where you set a server firewall in SQL Database:



The screenshot shows the Azure portal interface for managing a SQL database. The URL in the browser is https://portal.azure.com/#resource/subscriptions/5d/resourceGroups/myResourceGroup/providers/Microsoft.Sql/servers/mynewserver-2017041/databases/mySampleDatabase. The main content area displays the 'mySampleDatabase' SQL database settings. On the right, under the 'Essentials' section, the 'Set server firewall' button is highlighted with a red box. In the left sidebar, the 'Overview' tab is selected, and the database name 'mySampleDatabase' is highlighted with a red box. Other tabs like 'Activity log' and 'Diagnose and solve problems' are also visible.

The Azure SQL Database service is available only through TCP port 1433. To access a SQL database from your computer, ensure that your client computer firewall allows outgoing TCP communication on TCP port 1433. Block

inbound connections on TCP port 1433 by using firewall rules, if you don't need these connections for other applications.

As part of the connection process, connections from Azure virtual machines are redirected to an IP address and port that are unique for each worker role. The port number is in the range from 11000 to 11999. For more information about TCP ports, see [Ports beyond 1433 for ADO.NET 4.5](#).

For more information about firewall rules in SQL Database, see [SQL Database firewall rules](#).

NOTE

In addition to IP rules, the firewall manages virtual network rules. Virtual network rules are based on virtual network service endpoints. Virtual network rules might be preferable to IP rules in some cases. To learn more, see [Virtual network service endpoints and rules for Azure SQL Database](#).

Enable database authentication

SQL Database supports two types of authentication, SQL Server authentication and Azure AD authentication.

SQL Server Authentication

Benefits include the following:

- It allows SQL Database to support environments with mixed operating systems, where all users are not authenticated by a Windows domain.
- Allows SQL Database to support older applications and partner-supplied applications that require SQL Server authentication.
- Allows users to connect from unknown or untrusted domains. An example is an application where established customers connect with assigned SQL Server logins to receive the status of their orders.
- Allows SQL Database to support web-based applications where users create their own identities.
- Allows software developers to distribute their applications by using a complex permission hierarchy based on known, preset SQL Server logins.

NOTE

SQL Server authentication cannot use the Kerberos security protocol.

If you use SQL Server authentication, you must:

- Manage the strong credentials yourself.
- Protect the credentials in the connection string.
- (Potentially) protect the credentials passed over the network from the web server to the database. For more information, see [How to: Connect to SQL Server Using SQL Authentication in ASP.NET 2.0](#).

Azure Active Directory (AD) authentication

Azure AD authentication is a mechanism of connecting to Azure SQL Database and [SQL Data Warehouse](#) by using identities in Azure AD. With Azure AD authentication, you can manage the identities of database users and other Microsoft services in one central location. Central ID management provides a single place to manage database users and simplifies permission management.

NOTE

We recommend the use of Azure AD authentication over the use of SQL Server authentication.

Benefits include the following:

- It provides an alternative to SQL Server authentication.
- It helps stop the proliferation of user identities across database servers.
- It allows password rotation in a single place.
- Customers can manage database permissions by using external (Azure AD) groups.
- It can eliminate storing passwords by enabling integrated Windows authentication and other forms of authentication supported by Azure Active Directory.
- It uses contained database users to authenticate identities at the database level.
- It supports token-based authentication for applications that connect to SQL Database.
- It supports AD FS (domain federation) or native user/password authentication for a local Azure Active Directory instance without domain synchronization.
- Azure AD supports connections from SQL Server Management Studio that use Active Directory Universal Authentication, which includes Multi-Factor Authentication. Multi-Factor Authentication provides strong authentication with a range of verification options—phone call, text message, smart cards with PIN, or mobile app notification. For more information, see [SSMS support for Azure AD Multi-Factor Authentication with SQL Database and SQL Data Warehouse](#).

The configuration steps include the following procedures to configure and use Azure AD authentication:

- Create and populate Azure AD.
- Optional: Associate or change the Active Directory instance that's currently associated with your Azure subscription.
- Create an Azure Active Directory administrator for Azure SQL Database or [Azure SQL Data Warehouse](#).
- Configure your client computers.
- Create contained database users in your database mapped to Azure AD identities.
- Connect to your database by using Azure AD identities.

You can find detailed information in [Use Azure Active Directory authentication for authentication with SQL Database, Managed Instance, or SQL Data Warehouse](#).

Protect your data by using encryption and row-level security

Azure SQL Database transparent data encryption helps protect data on disk and protects against unauthorized access to hardware. It performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application. Transparent data encryption encrypts the storage of an entire database by using a symmetric key called the database encryption key.

Even when the entire storage is encrypted, it's important to also encrypt the database itself. This is an implementation of the defense-in-depth approach for data protection. If you're using Azure SQL Database and want to protect sensitive data (such as credit card or social security numbers), you can encrypt databases with FIPS 140-2 validated 256-bit AES encryption. This encryption meets the requirements of many industry standards (for example, HIPAA and PCI).

Files related to [buffer pool extension \(BPE\)](#) are not encrypted when you encrypt a database by using transparent data encryption. You must use file-system-level encryption tools like [BitLocker](#) or the [Encrypting File System \(EFS\)](#) for BPE-related files.

Because an authorized user like a security administrator or a database administrator can access the data even if the database is encrypted with transparent data encryption, you should also follow these recommendations:

- Enable SQL Server authentication at the database level.
- Use Azure AD authentication by using [RBAC roles](#).
- Make sure that users and applications use separate accounts to authenticate. This way, you can limit the

permissions granted to users and applications and reduce the risk of malicious activity.

- Implement database-level security by using fixed database roles (such as db_datareader or db_datawriter). Or you can create custom roles for your application to grant explicit permissions to selected database objects.

For other ways to secure your data, consider:

- [Cell-level encryption](#) to encrypt specific columns or even cells of data with different encryption keys.
- [Always Encrypted](#), which allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to the Database Engine (SQL Database or SQL Server). As a result, Always Encrypted provides a separation between those who own the data (and can view it) and those who manage the data (but should have no access).
- [Row-Level Security](#), which enables customers to control access to rows in a database table based on the characteristics of the user who is executing a query. (Example characteristics are group membership and execution context.)

Organizations that are not using database-level encryption might be more susceptible to attacks that compromise data located in SQL databases.

You can learn more about SQL Database transparent data encryption by reading the article [Transparent Data Encryption with Azure SQL Database](#).

Enable database auditing

Auditing an instance of the SQL Server Database Engine or an individual database involves tracking and logging events. For SQL Server, you can create audits that contain specifications for server-level events and specifications for database-level events. Audited events can be written to the event logs or to audit files.

There are several levels of auditing for SQL Server, depending on government or standards requirements for your installation. SQL Server auditing provides tools and processes for enabling, storing, and viewing audits on various server and database objects.

[Azure SQL Database auditing](#) tracks database events and writes them to an audit log in your Azure storage account.

Auditing can help you maintain regulatory compliance, understand database activity, and find discrepancies and anomalies that might point to business concerns or security violations. Auditing facilitates adherence to compliance standards but doesn't guarantee compliance.

To learn more about database auditing and how to enable it, see [Get started with SQL database auditing](#).

Enable database threat detection

Threat protection goes beyond detection. Database threat protection includes:

- Discovering and classifying your most sensitive data so you can protect your data.
- Implementing secure configurations on your database so you can protect your database.
- Detecting and responding to potential threats as they occur so you can quickly respond and remediate.

Best practice: Discover, classify, and label the sensitive data in your databases.

Detail: Classify the data in your SQL database by enabling [Data Discovery and Classification](#) in Azure SQL Database. You can monitor access to your sensitive data in the Azure dashboard or download reports.

Best practice: Track database vulnerabilities so you can proactively improve your database security.

Detail: Use the Azure SQL Database [Vulnerability Assessment](#) service, which scans for potential database vulnerabilities. The service employs a knowledge base of rules that flag security vulnerabilities and show deviations from best practices, such as misconfigurations, excessive permissions, and unprotected sensitive data.

The rules are based on Microsoft best practices and focus on the security issues that present the biggest risks to your database and its valuable data. They cover both database-level issues and server-level security issues, like server firewall settings and server-level permissions. These rules also represent many of the requirements from regulatory bodies to meet their compliance standards.

Best practice: Enable threat detection.

Detail: Enable Azure SQL Database [Threat Detection](#) to get security alerts and recommendations on how to investigate and mitigate threats. You get alerts about suspicious database activities, potential vulnerabilities, and SQL injection attacks, as well as anomalous database access and query patterns.

[Advanced Threat Protection](#) is a unified package for advanced SQL security capabilities. It includes the services mentioned earlier: Data Discovery and Classification, Vulnerability Assessment, and Threat Detection. It provides a single location for enabling and managing these capabilities.

Enabling these capabilities helps you:

- Meet data privacy standards and regulatory compliance requirements.
- Control access to your databases and harden their security.
- Monitor a dynamic database environment where changes are hard to track.
- Detect and respond to potential threats.

In addition, Threat Detection integrates alerts with Azure Security Center for a central view of the security state of all of your Azure resources.

Next steps

See [Azure security best practices and patterns](#) for more security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.

The following resources are available to provide more general information about Azure security and related Microsoft services:

- [Azure Security Team Blog](#) - for up to date information on the latest in Azure Security
- [Microsoft Security Response Center](#) - where Microsoft security vulnerabilities, including issues with Azure, can be reported or via email to secure@microsoft.com

Azure database security checklist

11/7/2018 • 2 minutes to read • [Edit Online](#)

To help improve security, Azure Database includes a number of built-in security controls that you can use to limit and control access.

These include:

- A firewall that enables you to create [firewall rules](#) limiting connectivity by IP address,
- Server-level firewall accessible from the Azure portal
- Database-level firewall rules accessible from SSMS
- Secure connectivity to your database using secure connection strings
- Use access management
- Data encryption
- SQL Database auditing
- SQL Database threat detection

Introduction

Cloud computing requires new security paradigms that are unfamiliar to many application users, database administrators, and programmers. As a result, some organizations are hesitant to implement a cloud infrastructure for data management due to perceived security risks. However, much of this concern can be alleviated through a better understanding of the security features built into Microsoft Azure and Microsoft Azure SQL Database.

Checklist

We recommend that you read the [Azure Database Security Best Practices](#) article prior to reviewing this checklist. You will be able to get the most out of this checklist after you understand the best practices. You can then use this checklist to make sure that you've addressed the important issues in Azure database security.

CHECKLIST CATEGORY	DESCRIPTION
Protect Data	
Encryption in Motion/Transit	<ul style="list-style-type: none">• Transport Layer Security, for data encryption when data is moving to the networks.• Database requires secure communication from clients based on the TDS(Tabular Data Stream) protocol over TLS (Transport Layer Security).
Encryption at rest	<ul style="list-style-type: none">• Transparent Data Encryption, when inactive data is stored physically in any digital form.
Control Access	

CHECKLIST CATEGORY	DESCRIPTION
Database Access	<ul style="list-style-type: none"> • Authentication (Azure Active Directory Authentication) AD authentication uses identities managed by Azure Active Directory. • Authorization grant users the least privileges necessary.
Application Access	<ul style="list-style-type: none"> • Row level Security (Using Security Policy, at the same time restricting row-level access based on a user's identity, role, or execution context). • Dynamic Data Masking (Using Permission & Policy, limits sensitive data exposure by masking it to non-privileged users)
Proactive Monitoring	
Tracking & Detecting	<ul style="list-style-type: none"> • Auditing tracks database events and writes them to an Audit log/ Activity log in your Azure Storage account. • Track Azure Database health using Azure Monitor Activity Logs. • Threat Detection detects anomalous database activities indicating potential security threats to the database.
Azure Security Center	<ul style="list-style-type: none"> • Data Monitoring Use Azure Security Center as a centralized security monitoring solution for SQL and other Azure services.

Conclusion

Azure Database is a robust database platform, with a full range of security features that meet many organizational and regulatory compliance requirements. You can easily protect data by controlling the physical access to your data, and using a variety of options for data security at the file-, column-, or row-level with Transparent Data Encryption, Cell-Level Encryption, or Row-Level Security. Always Encrypted also enables operations against encrypted data, simplifying the process of application updates. In turn, access to auditing logs of SQL Database activity provides you with the information you need, allowing you to know how and when data is accessed.

Next steps

You can improve the protection of your database against malicious users or unauthorized access with just a few simple steps. In this tutorial you learn to:

- Set up [firewall rules](#) for your server and or database.
- Protect your data with [encryption](#).
- Enable [SQL Database auditing](#).

Azure Data Security and Encryption Best Practices

11/7/2018 • 9 minutes to read • [Edit Online](#)

To help protect data in the cloud, you need to account for the possible states in which your data can occur, and what controls are available for that state. Best practices for Azure data security and encryption relate to the following data states:

- At rest: This includes all information storage objects, containers, and types that exist statically on physical media, whether magnetic or optical disk.
- In transit: When data is being transferred between components, locations, or programs, it's in transit. Examples are transfer over the network, across a service bus (from on-premises to cloud and vice-versa, including hybrid connections such as ExpressRoute), or during an input/output process.

In this article we will discuss a collection of Azure data security and encryption best practices. These best practices are derived from our experience with Azure data security and encryption and the experiences of customers like yourself.

For each best practice, we'll explain:

- What the best practice is
- Why you want to enable that best practice
- What might be the result if you fail to enable the best practice
- Possible alternatives to the best practice
- How you can learn to enable the best practice

This Azure Data Security and Encryption Best Practices article is based on a consensus opinion, and Azure platform capabilities and feature sets, as they exist at the time this article was written. Opinions and technologies change over time and this article will be updated on a regular basis to reflect those changes.

Choose a key management solution

Protecting your keys is essential to protecting your data in the cloud.

[Azure Key Vault](#) helps safeguard cryptographic keys and secrets that cloud applications and services use. Key Vault streamlines the key management process and enables you to maintain control of keys that access and encrypt your data. Developers can create keys for development and testing in minutes, and then migrate them to production keys. Security administrators can grant (and revoke) permission to keys, as needed.

You can use Key Vault to create multiple secure containers, called vaults. These vaults are backed by HSMs. Vaults help reduce the chances of accidental loss of security information by centralizing the storage of application secrets. Key vaults also control and log the access to anything stored in them. Azure Key Vault can handle requesting and renewing Transport Layer Security (TLS) certificates. It provides features for a robust solution for certificate lifecycle management.

Azure Key Vault is designed to support application keys and secrets. Key Vault is not intended to be a store for user passwords.

Following are security best practices for using Key Vault.

Best practice: Grant access to users, groups, and applications at a specific scope.

Detail: Use RBAC's predefined roles. For example, to grant access to a user to manage key vaults, you would assign the predefined role [Key Vault Contributor](#) to this user at a specific scope. The scope in this case would be a

subscription, a resource group, or just a specific key vault. If the predefined roles don't fit your needs, you can [define your own roles](#).

Best practice: Control what users have access to.

Detail: Access to a key vault is controlled through two separate interfaces: management plane and data plane. The management plane and data plane access controls work independently.

Use RBAC to control what users have access to. For example, if you want to grant an application access to use keys in a key vault, you only need to grant data plane access permissions by using key vault access policies, and no management plane access is needed for this application. Conversely, if you want a user to be able to read vault properties and tags but not have any access to keys, secrets, or certificates, you can grant this user read access by using RBAC, and no access to the data plane is required.

Best practice: Store certificates in your key vault. Your certificates are of high value. In the wrong hands, your application's security or the security of your data can be compromised.

Detail: Azure Resource Manager can securely deploy certificates stored in Azure Key Vault to Azure VMs when the VMs are deployed. By setting appropriate access policies for the key vault, you also control who gets access to your certificate. Another benefit is that you manage all your certificates in one place in Azure Key Vault. See [Deploy Certificates to VMs from customer-managed Key Vault](#) for more information.

Best practice: Ensure that you can recover a deletion of key vaults or key vault objects.

Detail: Deletion of key vaults or key vault objects can be inadvertent or malicious. Enable the soft delete and purge protection features of Key Vault, particularly for keys that are used to encrypt data at rest. Deletion of these keys is equivalent to data loss, so you can recover deleted vaults and vault objects if needed. Practice Key Vault recovery operations on a regular basis.

NOTE

If a user has contributor permissions (RBAC) to a key vault management plane, they can grant themselves access to the data plane by setting a key vault access policy. We recommend that you tightly control who has contributor access to your key vaults, to ensure that only authorized persons can access and manage your key vaults, keys, secrets, and certificates.

Manage with secure workstations

NOTE

The subscription administrator or owner should use a secure access workstation or a privileged access workstation.

Because the vast majority of attacks target the end user, the endpoint becomes one of the primary points of attack. An attacker who compromises the endpoint can use the user's credentials to gain access to the organization's data. Most endpoint attacks take advantage of the fact that users are administrators in their local workstations.

Best practice: Use a secure management workstation to protect sensitive accounts, tasks, and data.

Detail: Use a [privileged access workstation](#) to reduce the attack surface in workstations. These secure management workstations can help you mitigate some of these attacks and ensure that your data is safer.

Best practice: Ensure endpoint protection.

Detail: Enforce security policies across all devices that are used to consume data, regardless of the data location (cloud or on-premises).

Protect data at rest

[Data encryption at rest](#) is a mandatory step toward data privacy, compliance, and data sovereignty.

Best practice: Apply disk encryption to help safeguard your data.

Detail: Use [Azure Disk Encryption](#). It enables IT administrators to encrypt Windows and Linux IaaS VM disks. Disk Encryption combines the industry-standard Windows BitLocker feature and the Linux dm-crypt feature to provide volume encryption for the OS and the data disks.

Azure Storage and Azure SQL Database encrypt data at rest by default, and many services offer encryption as an option. You can use Azure Key Vault to maintain control of keys that access and encrypt your data. See [Azure resource providers encryption model support to learn more](#).

Best practices: Use encryption to help mitigate risks related to unauthorized data access.

Detail: Encrypt your drives before you write sensitive data to them.

Organizations that don't enforce data encryption are more exposed to data-integrity issues. For example, unauthorized or rogue users might steal data in compromised accounts or gain unauthorized access to data coded in Clear Format. Companies also must prove that they are diligent and using correct security controls to enhance their data security in order to comply with industry regulations.

Protect data in transit

Protecting data in transit should be an essential part of your data protection strategy. Because data is moving back and forth from many locations, we generally recommend that you always use SSL/TLS protocols to exchange data across different locations. In some circumstances, you might want to isolate the entire communication channel between your on-premises and cloud infrastructures by using a VPN.

For data moving between your on-premises infrastructure and Azure, consider appropriate safeguards such as HTTPS or VPN. When sending encrypted traffic between an Azure virtual network and an on-premises location over the public internet, use [Azure VPN Gateway](#).

Following are best practices specific to using Azure VPN Gateway, SSL/TLS, and HTTPS.

Best practice: Secure access from multiple workstations located on-premises to an Azure virtual network.

Detail: Use [site-to-site VPN](#).

Best practice: Secure access from an individual workstation located on-premises to an Azure virtual network.

Detail: Use [point-to-site VPN](#).

Best practice: Move larger data sets over a dedicated high-speed WAN link.

Detail: Use [ExpressRoute](#). If you choose to use ExpressRoute, you can also encrypt the data at the application level by using [SSL/TLS](#) or other protocols for added protection.

Best practice: Interact with Azure Storage through the Azure portal.

Detail: All transactions occur via HTTPS. You can also use [Storage REST API](#) over HTTPS to interact with [Azure Storage](#) and [Azure SQL Database](#).

Organizations that fail to protect data in transit are more susceptible to [man-in-the-middle attacks](#), [eavesdropping](#), and session hijacking. These attacks can be the first step in gaining access to confidential data.

Secure email, documents, and sensitive data

You want to control and secure email, documents, and sensitive data that you share outside your company. [Azure Information Protection](#) is a cloud-based solution that helps an organization to classify, label, and protect its documents and emails. This can be done automatically by administrators who define rules and conditions, manually by users, or a combination where users get recommendations.

Classification is identifiable at all times, regardless of where the data is stored or with whom it's shared. The labels include visual markings such as a header, footer, or watermark. Metadata is added to files and email headers in clear text. The clear text ensures that other services, such as solutions to prevent data loss, can identify the classification and take appropriate action.

The protection technology uses Azure Rights Management (Azure RMS). This technology is integrated with other Microsoft cloud services and applications, such as Office 365 and Azure Active Directory. This protection technology uses encryption, identity, and authorization policies. Protection that is applied through Azure RMS stays with the documents and emails, independently of the location—inside or outside your organization, networks, file servers, and applications.

This information protection solution keeps you in control of your data, even when it's shared with other people. You can also use Azure RMS with your own line-of-business applications and information protection solutions from software vendors, whether these applications and solutions are on-premises or in the cloud.

We recommend that you:

- [Deploy Azure Information Protection](#) for your organization.
- Apply labels that reflect your business requirements. For example: Apply a label named "highly confidential" to all documents and emails that contain top-secret data, to classify and protect this data. Then, only authorized users can access this data, with any restrictions that you specify.
- Configure [usage logging for Azure RMS](#) so that you can monitor how your organization is using the protection service.

Organizations that are weak on [data classification](#) and file protection might be more susceptible to data leakage or data misuse. With proper file protection, you can analyze data flows to gain insight into your business, detect risky behaviors and take corrective measures, track access to documents, and so on.

Next steps

See [Azure security best practices and patterns](#) for more security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.

The following resources are available to provide more general information about Azure security and related Microsoft services:

- [Azure Security Team Blog](#) - for up to date information on the latest in Azure Security
- [Microsoft Security Response Center](#) - where Microsoft security vulnerabilities, including issues with Azure, can be reported or via email to secure@microsoft.com

Azure Data Encryption-at-Rest

10/31/2018 • 19 minutes to read • [Edit Online](#)

Microsoft Azure includes tools to safeguard data according to your company's security and compliance needs.

This paper focuses on:

- How data is protected at rest across Microsoft Azure
- Discusses the various components taking part in the data protection implementation,
- Reviews pros and cons of the different key management protection approaches.

Encryption at Rest is a common security requirement. In Azure, organizations can encrypt data at rest without the risk or cost of a custom key management solution. Organizations have the option of letting Azure completely manage Encryption at Rest. Additionally, organizations have various options to closely manage encryption or encryption keys.

What is encryption at rest?

Encryption at Rest is the encoding (encryption) of data when it is persisted. The Encryption at Rest designs in Azure use symmetric encryption to encrypt and decrypt large amounts of data quickly according to a simple conceptual model:

- A symmetric encryption key is used to encrypt data as it is written to storage.
- The same encryption key is used to decrypt that data as it is readied for use in memory.
- Data may be partitioned, and different keys may be used for each partition.
- Keys must be stored in a secure location with identity-based access control and audit policies. Data encryption keys are often encrypted with asymmetric encryption to further limit access.

In practice, key management and control scenarios, as well as scale and availability assurances, require additional constructs. Microsoft Azure Encryption at Rest concepts and components are described below.

The purpose of encryption at rest

Encryption at rest provides data protection for stored data (at rest). Attacks against data at-rest include attempts to obtain physical access to the hardware on which the data is stored, and then compromise the contained data. In such an attack, a server's hard drive may have been mishandled during maintenance allowing an attacker to remove the hard drive. Later the attacker would put the hard drive into a computer under their control to attempt to access the data.

Encryption at rest is designed to prevent the attacker from accessing the unencrypted data by ensuring the data is encrypted when on disk. If an attacker obtains a hard drive with encrypted data but not the encryption keys, the attacker must defeat the encryption to read the data. This attack is much more complex and resource consuming than accessing unencrypted data on a hard drive. For this reason, encryption at rest is highly recommended and is a high priority requirement for many organizations.

Encryption at rest may also be required by an organization's need for data governance and compliance efforts. Industry and government regulations such as HIPAA, PCI and FedRAMP, lay out specific safeguards regarding data protection and encryption requirements. Encryption at rest is a mandatory measure required for compliance with some of those regulations.

In addition to satisfying compliance and regulatory requirements, encryption at rest provides defense-in-depth protection. Microsoft Azure provides a compliant platform for services, applications, and data. It also provides

comprehensive facility and physical security, data access control, and auditing. However, it's important to provide additional "overlapping" security measures in case one of the other security measures fails and encryption at rest provides such a security measure

Microsoft is committed to encryption at rest options across cloud services and giving customers control of encryption keys and logs of key use. Additionally, Microsoft is working towards encrypting all customer data at rest by default.

Azure Encryption at Rest Components

As described previously, the goal of encryption at rest is that data that is persisted on disk is encrypted with a secret encryption key. To achieve that goal secure key creation, storage, access control, and management of the encryption keys must be provided. Though details may vary, Azure services Encryption at Rest implementations can be described in terms illustrated in the following diagram.



Azure Key Vault

The storage location of the encryption keys and access control to those keys is central to an encryption at rest model. The keys need to be highly secured but manageable by specified users and available to specific services. For Azure services, Azure Key Vault is the recommended key storage solution and provides a common management experience across services. Keys are stored and managed in key vaults, and access to a key vault can be given to users or services. Azure Key Vault supports customer creation of keys or import of customer keys for use in customer-managed encryption key scenarios.

Azure Active Directory

Permissions to use the keys stored in Azure Key Vault, either to manage or to access them for Encryption at Rest encryption and decryption, can be given to Azure Active Directory accounts.

Key Hierarchy

More than one encryption key is used in an encryption at rest implementation. Asymmetric encryption is useful for establishing the trust and authentication needed for key access and management. Symmetric encryption is more efficient for bulk encryption and decryption, allowing for stronger encryption and better performance. Limiting the use of a single encryption key decreases the risk that the key will be compromised and the cost of re-encryption when a key must be replaced. Azure encryptions at rest models use a key hierarchy made up of the following types of keys:

- **Data Encryption Key (DEK)** – A symmetric AES256 key used to encrypt a partition or block of data. A single resource may have many partitions and many Data Encryption Keys. Encrypting each block of data with a different key makes crypto analysis attacks more difficult. Access to DEKs is needed by the resource provider or application instance that is encrypting and decrypting a specific block. When a DEK is replaced with a new key only the data in its associated block must be re-encrypted with the new key.

- **Key Encryption Key (KEK)** – An asymmetric encryption key used to encrypt the Data Encryption Keys. Use of a Key Encryption Key allows the data encryption keys themselves to be encrypted and controlled. The entity that has access to the KEK may be different than the entity that requires the DEK. An entity may broker access to the DEK to limit the access of each DEK to a specific partition. Since the KEK is required to decrypt the DEKs, the KEK is effectively a single point by which DEKs can be effectively deleted by deletion of the KEK.

The Data Encryption Keys, encrypted with the Key Encryption Keys are stored separately and only an entity with access to the Key Encryption Key can get any Data Encryption Keys encrypted with that key. Different models of key storage are supported. We will discuss each model in more detail later in the next section.

Data Encryption Models

An understanding of the various encryption models and their pros and cons is essential for understanding how the various resource providers in Azure implement encryption at Rest. These definitions are shared across all resource providers in Azure to ensure common language and taxonomy.

There are three scenarios for server-side encryption:

- Server-side encryption using Service-Managed keys
 - Azure Resource Providers perform the encryption and decryption operations
 - Microsoft manages the keys
 - Full cloud functionality
- Server-side encryption using customer-managed keys in Azure Key Vault
 - Azure Resource Providers perform the encryption and decryption operations
 - Customer controls keys via Azure Key Vault
 - Full cloud functionality
- Server-side encryption using customer-managed keys on customer-controlled hardware
 - Azure Resource Providers perform the encryption and decryption operations
 - Customer controls keys on customer-controlled hardware
 - Full cloud functionality

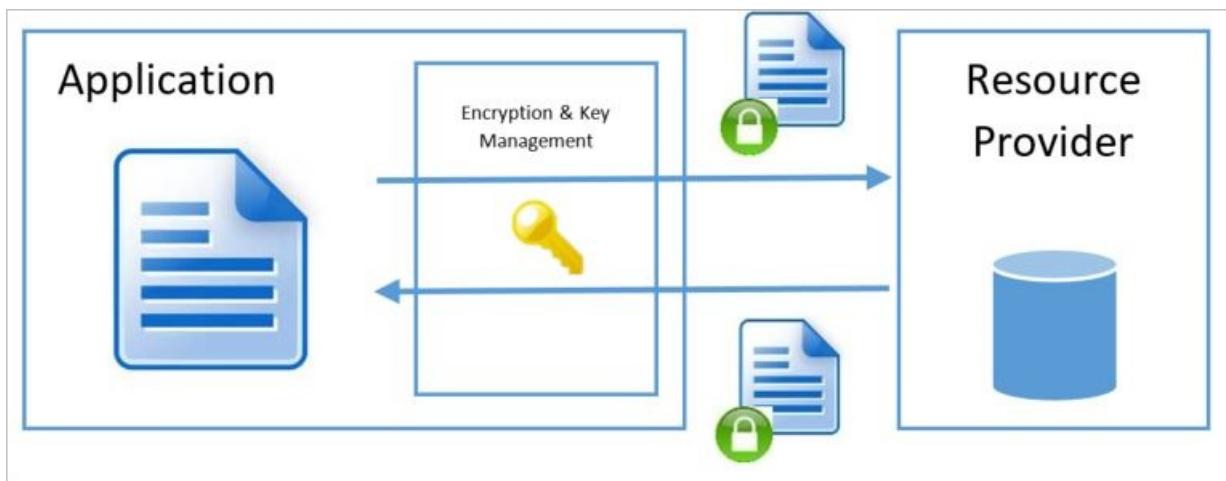
For client-side encryption, consider the following:

- Azure services cannot see decrypted data
- Customers manage and store keys on-premises (or in other secure stores). Keys are not available to Azure services
- Reduced cloud functionality

The supported encryption models in Azure split into two main groups: "Client Encryption" and "Server-side Encryption" as mentioned previously. Independent of the encryption at rest model used, Azure services always recommend the use of a secure transport such as TLS or HTTPS. Therefore, encryption in transport should be addressed by the transport protocol and should not be a major factor in determining which encryption at rest model to use.

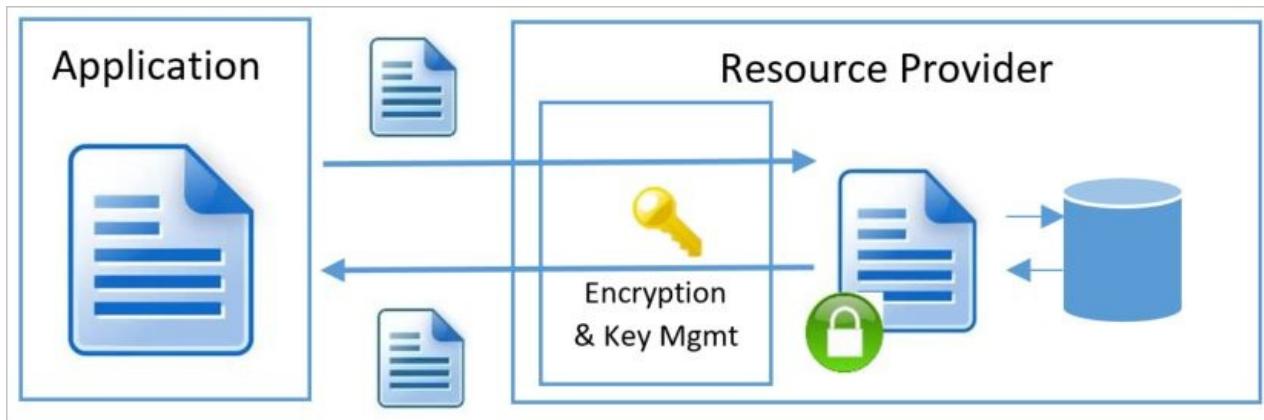
Client encryption model

Client Encryption model refers to encryption that is performed outside of the Resource Provider or Azure by the service or calling application. The encryption can be performed by the service application in Azure, or by an application running in the customer data center. In either case, when leveraging this encryption model, the Azure Resource Provider receives an encrypted blob of data without the ability to decrypt the data in any way or have access to the encryption keys. In this model, the key management is done by the calling service/application and is opaque to the Azure service.



Server-side encryption model

Server-side Encryption models refer to encryption that is performed by the Azure service. In that model, the Resource Provider performs the encrypt and decrypt operations. For example, Azure Storage may receive data in plain text operations and will perform the encryption and decryption internally. The Resource Provider might use encryption keys that are managed by Microsoft or by the customer depending on the provided configuration.

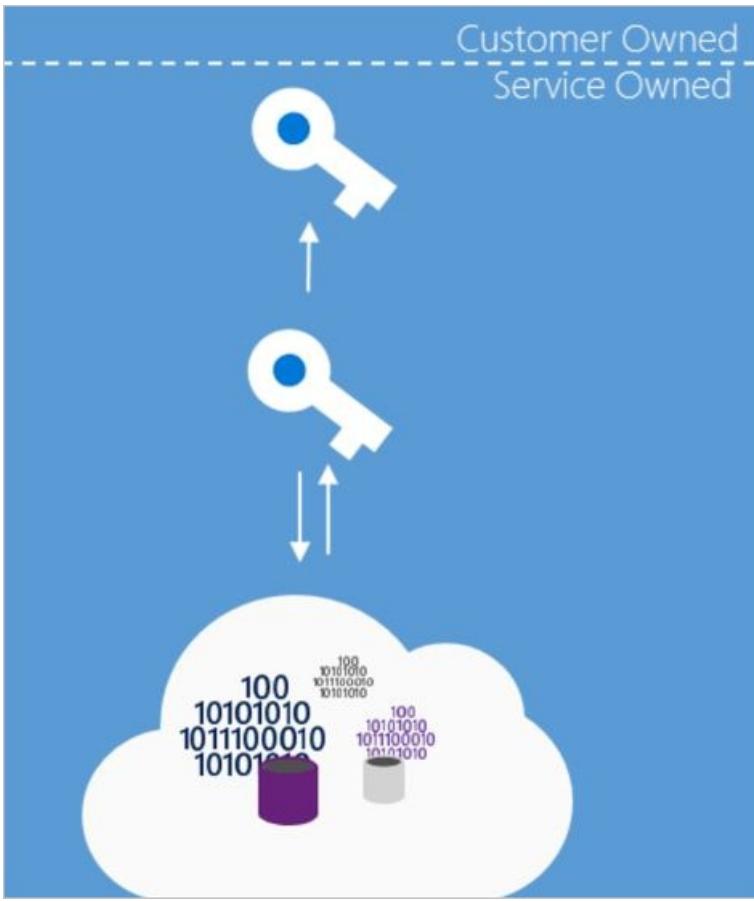


Server-side encryption key management models

Each of the server-side encryption at rest models implies distinctive characteristics of key management. This includes where and how encryption keys are created, and stored as well as the access models and the key rotation procedures.

Server-side encryption using service-managed keys

For many customers, the essential requirement is to ensure that the data is encrypted whenever it is at rest. Server-side encryption using service-managed Keys enables this model by allowing customers to mark the specific resource (Storage Account, SQL DB, etc.) for encryption and leaving all key management aspects such as key issuance, rotation, and backup to Microsoft. Most Azure Services that support encryption at rest typically support this model of offloading the management of the encryption keys to Azure. The Azure resource provider creates the keys, places them in secure storage, and retrieves them when needed. This means that the service has full access to the keys and the service has full control over the credential lifecycle management.



Server-side encryption using service-managed keys therefore quickly addresses the need to have encryption at rest with low overhead to the customer. When available a customer typically opens the Azure portal for the target subscription and resource provider and checks a box indicating, they would like the data to be encrypted. In some Resource Managers server-side encryption with service-managed keys is on by default.

Server-side encryption with Microsoft-managed keys does imply the service has full access to store and manage the keys. While some customers may want to manage the keys because they feel they gain greater security, the cost and risk associated with a custom key storage solution should be considered when evaluating this model. In many cases, an organization may determine that resource constraints or risks of an on-premises solution may be greater than the risk of cloud management of the encryption at rest keys. However, this model might not be sufficient for organizations that have requirements to control the creation or lifecycle of the encryption keys or to have different personnel manage a service's encryption keys than those managing the service (that is, segregation of key management from the overall management model for the service).

Key access

When Server-side encryption with service-managed keys is used, the key creation, storage, and service access are all managed by the service. Typically, the foundational Azure resource providers will store the Data Encryption Keys in a store that is close to the data and quickly available and accessible while the Key Encryption Keys are stored in a secure internal store.

Advantages

- Simple setup
- Microsoft manages key rotation, backup, and redundancy
- Customer does not have the cost associated with implementation or the risk of a custom key management scheme.

Disadvantages

- No customer control over the encryption keys (key specification, lifecycle, revocation, etc.)
- No ability to segregate key management from overall management model for the service

Server-side encryption using customer-managed keys in Azure Key Vault

For scenarios where the requirement is to encrypt the data at rest and control the encryption keys customers can use server-side encryption using customer-managed Keys in Key Vault. Some services may store only the root Key Encryption Key in Azure Key Vault and store the encrypted Data Encryption Key in an internal location closer to the data. In that scenario customers can bring their own keys to Key Vault (BYOK – Bring Your Own Key), or generate new ones, and use them to encrypt the desired resources. While the Resource Provider performs the encryption and decryption operations it uses the configured key as the root key for all encryption operations.

Key Access

The server-side encryption model with customer-managed keys in Azure Key Vault involves the service accessing the keys to encrypt and decrypt as needed. Encryption at rest keys are made accessible to a service through an access control policy. This policy grants the service identity access to receive the key. An Azure service running on behalf of an associated subscription can be configured with an identity in that subscription. The service can perform Azure Active Directory authentication and receive an authentication token identifying itself as that service acting on behalf of the subscription. That token can then be presented to Key Vault to obtain a key it has been given access to.

For operations using encryption keys, a service identity can be granted access to any of the following operations: decrypt, encrypt, unwrapKey, wrapKey, verify, sign, get, list, update, create, import, delete, backup, and restore.

To obtain a key for use in encrypting or decrypting data at rest the service identity that the Resource Manager service instance will run as must have UnwrapKey (to get the key for decryption) and WrapKey (to insert a key into key vault when creating a new key).

NOTE

For more detail on Key Vault authorization see the secure your key vault page in the [Azure Key Vault documentation](#).

Advantages

- Full control over the keys used – encryption keys are managed in the customer's Key Vault under the customer's control.
- Ability to encrypt multiple services to one master
- Can segregate key management from overall management model for the service
- Can define service and key location across regions

Disadvantages

- Customer has full responsibility for key access management
- Customer has full responsibility for key lifecycle management
- Additional Setup & configuration overhead

Server-side encryption using service-managed keys in customer-controlled hardware

Some Azure services enable the Host Your Own Key (HYOK) key management model. This management mode is useful in scenarios where there is a need to encrypt the data at rest and manage the keys in a proprietary repository outside of Microsoft's control. In this model, the service must retrieve the key from an external site. Performance and availability guarantees are impacted, and configuration is more complex. Additionally, since the service does have access to the DEK during the encryption and decryption operations the overall security guarantees of this model are similar to when the keys are customer-managed in Azure Key Vault. As a result, this model is not appropriate for most organizations unless they have specific key management requirements. Due to these limitations, most Azure Services do not support server-side encryption using server-managed keys in customer-controlled hardware.

Key Access

When server-side encryption using service-managed keys in customer-controlled hardware is used the keys are

maintained on a system configured by the customer. Azure services that support this model provide a means of establishing a secure connection to a customer supplied key store.

Advantages

- Full control over the root key used – encryption keys are managed by a customer provided store
- Ability to encrypt multiple services to one master
- Can segregate key management from overall management model for the service
- Can define service and key location across regions

Disadvantages

- Full responsibility for key storage, security, performance, and availability
- Full responsibility for key access management
- Full responsibility for key lifecycle management
- Significant setup, configuration, and ongoing maintenance costs
- Increased dependency on network availability between the customer datacenter and Azure datacenters.

Encryption at rest in Microsoft cloud services

Microsoft Cloud services are used in all three cloud models: IaaS, PaaS, SaaS. Below you have examples of how they fit on each model:

- Software services, referred to as Software as a Server or SaaS, which have application provided by the cloud such as Office 365.
- Platform services which customers leverage the cloud in their applications, using the cloud for things like storage, analytics, and service bus functionality.
- Infrastructure services, or Infrastructure as a Service (IaaS) in which customer deploys operating systems and applications that are hosted in the cloud and possibly leveraging other cloud services.

Encryption at rest for SaaS customers

Software as a Service (SaaS) customers typically have encryption at rest enabled or available in each service. Office 365 has several options for customers to verify or enable encryption at rest. For information about Office 365 services, see [Encryption in Office 365](#).

Encryption at rest for PaaS customers

Platform as a Service (PaaS) customer's data typically resides in an application execution environment and any Azure Resource Providers used to store customer data. To see the encryption at rest options available to you, examine the table below for the storage and application platforms that you use. Where supported, links to instructions on enabling Encryption at Rest are provided for each resource provider.

Encryption at rest for IaaS customers

Infrastructure as a Service (IaaS) customers can have a variety of services and applications in use. IaaS services can enable encryption at rest in their Azure hosted virtual machines and VHDS using Azure Disk Encryption.

Encrypted storage

Like PaaS, IaaS solutions can leverage other Azure services that store data encrypted at rest. In these cases, you can enable the Encryption at Rest support as provided by each consumed Azure service. The below table enumerates the major storage, services, and application platforms and the model of Encryption at Rest supported. Where supported, links are provided to instructions on enabling Encryption at Rest.

Encrypted compute

A complete Encryption at Rest solution requires that the data is never persisted in unencrypted form. While in use, on a server loading the data in memory, data can be persisted locally in various ways including the Windows page file, a crash dump, and any logging the application may perform. To ensure this data is encrypted at rest, IaaS

applications can use Azure Disk Encryption on an Azure IaaS virtual machine (Windows or Linux) and virtual disk.

Custom encryption at rest

It is recommended that whenever possible, IaaS applications leverage Azure Disk Encryption and Encryption at Rest options provided by any consumed Azure services. In some cases, such as irregular encryption requirements or non-Azure based storage, a developer of an IaaS application may need to implement encryption at rest themselves. Developers of IaaS solutions can better integrate with Azure management and customer expectations by leveraging certain Azure components. Specifically, developers should use the Azure Key Vault service to provide secure key storage as well as provide their customers with consistent key management options with that of most Azure platform services. Additionally, custom solutions should use Azure-Managed Service Identities to enable service accounts to access encryption keys. For developer information on Azure Key Vault and Managed Service Identities, see their respective SDKs.

Azure resource providers encryption model support

Microsoft Azure Services each support one or more of the encryption at rest models. For some services, however, one or more of the encryption models may not be applicable. For services that support customer-managed key scenarios, they may support only a subset of the key types that Azure Key Vault supports for key encryption keys. Additionally, services may release support for these scenarios and key types at different schedules. This section describes the encryption at rest support at the time of this writing for each of the major Azure data storage services.

Azure disk encryption

Any customer using Azure Infrastructure as a Service (IaaS) features can achieve encryption at rest for their IaaS VMs and disks through Azure Disk Encryption. For more information on Azure Disk encryption, see the [Azure Disk Encryption documentation](#).

Azure storage

All Azure Storage services (Blob storage, Queue storage, Table storage, and Azure Files) support server-side encryption at rest, with some services supporting customer-managed keys and client-side encryption.

- Server-side: All Azure Storage Services enable server-side encryption by default using service-managed keys, which is transparent to the application. For more information, see [Azure Storage Service Encryption for Data at Rest](#). Azure Blob storage and Azure Files also support RSA 2048-bit customer-managed keys in Azure Key Vault. For more information, see [Storage Service Encryption using customer-managed keys in Azure Key Vault](#).
- Client-side: Azure Blobs, Tables, and Queues support client-side encryption. When using client-side encryption, customers encrypt the data and upload the data as an encrypted blob. Key management is done by the customer. For more information, see [Client-Side Encryption and Azure Key Vault for Microsoft Azure Storage](#).

Azure SQL Database

Azure SQL Database currently supports encryption at rest for Microsoft-managed service side and client-side encryption scenarios.

Support for server encryption is currently provided through the SQL feature called Transparent Data Encryption. Once an Azure SQL Database customer enables TDE key are automatically created and managed for them. Encryption at rest can be enabled at the database and server levels. As of June 2017, [Transparent Data Encryption \(TDE\)](#) is enabled by default on newly created databases. Azure SQL Database supports RSA 2048-bit customer-managed keys in Azure Key Vault. For more information, see [Transparent Data Encryption with Bring Your Own Key support for Azure SQL Database and Data Warehouse](#).

Client-side encryption of Azure SQL Database data is supported through the [Always Encrypted](#) feature. Always Encrypted uses a key that created and stored by the client. Customers can store the master key in a Windows certificate store, Azure Key Vault, or a local Hardware Security Module. Using SQL Server Management Studio, SQL users choose what key they'd like to use to encrypt which column.

		ENCRYPTION MODEL AND KEY MANAGEMENT	
	Server-Side Using Service-Managed Key	Server-Side Using Customer-Managed in Key Vault	Client-Side Using Client-Managed
Storage and Databases			
Disk (IaaS)	-	Yes, RSA 2048-bit	-
SQL Server (IaaS)	Yes	Yes, RSA 2048-bit	Yes
Azure SQL (Database/Data Warehouse)	Yes	Yes, RSA 2048-bit	Yes
Azure Storage (Block/Page Blobs)	Yes	Yes, RSA 2048-bit	Yes
Azure Storage (Files)	Yes	Yes, RSA 2048-bit	-
Azure Storage (Tables, Queues)	Yes	-	Yes
Cosmos DB (Document DB)	Yes	-	-
StorSimple	Yes	-	Yes
Backup	-	-	Yes
Intelligence and Analytics			
Azure Data Factory	Yes	-	-
Azure Machine Learning	-	Preview, RSA 2048-bit	-
Azure Stream Analytics	Yes	-	-
HDInsight (Azure Blob Storage)	Yes	-	-
HDInsight (Data Lake Storage)	Yes	-	-
Apache Kafka for HDInsight	Yes	Preview, All RSA Lengths	-
Azure Data Lake Store	Yes	Yes, RSA 2048-bit	-
Azure Data Catalog	Yes	-	-
Power BI	Yes	-	-
IoT Services			

		ENCRYPTION MODEL AND KEY MANAGEMENT	
IoT Hub	-	-	Yes
Service Bus	Yes	-	Yes
Event Hubs	Yes	-	-
Event Grid	Yes	-	-

Conclusion

Protection of customer data stored within Azure Services is of paramount importance to Microsoft. All Azure hosted services are committed to providing Encryption at Rest options. Foundational services such as Azure Storage, Azure SQL Database, and key analytics and intelligence services already provide Encryption at Rest options. Some of these services support either customer controlled keys and client-side encryption as well as service-managed keys and encryption. Microsoft Azure services are broadly enhancing Encryption at Rest availability and new options are planned for preview and general availability in the upcoming months.

Azure Disk Encryption for IaaS VMs

12/7/2018 • 9 minutes to read • [Edit Online](#)

Microsoft Azure is committed to ensuring your data privacy and data sovereignty. Azure enables you to control your Azure-hosted data through a range of advanced technologies to encrypt, control and manage encryption keys, and control and audit access of data. This control provides Azure customers with the flexibility to choose the solution that best meets their business needs. This article introduces you to a technology solution: "Azure Disk Encryption for Windows and Linux IaaS virtual machines (VMs)." This technology helps protect and safeguard your data to meet your organizational security and compliance commitments.

NOTE

If you're interested in viewing or deleting personal data, please see the [Azure Data Subject Requests for the GDPR](#) article. If you're looking for general info about GDPR, see the [GDPR section of the Service Trust portal](#).

Overview

Azure Disk Encryption is a capability that helps you encrypt your Windows and Linux IaaS VM disks. Disk Encryption leverages the industry standard [BitLocker](#) feature of Windows and the [DM-Crypt](#) feature of Linux to provide volume encryption for the OS and data disks. The solution is integrated with [Azure Key Vault](#) to help you control and manage the disk-encryption keys and secrets. The solution also ensures that all data on the VM disks are encrypted at rest in your Azure storage.

Disk Encryption for Windows and Linux IaaS VMs is in General Availability in all Azure public regions and Azure Government regions for Standard VMs and VMs with Azure Premium Storage. When you apply the Disk Encryption management solution, you can satisfy the following business needs:

- IaaS VMs are secured at rest by using industry-standard encryption technology to address organizational security and compliance requirements.
- IaaS VMs boot under customer-controlled keys and policies. You can audit their usage in your key vault.

If you use Azure Security Center, 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			
Virtual machines					
NAME	ONBOARDING	SYSTEM UPDATES	ANTIMALWARE	BASELINE	DISK ENCRYPTION
ASC-VM1	✓	✓	✓	✓	!
ASC-VM2	✓	✓	✓	✓	!

NOTE

Certain recommendations might increase data, network, or compute resource usage and result in additional license or subscription costs.

Encryption scenarios

The Disk Encryption solution supports the following customer scenarios:

- Enable encryption on new Windows IaaS VMs created from pre-encrypted VHD and encryption keys.
- Enable encryption on new IaaS VMs created from the supported Azure Gallery images.
- Enable encryption on existing IaaS VMs that run in Azure.
- Enable encryption on Windows virtual machine scale sets.
- Enable encryption on data drives for Linux virtual machine scale sets.
- Disable encryption on Windows IaaS VMs.
- Disable encryption on data drives for Linux IaaS VMs.
- Disable encryption on Windows virtual machine scale sets.
- Disable encryption on data drives for Linux virtual machine scale sets.
- Enable encryption of managed disk VMs.
- Update encryption settings of an existing encrypted Premium and non-Premium Storage VM.
- Back up and restore of encrypted VMs.

The solution supports the following scenarios for IaaS VMs when they're enabled in Microsoft Azure:

- Integration with Azure Key Vault.
- Standard tier VMs: [A](#), [D](#), [DS](#), [G](#), [GS](#), [F](#), and so on, series IaaS VMs. Linux VMs within these tiers must meet the minimum memory requirement of 7 GB.
- Enable encryption on Windows and Linux IaaS VMs, managed disk, and scale set VMs from the supported Azure Gallery images.
- Disable encryption on OS and data drives for Windows IaaS VMs, scale set VMs, and managed disk VMs.
- Disable encryption on data drives for Linux IaaS VMs, scale set VMs, and managed disk VMs.
- Enable encryption on IaaS VMs that run the Windows Client OS.
- Enable encryption on volumes with mount paths.
- Enable encryption on Linux VMs that are configured with disk striping (RAID) by using mdadm.
- Enable encryption on Linux VMs that use LVM for data disks.
- Enable encryption on the Linux VM OS and data disks.

NOTE

OS drive encryption for some Linux distributions isn't supported. For more information, see the [Azure Disk Encryption FAQ](#) article.

- Enable encryption on Windows VMs that are configured with Windows Storage Spaces.
- Update encryption settings for an existing encrypted Premium and non-Premium Storage VM.
- Back up and restore of encrypted VMs for both key encryption key (KEK) and non-KEK scenarios.
- All Azure Public and Azure Government regions are supported.

The solution doesn't support the following scenarios, features, and technology:

- Basic tier IaaS VMs.

- Disable encryption on an OS drive for Linux IaaS VMs.
- Disable encryption on a data drive when the OS drive is encrypted for Linux IaaS VMs.
- IaaS VMs that are created by using the classic VM creation method.
- Enable encryption of customer custom images on Linux IaaS VMs.
- Integration with your on-premises key management system.
- Azure Files (shared file system).
- Network File System (NFS).
- Dynamic volumes.
- Windows VMs that are configured with software-based RAID systems.

Encryption features

When you enable and deploy Disk Encryption for Azure IaaS VMs, the following capabilities are enabled depending on the provided configuration:

- Encryption of the OS volume to protect the boot volume at rest in your storage.
- Encryption of data volumes to protect the data volumes at rest in your storage.
- Disable encryption on the OS and data drives for Windows IaaS VMs.
- Disable encryption on the data drives for Linux IaaS VMs (only when the OS drive isn't encrypted).
- Safeguard the encryption keys and secrets in your Azure Key Vault subscription.
- Report the encryption status of the encrypted IaaS VM.
- Remove the disk encryption configuration settings from the IaaS VM.
- Back up and restore the encrypted VMs by using the Azure Backup service.

Azure Disk Encryption for IaaS VMs for Windows and Linux solution includes:

- The disk encryption extension for Windows.
- The disk encryption extension for Linux.
- The PowerShell disk encryption cmdlets.
- The Azure CLI disk encryption cmdlets.
- The Azure Resource Manager disk encryption templates.

The Azure Disk Encryption solution is supported on IaaS VMs that run Windows or Linux OS. For more information about the supported operating systems, see the [Prerequisites](#) article.

NOTE

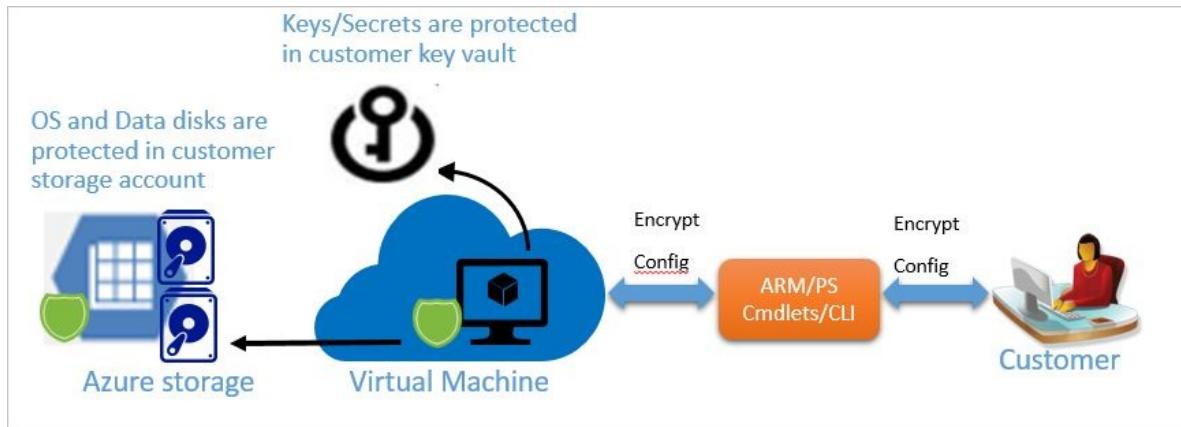
There's no additional charge to encrypt VM disks with Azure Disk Encryption. Standard [Key Vault pricing](#) applies to the key vault that's used to store the encryption keys.

Encryption workflow

To enable disk encryption for Windows and Linux VMs, do the following steps:

1. Choose an encryption scenario from the scenarios listed in the [Encryption scenarios](#) section.
2. Opt in to enable disk encryption via the Azure Disk Encryption Resource Manager template, PowerShell cmdlets, or the Azure CLI, and specify the encryption configuration.
 - For the customer-encrypted VHD scenario, upload the encrypted VHD to your storage account and the encryption key material to your key vault. Then, provide the encryption configuration to enable encryption on a new IaaS VM.

- For new VMs that are created from the Marketplace and existing VMs that already run in Azure, provide the encryption configuration to enable encryption on the IaaS VM.
- Grant access to the Azure platform to read the encryption key material (BitLocker encryption keys for Windows systems and Passphrase for Linux) from your key vault to enable encryption on the IaaS VM.
 - Azure updates the VM service model with encryption and the key vault configuration, and sets up your encrypted VM.



Decryption workflow

To disable disk encryption for IaaS VMs, complete the following high-level steps:

- Choose to disable encryption (decryption) on a running IaaS VM in Azure and specify the decryption configuration. You can disable via the Azure Disk Encryption Resource Manager template, PowerShell cmdlets, or the Azure CLI.

This step disables encryption of the OS or the data volume or both on the running Windows IaaS VM. As mentioned in the previous section, disabling OS disk encryption for Linux isn't supported. The decryption step is allowed only for data drives on Linux VMs as long as the OS disk isn't encrypted.

- Azure updates the VM service model and the IaaS VM is marked as decrypted. The contents of the VM are no longer encrypted at rest.

NOTE

The disable encryption operation doesn't delete your key vault and the encryption key material (BitLocker encryption keys for Windows systems or Passphrase for Linux).

Disabling OS disk encryption for Linux isn't supported. The decryption step is allowed only for data drives on Linux VMs.

Disabling data disk encryption for Linux isn't supported if the OS drive is encrypted.

Encryption workflow (previous release)

The new release of Azure Disk Encryption eliminates the requirement to provide an Azure Active Directory (Azure AD) application parameter to enable VM disk encryption. With the new release, you're no longer required to provide an Azure AD credential during the enable encryption step. All new VMs must be encrypted without the Azure AD application parameters when you use the new release. VMs that were already encrypted with Azure AD application parameters are still supported and should continue to be maintained with the Azure AD syntax. To enable disk encryption for Windows and Linux VMs (previous release), do the following steps:

- Choose an encryption scenario from the scenarios listed in the [Encryption scenarios](#) section.

2. Opt in to enable disk encryption via the Azure Disk Encryption Resource Manager template, PowerShell cmdlets, or the Azure CLI, and specify the encryption configuration.
 - For the customer-encrypted VHD scenario, upload the encrypted VHD to your storage account and the encryption key material to your key vault. Then, provide the encryption configuration to enable encryption on a new IaaS VM.
 - For new VMs that are created from the Marketplace and existing VMs that already run in Azure, provide the encryption configuration to enable encryption on the IaaS VM.
3. Grant access to the Azure platform to read the encryption key material (BitLocker encryption keys for Windows systems and Passphrase for Linux) from your key vault to enable encryption on the IaaS VM.
4. Provide the Azure AD application identity to write the encryption key material to your key vault. This step enables encryption on the IaaS VM for the scenarios mentioned in step 2.
5. Azure updates the VM service model with encryption and the key vault configuration, and sets up your encrypted VM.

Terminology

The following table defines some of the common terms that are used in this technology:

TERMINOLOGY	DEFINITION
Azure AD	An Azure AD account is used to authenticate, store, and retrieve secrets from a key vault.
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.
BitLocker	BitLocker is an industry-recognized Windows volume encryption technology that's used to enable disk encryption on Windows IaaS VMs.
BEK	BitLocker encryption keys (BEK) are used to encrypt the OS boot volume and data volumes. BEKs are safeguarded in a key vault as secrets.
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 IaaS VMs.
KEK	Key encryption key (KEK) is 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.
PowerShell cmdlets	For more information, see Azure PowerShell cmdlets .

Next steps

Azure Disk Encryption prerequisites

Quickstart: Encrypt a Windows IaaS VM with Azure PowerShell

12/7/2018 • 4 minutes to read • [Edit Online](#)

Azure Disk Encryption helps you encrypt your Windows and Linux IaaS virtual machine disks. The solution is integrated with Azure Key Vault to help you control and manage the disk-encryption keys and secrets. By using Azure Disk encryption, you can ensure that your VMs are secured at rest using industry-standard encryption technology. In this quickstart, you'll create a Windows Server 2016 VM and encrypt the OS disk.

If you don't have an Azure subscription, create a [free account](#) before you begin.

Prerequisites

- Windows PowerShell ISE
- Install or update to the [latest version of Azure PowerShell](#)
 - The AzureRM module version needs to be 6.0.0 or higher.

```
Get-Module AzureRM -ListAvailable | Select-Object -Property Name,Version,Path
```
- A copy of the [Azure Disk Encryption prerequisites script](#).
 - If you have this script already, download a new copy as it has recently changed.
 - Use **CTRL-A** to select all the text then use **CTRL-C** to copy all the text into Notepad.
 - Save the file as **ADEPrereqScript.ps1**

Sign in to Azure

1. Right-click **Windows PowerShell ISE** and click **Run as administrator**.
2. In the **Administrator: Windows PowerShell ISE** window, click **View** and then click **Show Script Pane**.
3. In the script pane, type the following cmdlet:

```
Connect-AzureRMAccount
```

4. Click on the green arrow for **Run Script**, or use F5.
5. Use the interactive sign-in to finish connecting to your Azure account.
6. Copy your **subscription ID** that is returned for use in running the next PowerShell script.

Run the Azure Disk Encryption prerequisites script

ADEPrereqScript.ps1 will create a resource group, a key vault, and set the key vault access policy. The script also creates a resource lock on the key vault to help protect it from accidental deletion.

1. In the **Administrator: Windows PowerShell ISE** window, click **File** and then click **Open**. Navigate to the **ADEPrereqScript.ps1** file and double-click on it. The script will open in the script pane.
2. Click on the green arrow for **Run Script**, or use F5 to run the script.
3. Type in names for a new **resource group** and a new **key vault**. Don't use an existing resource group or key vault for this quickstart since we'll delete the resource group later.
4. Type in the location where you want to create the resources, such as **EastUS**. Get a location list with

```
Get-AzureRMLocation .
```

5. Copy in your **subscription ID**. You can get your Subscription ID with `Get-AzureRMSubscription`.
6. Click on the green arrow for **Run Script**.
7. Copy the returned **DiskEncryptionKeyVaultUrl** and **DiskEncryptionKeyId** to be used later.

```

1 #Requires -Module AzureRM.Resources
2 #Requires -Module AzureRM.KeyVault
3
4 [CmdletBinding()]
5 Param(
6     [Parameter(Mandatory = $true,
7             HelpMessage="Name of the resource group to which the KeyVault belongs to. A new resource group
8             [ValidateNotNullOrEmpty()]$resourceGroupName,
9
10    [Parameter(Mandatory = $true,
11             HelpMessage="Name of the KeyVault in which encryption keys are to be placed. A new vault with t
12             [ValidateNotNullOrEmpty()]$keyVaultName,
13
14    [Parameter(Mandatory = $true,
15
PS C:\windows\system32> C:\Scripts\ADEPrereqScript.ps1
cmdlet ADEPrereqScript.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
resourceGroupName: MySecureRG
keyVaultName: MySecureKV
location: EastUS
subscriptionId: 12345678-1234-1234-1234-123456789012

```

Running script / selection. Press Ctrl+Break to stop. Press Ctrl+B to break into debugger. | Ln 20 Col 53 | 100%

Create a virtual machine

You now need to create a virtual machine so you can encrypt its disk. The script you'll use creates a Windows Server 2016 VM with 8-GB RAM and a 30-GB OS disk.

1. Copy the script into the **Administrator: Windows PowerShell ISE** script pane and change the top three variables. The resource group and location need to be the same as you used for the [prerequisites script](#).

```

# Variables for common values
$resourceGroup = "MySecureRG"
.setLocation = "EastUS"
$vmName = "MySecureVM"

# Create user object
$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

# Create a resource group
#New-AzureRmResourceGroup -Name $resourceGroup -Location $location

# Create a subnet configuration
$subnetConfig = New-AzureRmVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix 192.168.1.0/24

# Create a virtual network
$vnet = New-AzureRmVirtualNetwork -ResourceGroupName $resourceGroup -Location $location ` 
-Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig

# Create a public IP address and specify a DNS name
$pip = New-AzureRmPublicIpAddress -ResourceGroupName $resourceGroup -Location $location ` 
-Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4

# Create an inbound network security group rule for port 3389
$nsgRuleRDP = New-AzureRmNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP -Protocol Tcp
` 
-Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -
DestinationAddressPrefix *
-DestinationPortRange 3389 -Access Allow

# Create a network security group
$nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location ` 
-Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP

# Create a virtual network card and associate with public IP address and NSG
$nic = New-AzureRmNetworkInterface -Name myNic -ResourceGroupName $resourceGroup -Location $location ` 
-SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration
$vmConfig = New-AzureRmVMConfig -VMName $vmName -VMSize Standard_D2_v3 | ` 
Set-AzureRmVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | ` 
Set-AzureRmVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus 2016-
Datacenter-smalldisk -Version latest | ` 
Add-AzureRmVMNetworkInterface -Id $nic.Id

# Create a virtual machine
New-AzureRmVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig

```

- Click on the green arrow for **Run Script** to build the VM.

Encrypt the disk of the VM

Now that you've created and configured a key vault and a VM, you can encrypt the disk with the **Set-AzureRmVmDiskEncryptionExtension** cmdlet.

- Run the following cmdlet to encrypt the VM's disk:

```

Set-AzureRmVmDiskEncryptionExtension -ResourceGroupName "MySecureRG" -VMName "MySecureVM" ` 
-DiskEncryptionKeyVaultId "<Returned by the prerequisites script>" -DiskEncryptionKeyVaultUrl " 
<Returned by the prerequisites script>"

```

- When the encryption finishes, you can verify that the disk is encrypted with the following cmdlet:

```
Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName "MySecureRG" -VMName "MySecureVM"
```

```
PS C:\windows\system32> Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName "MySecureRG" -VMName "MySecureVM"
OsVolumeEncrypted      : Encrypted
DataVolumesEncrypted   : Unknown
OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage        : Provisioning succeeded
```

Clean up resources

ADEPrereqScript.ps1 creates a resource lock on the key vault. To clean up the resources from this quickstart, you need to remove the resource lock first then delete the resource group.

1. Remove the resource lock from the key vault

```
$LockId =(Get-AzureRMResourceLock -ResourceGroupName "MySecureRG" -ResourceName "MySecureVault" -
 ResourceType "Microsoft.KeyVault/vaults").LockID
Remove-AzureRmResourceLock -LockID $LockId
```

2. Remove the resource group. This will delete all resources in the group too.

```
Remove-AzureRmResourceGroup -Name "MySecureRG"
```

Next steps

Advance to the next article to learn more about Azure Disk Encryption prerequisites for IaaS VMs.

[Azure Disk Encryption Prerequisites](#)

Azure Disk Encryption prerequisites

12/13/2018 • 12 minutes to read • [Edit Online](#)

This article, Azure Disk Encryption Prerequisites, explains items that need to be in place before you can use Azure Disk Encryption. Azure Disk Encryption is integrated with [Azure Key Vault](#) to help manage encryption keys. You can use [Azure PowerShell](#), [Azure CLI](#), or the [Azure portal](#) to configure Azure Disk Encryption.

Before you enable Azure Disk Encryption on Azure IaaS VMs for the supported scenarios that were discussed in the [Azure Disk Encryption Overview](#) article, be sure to have the prerequisites in place.

NOTE

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.

Supported operating systems

Azure Disk Encryption is supported on the following operating systems:

- Windows Server versions: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016.
 - For Windows Server 2008 R2, you must have .NET Framework 4.5 installed before you enable encryption in Azure. 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 client versions: Windows 8 client and Windows 10 client.
- Azure Disk Encryption is only supported on specific Azure Gallery based Linux server distributions and versions. For the list of currently supported versions, refer to the [Azure Disk Encryption FAQ](#).
- Azure Disk Encryption requires that your key vault and VMs reside in the same Azure region and subscription. Configuring the resources in separate regions causes a failure in enabling the Azure Disk Encryption feature.

Additional prerequisites for Linux IaaS VMs

- Azure Disk Encryption for Linux requires 7 GB of RAM on the VM to enable OS disk encryption on [supported images](#). Once the OS disk encryption process is complete, the VM can be configured to run with less memory.
- Before enabling encryption, the data disks to be encrypted need to be properly listed in /etc/fstab. Use a persistent block device name for this entry, as device names in the "/dev/sdX" format can't be relied upon to be associated with the same disk across reboots, particularly after encryption is applied. 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.

- 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.

An example of commands that can be used to mount the data disks and create the necessary /etc/fstab entries can be found in [lines 244-248 of this script file](#).

Networking and Group Policy

To enable the Azure Disk Encryption feature, the IaaS VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the IaaS 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 IaaS VM must be able to connect to the key vault endpoint.
- The IaaS 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:

- The Azure Disk Encryption solution uses the BitLocker external key protector for Windows IaaS 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, force the new policy to update (gpupdate.exe /force), and then restarting may be required.

Azure PowerShell

[Azure PowerShell](#) provides a set of cmdlets that uses the [Azure Resource Manager](#) model for managing your Azure resources. You can use it in your browser with [Azure Cloud Shell](#), or you can install it on your local machine using the instructions below to use it in any PowerShell session. If you already have it installed locally, make sure you use the latest version of Azure PowerShell SDK version to configure Azure Disk Encryption. Download the latest version of [Azure PowerShell release](#).

Install Azure PowerShell for use on your local machine (optional):

1. Follow the instructions in the links for your operating system, then continue though the rest of the steps below.
 - [Install and configure Azure PowerShell for Windows](#).
 - Install PowerShellGet, Azure PowerShell, and load the AzureRM module.
 - [Install and configure Azure Powershell on macOS and Linux](#).
 - Install PowerShell Core, Azure PowerShell for .NET Core, and load the Az module.
2. Verify the installed versions of the AzureRM module. If needed, [update the Azure PowerShell module](#).
 - The AzureRM module version needs to be 6.0.0 or higher.
 - Using the latest AzureRM module version is recommended.

```
Get-Module AzureRM -ListAvailable | Select-Object -Property Name,Version,Path
```

3. Sign in to Azure using the [Connect-AzureRmAccount](#) cmdlet.

```
Connect-AzureRmAccount  
# For specific instances of Azure, use the -Environment parameter.  
Connect-AzureRmAccount -Environment (Get-AzureRmEnvironment -Name AzureUSGovernment)  
  
<# If you have multiple subscriptions and want to specify a specific one,  
get your subscription list with Get-AzureRmSubscription and  
specify it with Set-AzureRmContext. #>  
Get-AzureRmSubscription  
Set-AzureRmContext -SubscriptionId "xxxx-xxxx-xxxx-xxxx"
```

4. If needed, review [Getting started with Azure PowerShell](#).

Install the Azure CLI for use on your local machine (optional)

The [Azure CLI 2.0](#) is a command-line tool for managing Azure resources. The CLI is designed to flexibly query data, support long-running operations as non-blocking processes, and make scripting easy. You can use it in your browser with [Azure Cloud Shell](#), or you can install it on your local machine and use it in any PowerShell session.

1. [Install Azure CLI](#) for use on your local machine (optional):

2. Verify the installed version.

```
az --version
```

3. Sign in to Azure using [az login](#).

```
az login  
  
# If you would like to select a tenant, use:  
az login --tenant "<tenant>"  
  
# If you have multiple subscriptions, get your subscription list with az account list and specify with  
az account set.  
az account list  
az account set --subscription "<subscription name or ID>"
```

4. Review [Get started with Azure CLI 2.0](#) if needed.

Prerequisite workflow for Key Vault

If you're already familiar with the Key Vault and Azure AD prerequisites for Azure Disk Encryption, you can use the [Azure Disk Encryption prerequisites PowerShell script](#). For more information on using the prerequisites script, see the [Encrypt a VM Quickstart](#) and the [Azure Disk Encryption Appendix](#).

1. If needed, create a resource group.
2. Create a key vault.
3. Set key vault advanced access policies.

WARNING

Before deleting a key vault, ensure that you did not encrypt any existing VMs with it. To protect a vault from accidental deletion, [enable soft delete](#) and a [resource lock](#) on the vault.

Create a key vault

Azure Disk Encryption is integrated with [Azure Key Vault](#) to help you control and manage the disk-encryption keys and secrets in your key vault subscription. You can create a key vault or use an existing one for Azure Disk Encryption. For more information about key vaults, see [Get started with Azure Key Vault](#) and [Secure your key vault](#). You can use a Resource Manager template, Azure PowerShell, or the Azure CLI to create a key vault.

WARNING

In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

Create a key vault with PowerShell

You can create a key vault with Azure PowerShell using the [New-AzureRmKeyVault](#) cmdlet. For additional cmdlets for Key Vault, see [AzureRM.KeyVault](#).

1. If needed, [connect to your Azure subscription](#).
2. Create a new resource group, if needed, with [New-AzureRmResourceGroup](#). To list data center locations, use [Get-AzureRmLocation](#).

```
# Get-AzureRmLocation  
New-AzureRmResourceGroup -Name 'MySecureRG' -Location 'East US'
```

3. Create a new key vault using [New-AzureRmKeyVault](#)

```
New-AzureRmKeyVault -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -Location 'East US'
```

4. Note the **Vault Name**, **Resource Group Name**, **Resource ID**, **Vault URI**, and the **Object ID** that are returned for later use when you encrypt the disks.

Create a key vault with Azure CLI

You can manage your key vault with Azure CLI using the [az keyvault](#) commands. To create a key vault, use [az keyvault create](#).

1. If needed, [connect to your Azure subscription](#).
2. Create a new resource group, if needed, with [az group create](#). To list locations, use [az account list-locations](#)

```
# To list locations: az account list-locations --output table  
az group create -n "MySecureRG" -l "East US"
```

3. Create a new key vault using [az keyvault create](#).

```
az keyvault create --name "MySecureVault" --resource-group "MySecureRG" --location "East US"
```

4. Note the **Vault Name** (name), **Resource Group Name**, **Resource ID** (ID), **Vault URI**, and the **Object ID**

that are returned for use later.

Create a key vault with a Resource Manager template

You can create a key vault by using the [Resource Manager template](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, Key Vault name, Object ID, legal terms, and agreement, and then click **Purchase**.

Set key vault advanced access policies

The Azure platform needs access to the encryption keys or secrets in your key vault to make them available to the VM for booting and decrypting the volumes. Enable disk encryption on the key vault or deployments will fail.

Set key vault advanced access policies with Azure PowerShell

Use the key vault PowerShell cmdlet [Set-AzureRmKeyVaultAccessPolicy](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** EnabledForDiskEncryption is required for Azure Disk encryption.

```
Set-AzureRmKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -  
EnabledForDiskEncryption
```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

```
Set-AzureRmKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -  
EnabledForDeployment
```

- **Enable Key Vault for template deployment, if needed:** Enables Azure Resource Manager to get secrets from this key vault when this key vault is referenced in a template deployment.

```
Set-AzureRmKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -  
EnabledForTemplateDeployment
```

Set key vault advanced access policies using the Azure CLI

Use [az keyvault update](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** Enabled-for-disk-encryption is required.

```
az keyvault update --name "MySecureVault" --resource-group "MySecureRG" --enabled-for-disk-encryption  
"true"
```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

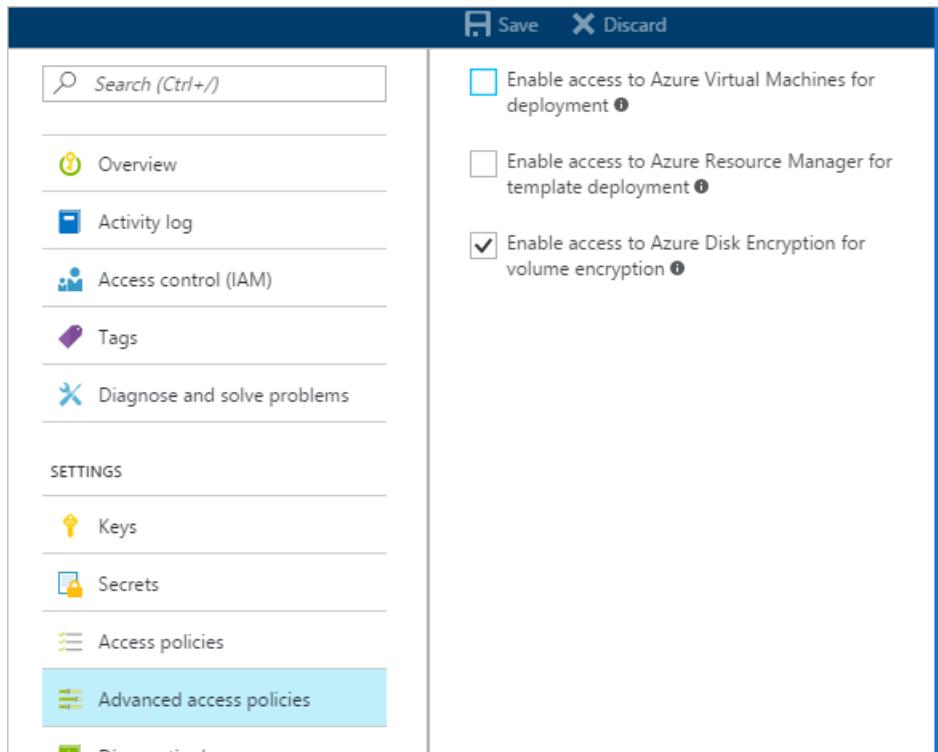
```
az keyvault update --name "MySecureVault" --resource-group "MySecureRG" --enabled-for-deployment  
"true"
```

- **Enable Key Vault for template deployment, if needed:** Allow Resource Manager to retrieve secrets from the vault.

```
az keyvault update --name "MySecureVault" --resource-group "MySecureRG" --enabled-for-template-deployment "true"
```

Set key vault advanced access policies through the Azure portal

1. Select your keyvault, go to **Access Policies**, and **Click to show advanced access policies**.
2. Select the box labeled **Enable access to Azure Disk Encryption for volume encryption**.
3. Select **Enable access to Azure Virtual Machines for deployment** and/or **Enable Access to Azure Resource Manager for template deployment**, if needed.
4. Click **Save**.



Set up a key encryption key (optional)

If you want to use a key encryption key (KEK) for an additional layer of security for encryption keys, add a KEK to your key vault. Use the [Add-AzureKeyVaultKey](#) cmdlet to create a key encryption key in the key vault. You can also import a KEK from your on-premises key management HSM. For more information, see [Key Vault Documentation](#). When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

- Your key vault secret and KEK URLs must be versioned. Azure enforces this restriction of versioning. For valid secret and KEK URLs, see the following examples:
 - Example of a valid secret URL:
<https://contosovault.vault.azure.net/secrets/EncryptionSecretWithKek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - Example of a valid KEK URL:
<https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
- Azure Disk Encryption doesn't support specifying port numbers as part of key vault secrets and KEK URLs. For examples of non-supported and supported key vault URLs, see the following examples:
 - Unacceptable key vault URL
<https://contosovault.vault.azure.net:443/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - Acceptable key vault URL:
<https://contosovault.vault.azure.net/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Set up a key encryption key with Azure PowerShell

Before using the PowerShell script, you should be familiar with the Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment. This script creates all Azure Disk Encryption prerequisites and encrypts an existing IaaS VM, wrapping the disk encryption key by using a key encryption key.

```
# Step 1: Create a new resource group and key vault in the same location.  
# Fill in 'MyLocation', 'MySecureRG', and 'MySecureVault' with your values.  
# Use Get-AzureRmLocation to get available locations and use the DisplayName.  
# To use an existing resource group, comment out the line for New-AzureRmResourceGroup  
  
$Loc = 'MyLocation';  
$rgname = 'MySecureRG';  
$KeyVaultName = 'MySecureVault';  
New-AzureRmResourceGroup -Name $rgname -Location $Loc;  
New-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $rgname -Location $Loc;  
$KeyVault = Get-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $rgname;  
$KeyVaultResourceId = (Get-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName  
$rgname).ResourceId;  
$diskEncryptionKeyVaultUrl = (Get-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName  
$rgname).VaultUri;  
  
#Step 2: Enable the vault for disk encryption.  
Set-AzureRmKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $rgname -  
EnabledForDiskEncryption;  
  
#Step 3: Create a new key in the key vault with the Add-AzureKeyVaultKey cmdlet.  
# Fill in 'MyKeyEncryptionKey' with your value.  
  
$keyEncryptionKeyName = 'MyKeyEncryptionKey';  
Add-AzureKeyVaultKey -VaultName $KeyVaultName -Name $keyEncryptionKeyName -Destination 'Software';  
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $KeyVaultName -Name  
$keyEncryptionKeyName).Key.kid;  
  
#Step 4: Encrypt the disks of an existing IaaS VM  
# Fill in 'MySecureVM' with your value.  
  
$VMName = 'MySecureVM';  
Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -  
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $KeyVaultResourceId -  
KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyId $KeyVaultResourceId;
```

Next steps

[Enable Azure Disk Encryption for Windows](#)

[Enable Azure Disk Encryption for Linux](#)

Enable Azure Disk Encryption for Windows IaaS VMs

12/12/2018 • 13 minutes to read • [Edit Online](#)

You can enable many disk-encryption scenarios, and the steps may vary according to the scenario. The following sections cover the scenarios in greater detail for Windows IaaS VMs. Before you can use disk encryption, the [Azure Disk Encryption prerequisites](#) need to be completed.

Take a [snapshot](#) and/or back up before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the `Set-AzureRmVMDiskEncryptionExtension` cmdlet to encrypt managed disks by specifying the `-skipVmBackup` parameter. For more information about how to back up and restore encrypted VMs, see the [Azure Backup](#) article.

WARNING

Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

Enable encryption on existing or running IaaS Windows VMs

In this scenario, you can enable encryption by using a template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail how to enable Azure Disk Encryption. If you need schema information for the virtual machine extension, see the [Azure Disk Encryption for Windows extension](#) article.

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the `Set-AzureRmVMDiskEncryptionExtension` cmdlet can be used to encrypt managed disks by specifying the `-skipVmBackup` parameter. The `Set-AzureRmVMDiskEncryptionExtension` command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Enable encryption on existing or running VMs with Azure PowerShell

Use the `Set-AzureRmVMDiskEncryptionExtension` cmdlet to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:** The script below initializes your variables and runs the `Set-AzureRmVMDiskEncryptionExtension` cmdlet. The resource group, VM, and key vault should have already been created as prerequisites. Replace `MySecureRg`, `MySecureVM`, and `MySecureVault` with your values.

```

$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId $keyVaultResourceId;

```

- **Encrypt a running VM using KEK:**

```

$rgName = 'MySecureRg';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId $keyVaultResourceId -
KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $keyVaultResourceId;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [Get-AzureRmVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName 'MySecureRg' -VMName 'MySecureVM'
```

- **Disable disk encryption:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet. Disabling data disk encryption on Windows VM when both OS and data disks have been encrypted doesn't work as expected. Disable encryption on all disks instead.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM'
```

Enable encryption on existing or running VMs with Azure CLI

Use the [az vm encryption enable](#) command to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault
"MySecureVault" --volume-type [All|OS|Data]
```

- **Encrypt a running VM using KEK:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [az vm encryption show](#) command.

```
az vm encryption show --name "MySecureVM" --resource-group "MySecureRg"
```

- **Disable encryption:** To disable encryption, use the [az vm encryption disable](#) command. Disabling data disk encryption on Windows VM when both OS and data disks have been encrypted doesn't work as expected. Disable encryption on all disks instead.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type [ALL, DATA, OS]
```

NOTE

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzureRmVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. This command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Using the Resource Manager template

You can enable disk encryption on existing or running IaaS Windows VMs in Azure by using the [Resource Manager template to encrypt a running Windows VM](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, location, settings, legal terms, and agreement. Click **Purchase** to enable encryption on the existing or running IaaS VM.

The following table lists the Resource Manager template parameters for existing or running VMs:

PARAMETER	DESCRIPTION
vmName	Name of the VM to run the encryption operation.

PARAMETER	DESCRIPTION
keyVaultName	Name of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet <code>(Get-AzureRmKeyVault -ResourceGroupName <MyResourceGroupName>).Vaultname</code> or the Azure CLI command `az keyvault list --resource-group "MySecureGroup"
keyVaultResourceGroup	Name of the resource group that contains the key vault
keyEncryptionKeyURL	URL of the key encryption key that's used to encrypt the generated BitLocker key. This parameter is optional if you select nokek in the UseExistingKek drop-down list. If you select kek in the UseExistingKek drop-down list, you must enter the <i>keyEncryptionKeyURL</i> value.
volumeType	Type of volume that the encryption operation is performed on. Valid values are <i>OS</i> , <i>Data</i> , and <i>All</i> .
forceUpdateTag	Pass in a unique value like a GUID every time the operation needs to be force run.
resizeOSDisk	Should the OS partition be resized to occupy full OS VHD before splitting system volume.
location	Location for all resources.

Encrypt virtual machine scale sets

Azure virtual machine scale sets let you create and manage a group of identical, load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Use the CLI or Azure PowerShell to encrypt virtual machine scale sets.

Register for disk encryption preview using Azure Powershell

The Azure disk encryption for virtual machine scale sets preview requires you to self-register your subscription with [Register-AzureRmProviderFeature](#). You only need to perform the following steps the first time that you use the disk encryption preview feature:

```
Register-AzureRmProviderFeature -ProviderNamespace Microsoft.Compute -FeatureName "UnifiedDiskEncryption"
```

It can take up to 10 minutes for the registration request to propagate. You can check on the registration state with [Get-AzureRmProviderFeature](#). When the `RegistrationState` reports *Registered*, re-register the *Microsoft.Compute* provider with [Register-AzureRmResourceProvider](#):

```
Get-AzureRmProviderFeature -ProviderNamespace "Microsoft.Compute" -FeatureName "UnifiedDiskEncryption"
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.Compute
```

Encrypt virtual machine scale sets with Azure PowerShell

Use the [Set-AzureRmVmssDiskEncryptionExtension](#) cmdlet to enable encryption on a Windows virtual machine scale set. The resource group, VM, and key vault should have already been created as prerequisites.

- **Encrypt a running virtual machine scale set:**

```
$rgName= "MySecureRg";
$VmssName = "MySecureVmss";
$keyVaultName= "MySecureVault";
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgName;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
Set-AzureRmVmssDiskEncryptionExtension -ResourceGroupName $rgName -VMScaleSetName $VmssName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId;
```

- **Encrypt a running virtual machine scale set using KEK to wrap the key:**

```
$rgName= "MySecureRg";
$VmssName = "MySecureVmss";
$keyVaultName= "MySecureVault";
$keyEncryptionKeyName = "MyKeyEncryptionKey";
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgName;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
Set-AzureRmVmssDiskEncryptionExtension -ResourceGroupName $rgName -VMScaleSetName $VmssName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -
KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyId $keyVaultResourceId;
```

- **Get encryption status for a virtual machine scale set:** Use the [Get-AzureRmVmssVMDiskEncryption](#) cmdlet.

```
get-AzureRmVmssVMDiskEncryption -ResourceGroupName "MySecureRG" -VMScaleSetName "MySecureVmss"
```

- **Disable encryption on a virtual machine scale set:** Use the [Disable-AzureRmVmssDiskEncryption](#) cmdlet.

```
Disable-AzureRmVmssDiskEncryption -ResourceGroupName "MySecureRG" -VMScaleSetName "MySecureVmss"
```

Register for disk encryption preview using Azure CLI

The Azure disk encryption for virtual machine scale sets preview requires you to self-register your subscription with [az feature register](#). You only need to perform the following steps the first time that you use the disk encryption preview feature:

```
az feature register --name UnifiedDiskEncryption --namespace Microsoft.Compute
```

It can take up to 10 minutes for the registration request to propagate. You can check on the registration state with [az feature show](#). When the `State` reports *Registered*, re-register the *Microsoft.Compute* provider with [az provider register](#):

```
az provider register --namespace Microsoft.Compute
```

Encrypt virtual machine scale sets with Azure CLI

Use the [az vmss encryption enable](#) to enable encryption on a Windows virtual machine scale set. If you set the upgrade policy on the scale set to manual, start the encryption with [az vmss update-instances](#). The resource group, VM, and key vault should have already been created as prerequisites.

- **Encrypt a running virtual machine scale set**

```
az vmss encryption enable --resource-group "MySecureRG" --name "MySecureVmss" --disk-encryption-keyvault "MySecureVault"
```

- **Encrypt a running virtual machine scale set using KEK to wrap the key**

```
az vmss encryption enable --resource-group "MySecureRG" --name "MySecureVmss" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK" --key-encryption-keyvault "MySecureVault"
```

- **Get encryption status for a virtual machine scale set:** Use [az vmss encryption show](#)

```
az vmss encryption show --resource-group "MySecureRG" --name "MySecureVmss"
```

- **Disable encryption on a virtual machine scale set:** Use [az vmss encryption disable](#)

```
az vmss encryption disable --resource-group "MySecureRG" --name "MySecureVmss"
```

Azure Resource Manager templates for Windows virtual machine scale sets

To encrypt or decrypt Windows virtual machine scale sets, use the Azure Resource Manager templates and instructions below:

- [Enable encryption on a Windows virtual machine scale set](#)
- [Deploy a VM scale set of Windows VMs with a jumpbox and enable encryption on the Windows VM scale set](#)
- [Disable encryption on a Windows VM scale set](#)
 1. Click **Deploy to Azure**.
 2. Fill in the required fields then agree to the terms and conditions.
 3. Click **Purchase** to deploy the template.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using PowerShell cmdlets or CLI commands.

Use the instructions in the appendix for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- [Prepare a pre-encrypted Windows VHD](#)
- [Prepare a pre-encrypted Linux VHD](#)

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzureRmVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. The Set-AzureRmVMDiskEncryptionExtension command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Encrypt VMs with pre-encrypted VHDS with Azure PowerShell

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet [Set-AzureRmVMOSDisk](#).

The example below gives you some common parameters.

```
$VirtualMachine = New-AzureRmVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzureRmVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri "os.vhd" Caching
ReadWrite -Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa" -DiskEncryptionKeyVaultId
"/subscriptions/00000000-0000-0000-0000-
0000000000/resourceGroups/myresourcegroup/providers/Microsoft.KeyVault/vaults/mytestvault"
New-AzureRmVM -VM $VirtualMachine -ResourceGroupName "MySecureRG"
```

Enable encryption on a newly added data disk

You can [add a new disk to a Windows VM using PowerShell](#), or [through the Azure portal](#).

Enable encryption on a newly added disk with Azure PowerShell

When using Powershell to encrypt a new disk for Windows VMs, a new sequence version should be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version. In some cases, a newly added data disk might be encrypted automatically by the Azure Disk Encryption extension. Auto encryption usually occurs when the VM reboots after the new disk comes online. This is typically caused because "All" was specified for the volume type when disk encryption previously ran on the VM. If auto encryption occurs on a newly added data disk, we recommend running the Set-AzureRmVmDiskEncryptionExtension cmdlet again with new sequence version. If your new data disk is auto encrypted and you do not wish to be encrypted, decrypt all drives first then re-encrypt with a new sequence version specifying OS for the volume type.

- **Encrypt a running VM:** The script below initializes your variables and runs the Set-AzureRmVmDiskEncryptionExtension cmdlet. The resource group, VM, and key vault should have already been created as prerequisites. Replace MySecureRg, MySecureVM, and MySecureVault with your values. This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

```
$sequenceVersion = [Guid]::.NewGuid();
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVmDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId $keyVaultResourceId -
VolumeType "All" -SequenceVersion $sequenceVersion;
```

- **Encrypt a running VM using KEK:** This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

```

$sequenceVersion = [Guid]::NewGuid();
$rgName = 'MySecureRg';
$vmName = 'MyExtraSecureVM';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -
KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $keyVaultResourceId -VolumeType "All"
-SequenceVersion $sequenceVersion;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

Enable encryption on a newly added disk with Azure CLI

The Azure CLI command will automatically provide a new sequence version for you when you run the command to enable encryption. The example uses "All" for the volume-type parameter. You may need to change the volume-type parameter to OS if you're only encrypting the OS disk. In contrast to Powershell syntax, the CLI does not require the user to provide a unique sequence version when enabling encryption. The CLI automatically generates and uses its own unique sequence version value.

- **Encrypt a running VM:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault
"MySecureVault" --volume-type "All"
```

- **Encrypt a running VM using KEK:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault
"MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault
"MySecureVaultContainingTheKEK" --volume-type "All"
```

Disable encryption

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template. Disabling data disk encryption on Windows VM when both OS and data disks have been encrypted doesn't work as expected. Disable encryption on all disks instead.

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM' -VolumeType "all"
```

- **Disable encryption with the Azure CLI:** To disable encryption, use the [az vm encryption disable](#)

command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type "all"
```

- **Disable encryption with a Resource Manager template:**

1. Click **Deploy to Azure** from the [Disable disk encryption on running Windows VM](#) template.
2. Select the subscription, resource group, location, VM, volume type, legal terms, and agreement.
3. Click **Purchase** to disable disk encryption on a running Windows VM.

Next steps

[Enable Azure Disk Encryption for Linux](#)

Enable Azure Disk Encryption for Linux IaaS VMs

12/12/2018 • 16 minutes to read • [Edit Online](#)

You can enable many disk-encryption scenarios, and the steps may vary according to the scenario. The following sections cover the scenarios in greater detail for Linux IaaS VMs. Before you can use disk encryption, the [Azure Disk Encryption prerequisites](#) need to be completed and the [Additional prerequisites for Linux IaaS VMs](#) section should be reviewed.

Take a [snapshot](#) and/or back up before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the `Set-AzureRmVMDiskEncryptionExtension` cmdlet to encrypt managed disks by specifying the `-skipVmBackup` parameter. For more information about how to back up and restore encrypted VMs, see the [Azure Backup](#) article.

WARNING

Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

When encrypting Linux OS volumes, the process can take a few hours. It is normal for Linux OS volumes to take longer than data volumes to encrypt.

Disabling encryption on Linux VMs is only supported for data volumes. It is not supported on data or OS volumes if the OS volume has been encrypted.

Enable encryption on an existing or running IaaS Linux VM

In this scenario, you can enable encryption by using the Resource Manager template, PowerShell cmdlets, or CLI commands. If you need schema information for the virtual machine extension, see the [Azure Disk Encryption for Linux extension](#) article.

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the `Set-AzureRmVMDiskEncryptionExtension` cmdlet can be used to encrypt managed disks by specifying the `-skipVmBackup` parameter. The `Set-AzureRmVMDiskEncryptionExtension` command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Enable encryption on an existing or running Linux VM using Azure CLI

You can enable disk encryption on your encrypted VHD by installing and using The [Azure CLI 2.0](#) command-line tool. You can use it in your browser with [Azure Cloud Shell](#), or you can install it on your local machine and use it in any PowerShell session. To enable encryption on existing or running IaaS Linux VMs in Azure, use the following CLI commands:

Use the `az vm encryption enable` command to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type [All|OS|Data]
```

- **Encrypt a running VM using KEK:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [az vm encryption show](#) command.

```
az vm encryption show --name "MySecureVM" --resource-group "MySecureRg"
```

- **Disable encryption:** To disable encryption, use the [az vm encryption disable](#) command. Disabling encryption is only allowed on data volumes for Linux VMs.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type DATA
```

Enable encryption on an existing or running Linux VM using PowerShell

Use the [Set-AzureRmVMDiskEncryptionExtension](#) cmdlet to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:** The script below initializes your variables and runs the Set-AzureRmVMDiskEncryptionExtension cmdlet. The resource group, VM, and key vault, should have already been created as prerequisites. Replace MySecureRg, MySecureVM, and MySecureVault with your values. You may need to add the -VolumeType parameter if you're encrypting data disks and not the OS disk.

```
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId $keyVaultResourceId;
```

- **Encrypt a running VM using KEK:** You may need to add the -VolumeType parameter if you're encrypting data disks and not the OS disk.

```

$rgName = 'MySecureRg';
$vmName = 'MyExtraSecureVM';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -
KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyId $keyVaultResourceId;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-
name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-
name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [Get-AzureRmVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName MySecureRg -VMName MySecureVM
```

- **Disable disk encryption:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet. Disabling encryption is only allowed on data volumes for Linux VMs.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM'
```

Enable encryption on an existing or running IaaS Linux VM with a template

You can enable disk encryption on an existing or running IaaS Linux VM in Azure by using the [Resource Manager template](#).

1. Click **Deploy to Azure** on the Azure quickstart template.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Create** to enable encryption on the existing or running IaaS VM.

The following table lists Resource Manager template parameters for existing or running VMs:

PARAMETER	DESCRIPTION
vmName	Name of the VM to run the encryption operation.
keyVaultName	Name of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet <code>(Get-AzureRmKeyVault -ResourceGroupName <MyResourceGroupName>).Vaultname</code> or the Azure CLI command `az keyvault list --resource-group "MySecureGroup"
keyVaultResourceGroup	Name of the resource group that contains the key vault

PARAMETER	DESCRIPTION
keyEncryptionKeyURL	URL of the key encryption key that's used to encrypt the generated BitLocker key. This parameter is optional if you select nokek in the UseExistingKek drop-down list. If you select kek in the UseExistingKek drop-down list, you must enter the <i>keyEncryptionKeyURL</i> value.
volumeType	Type of volume that the encryption operation is performed on. Valid values are <i>OS</i> , <i>Data</i> , and <i>All</i> .
forceUpdateTag	Pass in a unique value like a GUID every time the operation needs to be force run.
resizeOSDisk	Should the OS partition be resized to occupy full OS VHD before splitting system volume.
location	Location for all resources.

Encrypt virtual machine scale sets

Azure [virtual machine scale sets](#) let you create and manage a group of identical, load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Use the CLI or Azure PowerShell to encrypt virtual machine scale sets.

A batch file example for Linux scale set data disk encryption can be found [here](#). This example creates a resource group, Linux scale set, mounts a 5-GB data disk, and encrypts the virtual machine scale set.

Register for disk encryption preview using Azure CLI

The Azure disk encryption for virtual machine scale sets preview requires you to self-register your subscription with [az feature register](#). You only need to perform the following steps the first time that you use the disk encryption preview feature:

```
az feature register --name UnifiedDiskEncryption --namespace Microsoft.Compute
```

It can take up to 10 minutes for the registration request to propagate. You can check on the registration state with [az feature show](#). When the `State` reports *Registered*, re-register the *Microsoft.Compute* provider with [az provider register](#):

```
az provider register --namespace Microsoft.Compute
```

Encrypt virtual machine scale sets with Azure CLI

Use the [az vmss encryption enable](#) to enable encryption on a Windows virtual machine scale set. If you set the upgrade policy on the scale set to manual, start the encryption with [az vmss update-instances](#). The resource group, VM, and key vault should have already been created as prerequisites.

- **Encrypt a running virtual machine scale set**

```
az vmss encryption enable --resource-group "MySecureRG" --name "MySecureVmss" --disk-encryption-keyvault "MySecureVault"
```

- **Encrypt a running virtual machine scale set using KEK to wrap the key**

```
az vmss encryption enable --resource-group "MySecureRG" --name "MySecureVmss" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK" --key-encryption-keyvault "MySecureVault"
```

- **Get encryption status for a virtual machine scale set:** Use [az vmss encryption show](#)

```
az vmss encryption show --resource-group "MySecureRG" --name "MySecureVmss"
```

- **Disable encryption on a virtual machine scale set:** Use [az vmss encryption disable](#)

```
az vmss encryption disable --resource-group "MySecureRG" --name "MySecureVmss"
```

Register for disk encryption preview using Azure Powershell

The Azure disk encryption for virtual machine scale sets preview requires you to self-register your subscription with [Register-AzureRmProviderFeature](#). You only need to perform the following steps the first time that you use the disk encryption preview feature:

```
Register-AzureRmProviderFeature -ProviderNamespace Microsoft.Compute -FeatureName "UnifiedDiskEncryption"
```

It can take up to 10 minutes for the registration request to propagate. You can check on the registration state with [Get-AzureRmProviderFeature](#). When the `RegistrationState` reports *Registered*, re-register the *Microsoft.Compute* provider with [Register-AzureRmResourceProvider](#):

```
Get-AzureRmProviderFeature -ProviderNamespace "Microsoft.Compute" -FeatureName "UnifiedDiskEncryption"
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.Compute
```

Encrypt virtual machine scale sets with Azure PowerShell

Use the [Set-AzureRmVmssDiskEncryptionExtension](#) cmdlet to enable encryption on a Windows virtual machine scale set. The resource group, VM, and key vault should have already been created as prerequisites.

- **Encrypt a running virtual machine scale set:**

```
$rgName= "MySecureRg";
$VmssName = "MySecureVmss";
$keyVaultName= "MySecureVault";
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgName;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
Set-AzureRmVmssDiskEncryptionExtension -ResourceGroupName $rgName -VMScaleSetName $VmssName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId;
```

- **Encrypt a running virtual machine scale set using KEK to wrap the key:**

```
$rgName= "MySecureRg";
$VmssName = "MySecureVmss";
$keyVaultName= "MySecureVault";
$keyEncryptionKeyName = "MyKeyEncryptionKey";
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgName;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
Set-AzureRmVmssDiskEncryptionExtension -ResourceGroupName $rgName -VMScaleSetName $VmssName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -
KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $keyVaultResourceId;
```

- **Get encryption status for a virtual machine scale set:** Use the [Get-AzureRmVmssVMDiskEncryption](#) cmdlet.

```
get-AzureRmVmssVMDiskEncryption -ResourceGroupName "MySecureRG" -VMSScaleSetName "MySecureVmss"
```

- **Disable encryption on a virtual machine scale set:** Use the [Disable-AzureRmVmssDiskEncryption](#) cmdlet.

```
Disable-AzureRmVmssDiskEncryption -ResourceGroupName "MySecureRG" -VMSScaleSetName "MySecureVmss"
```

Azure Resource Manager templates for Linux virtual machine scale sets

To encrypt or decrypt Linux virtual machine scale sets, use the Azure Resource Manager templates and instructions below:

- [Enable encryption on a Linux virtual machine scale set](#)
- [Deploy a VM scale set of Linux VMs with a jumpbox and enable encryption on the Linux VM scale set](#)
- [Disable encryption on a Linux VM scale set](#)
 1. Click **Deploy to Azure**.
 2. Fill in the required fields then agree to the terms and conditions.
 3. Click **Purchase** to deploy the template.

Use EncryptFormatAll feature for data disks on Linux IaaS VMs

The **EncryptFormatAll** parameter reduces the time for Linux data disks to be encrypted. Partitions meeting certain criteria will be formatted (with its current file system). Then they'll be remounted back to where it was before command execution. If you wish to exclude a data disk that meets the criteria, you can unmount it before running the command.

After running this command, any drives that were mounted previously will be reformatted. Then the encryption layer will be started on top of the now empty drive. When this option is selected, the ephemeral resource disk attached to the VM will also be encrypted. If the ephemeral drive is reset, it will be reformatted and re-encrypted for the VM by the Azure Disk Encryption solution at the next opportunity.

WARNING

EncryptFormatAll shouldn't be used when there is needed data on a VM's data volumes. You may exclude disks from encryption by unmounting them. You should first try out the EncryptFormatAll first on a test VM, understand the feature parameter and its implication before trying it on the production VM. The EncryptFormatAll option formats the data disk and all the data on it will be lost. Before proceeding, verify that disks you wish to exclude are properly unmounted.

If you're setting this parameter while updating encryption settings, it might lead to a reboot before the actual encryption. In this case, you will also want to remove the disk you don't want formatted from the fstab file. Similarly, you should add the partition you want encrypt-formatted to the fstab file before initiating the encryption operation.

EncryptFormatAll criteria

The parameter goes through all partitions and encrypts them as long as they meet **all** of the criteria below:

- Is not a root/OS/boot partition
- Is not already encrypted
- Is not a BEK volume
- Is not a RAID volume
- Is not an LVM volume

- Is mounted

Encrypt the disks that compose the RAID or LVM volume rather than the RAID or LVM volume.

Use the `EncryptFormatAll` parameter with Azure CLI

Use the [az vm encryption enable](#) command to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM using `EncryptFormatAll`:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault
"MySecureVault" --encrypt-format-all
```

Use the `EncryptFormatAll` parameter with a PowerShell cmdlet

Use the [Set-AzureRmVMDiskEncryptionExtension](#) cmdlet with the `EncryptFormatAll` parameter.

Encrypt a running VM using `EncryptFormatAll`: As an example, the script below initializes your variables and runs the `Set-AzureRmVMDiskEncryptionExtension` cmdlet with the `EncryptFormatAll` parameter. The resource group, VM, and key vault should have already been created as prerequisites. Replace `MySecureRg`, `MySecureVM`, and `MySecureVault` with your values.

```
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -DiskEncryptionKeyVaultUrl
$diskEncryptionKeyVaultUrl -DiskEncryptionKeyId $keyVaultResourceId -EncryptFormatAll
```

Use the `EncryptFormatAll` parameter with Logical Volume Manager (LVM)

We recommend an LVM-on-crypt setup. For all the following examples, replace the device-path and mountpoints with whatever suits your use-case. This setup can be done as follows:

- Add the data disks that will compose the VM.
- Format, mount, and add these disks to the fstab file.

1. Format the newly added disk. We use symlinks generated by Azure here. Using symlinks avoids problems related to device names changing. For more information, see the [Troubleshoot Device Names problems](#) article.

```
mkfs -t ext4 /dev/disk/azure/scsi1/lun0
```

2. Mount the disks.

```
mount /dev/disk/azure/scsi1/lun0 /mnt/mountpoint
```

3. Add to fstab.

```
echo "/dev/disk/azure/scsi1/lun0 /mnt/mountpoint ext4 defaults,nofail 1 2" >> /etc/fstab
```

4. Run the `Set-AzureRmVMDiskEncryptionExtension` PowerShell cmdlet with `-EncryptFormatAll` to encrypt these disks.

```
Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName "MySecureGroup" -VMName "MySecureVM" -
DiskEncryptionKeyVaultUrl "https://mykeyvault.vault.azure.net/" -EncryptFormatAll
```

- Set up LVM on top of these new disks. Note the encrypted drives are unlocked after the VM has finished booting. So, the LVM mounting will also have to be subsequently delayed.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using PowerShell cmdlets or CLI commands.

Use the instructions in the appendix for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- [Prepare a pre-encrypted Windows VHD](#)
- [Prepare a pre-encrypted Linux VHD](#)

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzureRmVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. The Set-AzureRmVMDiskEncryptionExtension command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Use Azure PowerShell to encrypt IaaS VMs with pre-encrypted VHDS

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet [Set-AzureRmVMOSDisk](#). The example below gives you some common parameters.

```
$VirtualMachine = New-AzureRmVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzureRmVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri "os.vhd" Caching
ReadWrite -Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa" -DiskEncryptionKeyVaultId
"/subscriptions/00000000-0000-0000-0000-
0000000000/resourceGroups/myresourcegroup/providers/Microsoft.KeyVault/vaults/mytestvault"
New-AzureRmVM -VM $VirtualMachine -ResourceGroupName "MySecureRG"
```

Enable encryption on a newly added data disk

You can add a new data disk using [az vm disk attach](#), or [through the Azure portal](#). Before you can encrypt, you need to mount the newly attached data disk first. You must request encryption of the data drive since the drive will be unusable while encryption is in progress.

Enable encryption on a newly added disk with Azure CLI

If the VM was previously encrypted with "All" then the --volume-type parameter should remain All. All includes both OS and data disks. If the VM was previously encrypted with a volume type of "OS", then the --volume-type parameter should be changed to All so that both the OS and the new data disk will be included. If the VM was encrypted with only the volume type of "Data", then it can remain "Data" as demonstrated below. Adding and attaching a new data disk to a VM is not sufficient preparation for encryption. The newly attached disk must also be formatted and properly mounted within the VM prior to enabling encryption. On Linux the disk must be mounted in /etc/fstab with a [persistent block device name](#).

In contrast to Powershell syntax, the CLI does not require the user to provide a unique sequence version when enabling encryption. The CLI automatically generates and uses its own unique sequence version value.

- **Encrypt data volumes of a running VM:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault
"MySecureVault" --volume-type "Data"
```

- **Encrypt data volumes of a running VM using KEK:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --disk-encryption-keyvault
"MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault
"MySecureVaultContainingTheKEK" --volume-type "Data"
```

Enable encryption on a newly added disk with Azure PowerShell

When using Powershell to encrypt a new disk for Linux, a new sequence version needs to be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version.

- **Encrypt data volumes of a running VM:** The script below initializes your variables and runs the Set-AzureRmVMDiskEncryptionExtension cmdlet. The resource group, VM, and key vault should have already been created as prerequisites. Replace MySecureRg, MySecureVM, and MySecureVault with your values. Acceptable values for the -VolumeType parameter are All, OS, and Data. If the VM was previously encrypted with a volume type of "OS" or "All", then the -VolumeType parameter should be changed to All so that both the OS and the new data disk will be included.

```
$sequenceVersion = [Guid]::NewGuid();
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId $keyVaultResourceId -
VolumeType 'data' -SequenceVersion $sequenceVersion;
```

- **Encrypt data volumes of a running VM using KEK:** Acceptable values for the -VolumeType parameter are All, OS, and Data. If the VM was previously encrypted with a volume type of "OS" or "All", then the -VolumeType parameter should be changed to All so that both the OS and the new data disk will be included.

```
$rgName = 'MySecureRg';
$vmName = 'MyExtraSecureVM';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId $keyVaultResourceId -
KeyEncryptionKeyUrl $keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $keyVaultResourceId -VolumeType
'data';
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

Disable encryption for Linux VMs

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template.

IMPORTANT

Disabling encryption with Azure Disk Encryption on Linux VMs is only supported for data volumes. It is not supported on data or OS volumes if the OS volume has been encrypted.

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM' [--volume-type {ALL, DATA, OS}]
```

- **Disable encryption with the Azure CLI:** To disable encryption, use the [az vm encryption disable](#) command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type [ALL, DATA, OS]
```

- **Disable encryption with a Resource Manager template:** Use the [Disable encryption on a running Linux VM](#) template to disable encryption.
 1. Click **Deploy to Azure**.
 2. Select the subscription, resource group, location, VM, legal terms, and agreement.
 3. Click **Purchase** to disable disk encryption on a running Windows VM.

Next steps

[Enable Azure Disk Encryption for Windows](#)

Appendix for Azure Disk Encryption

12/12/2018 • 18 minutes to read • [Edit Online](#)

This article is an appendix to [Azure Disk Encryption for IaaS VMs](#). Make sure you read the Azure Disk Encryption for IaaS VMs articles first to understand the context. This article describes how to prepare pre-encrypted VHDs and other tasks.

Connect to your subscription

Before you start, review the [Prerequisites](#) article. After all the prerequisites have been met, connect to your subscription by running the following cmdlets:

Connect to your subscription with PowerShell

1. Start an Azure PowerShell session, and sign in to your Azure account with the following command:

```
Connect-AzureRmAccount
```

2. If you have multiple subscriptions and want to specify one to use, type the following to see the subscriptions for your account:

```
Get-AzureRmSubscription
```

3. To specify the subscription you want to use, type:

```
Select-AzureRmSubscription -SubscriptionName <Yoursubscriptionname>
```

4. To verify that the subscription configured is correct, type:

```
Get-AzureRmSubscription
```

5. If needed, connect to Azure AD with [Connect-AzureAD](#).

```
Connect-AzureAD
```

6. To confirm the Azure Disk Encryption cmdlets are installed, type:

```
Get-command *diskencryption*
```

7. Review [Getting started with Azure PowerShell](#) and [AzureAD](#), if needed.

Connect to your subscription with the Azure CLI

1. Sign in to Azure with [az login](#).

```
az login
```

2. If you would like to select a tenant to sign in under, use:

```
az login --tenant <tenant>
```

3. If you have multiple subscriptions and want to specify a specific one, get your subscription list with [az account list](#) and specify with [az account set](#).

```
az account list  
az account set --subscription "<subscription name or ID>"
```

4. Verify the installed version.

```
az --version
```

5. Review [Get started with Azure CLI 2.0](#) if needed.

Sample PowerShell scripts for Azure Disk Encryption

- **List all encrypted VMs in your subscription**

```
$osVolEncrypted = {(Get-AzureRmVMDiskEncryptionStatus -ResourceGroupName $_.ResourceGroupName -VMName  
$_.Name).OsVolumeEncrypted}  
$dataVolEncrypted= {(Get-AzureRmVMDiskEncryptionStatus -ResourceGroupName $_.ResourceGroupName -VMName  
$_.Name).DataVolumesEncrypted}  
Get-AzureRmVm | Format-Table @{Label="MachineName"; Expression={$_.Name}}, @{Label="OsVolumeEncrypted";  
Expression=$osVolEncrypted}, @{Label="DataVolumesEncrypted"; Expression=$dataVolEncrypted}
```

- **List all disk encryption secrets used for encrypting VMs in a key vault**

```
Get-AzureKeyVaultSecret -VaultName $KeyVaultName | where  
{$_._Tags.ContainsKey('DiskEncryptionKeyFileName')} | format-table @{Label="MachineName"; Expression=  
{$_._Tags['MachineName']}}, @{Label="VolumeLetter"; Expression={$_._Tags['VolumeLetter']}},  
@{Label="EncryptionKeyURL"; Expression={$_._Id}}
```

Using the Azure Disk Encryption prerequisites PowerShell script

If you're already familiar with the prerequisites for Azure Disk Encryption, you can use the [Azure Disk Encryption prerequisites PowerShell script](#). For an example of using this PowerShell script, see the [Encrypt a VM Quickstart](#). You can remove the comments from a section of the script, starting at line 211, to encrypt all disks for existing VMs in an existing resource group.

The following table shows which parameters can be used in the PowerShell script:

PARAMETER	DESCRIPTION	IS MANDATORY
\$resourceGroupName	Name of the resource group to which the KeyVault belongs to. A new resource group with this name will be created if one doesn't exist.	True
\$keyVaultName	Name of the KeyVault in which encryption keys are to be placed. A new vault with this name will be created if one doesn't exist.	True

PARAMETER	DESCRIPTION	IS MANDATORY
\$location	Location of the KeyVault. Make sure the KeyVault and VMs to be encrypted are in the same location. Get a location list with <code>Get-AzureRMLocation</code> .	True
\$subscriptionId	Identifier of the Azure subscription to be used. You can get your Subscription ID with <code>Get-AzureRMSubscription</code> .	True
\$aadAppName	Name of the Azure AD application that will be used to write secrets to KeyVault. A new application with this name will be created if one doesn't exist. If this app already exists, pass <code>aadClientSecret</code> parameter to the script.	False
\$aadClientSecret	Client secret of the Azure AD application that was created earlier.	False
\$keyEncryptionKeyName	Name of optional key encryption key in KeyVault. A new key with this name will be created if one doesn't exist.	False

Resource Manager templates

Encrypt or decrypt VMs without an Azure AD app

- [Enable disk encryption on existing or running IaaS Windows VMs](#)
- [Disable disk encryption on existing or running IaaS Windows VMs](#)
- [Enable disk encryption on an existing or running IaaS Linux VM](#)
 - [Disable encryption on a running Linux VM](#)
 - Disabling encryption is only allowed on Data volumes for Linux VMs.

Encrypt or decrypt VM scale sets

- [Enable disk encryption on a running Linux virtual machine scale set](#)
- [Enable disk encryption on a running Windows virtual machine scale set](#)
 - [Deploy a VM Scale Set of Linux VMs with a jumpbox and enables encryption on Linux VMSS](#)
 - [Deploy a VM Scale Set of Windows VMs with a jumpbox and enables encryption on Windows VMSS](#)
- [Disable disk encryption on a running Linux virtual machine scale set](#)
- [Disable disk encryption on a running Windows virtual machine scale set](#)

Encrypt or decrypt VMs with an Azure AD app (previous release)

- [Enable disk encryption on existing or running IaaS Windows VMs](#)
- [Enable disk encryption on an existing or running IaaS Linux VM](#)
- [Disable disk encryption on running Windows IaaS](#)
- [Disable encryption on a running Linux VM](#)
 - Disabling encryption is only allowed on Data volumes for Linux VMs.

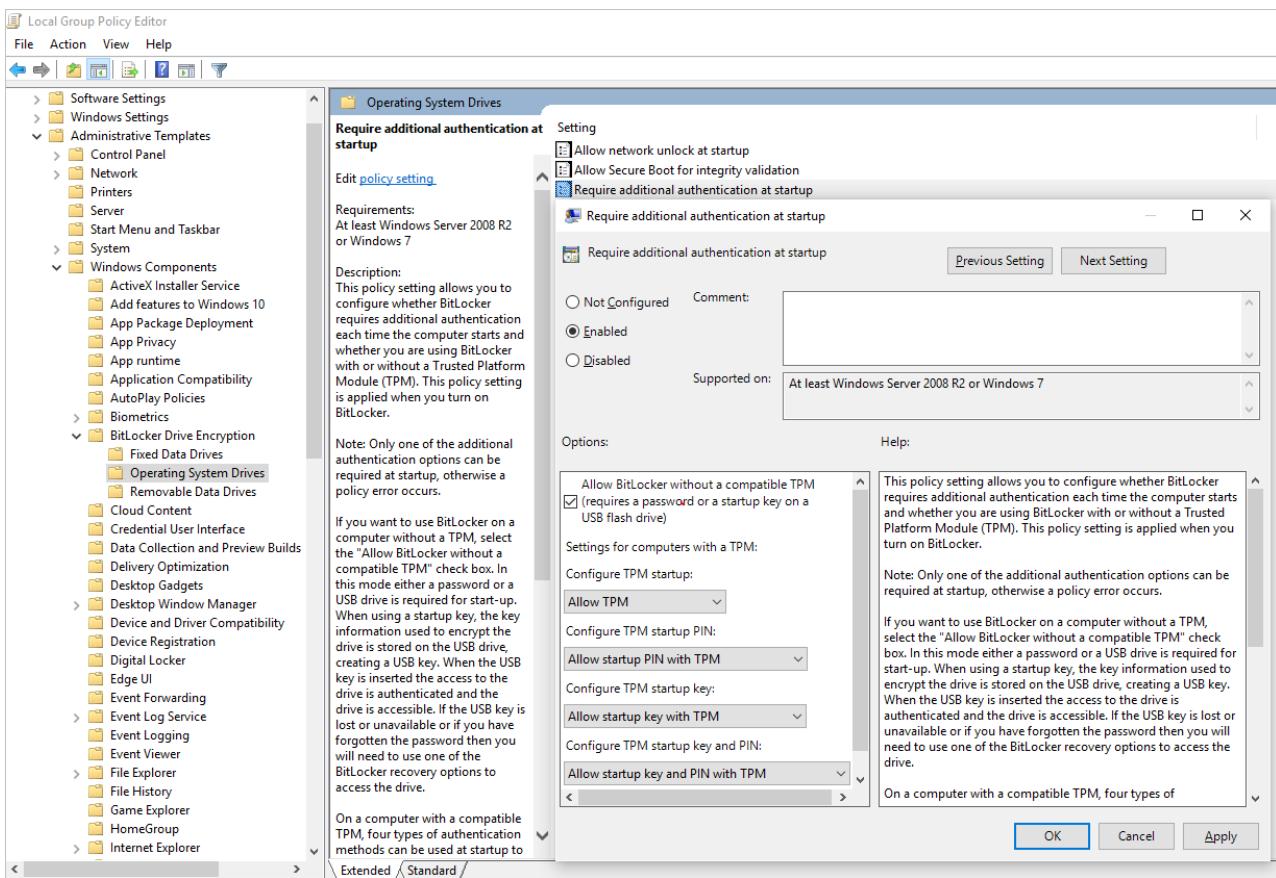
- [Enable disk encryption on new IaaS Windows VM from the Marketplace](#)
 - This template creates a new encrypted Windows VM that uses the Windows Server 2012 gallery image.
- [Create a new encrypted Windows IaaS Managed Disk VM from gallery image](#)
 - This template creates a new encrypted Windows VM with managed disks using the Windows Server 2012 gallery image.
- [Deployment of RHEL 7.2 with full disk encryption with managed disks](#)
 - This template creates a fully encrypted RHEL 7.2 VM in Azure using managed disks. It includes an encrypted 30-GB OS drive and an encrypted 200-GB array (RAID-0) mounted at /mnt/raidencrypted. See the [FAQ](#) article for supported Linux server distributions.
- [Deployment of RHEL 7.2 with full disk encryption with unmanaged disks](#)
 - This template creates a fully encrypted RHEL 7.2 VM in Azure with an encrypted 30-GB OS drive and an encrypted 200-GB array (RAID-0) mounted at /mnt/raidencrypted. See the [FAQ](#) article for supported Linux server distributions.
- [Enable disk encryption on a pre-encrypted VHD for Windows or Linux](#)
- [Create a new encrypted managed disk from a pre-encrypted VHD/storage blob](#)
 - Creates a new encrypted managed disk provided a pre-encrypted VHD and its corresponding encryption settings
- [Enable disk encryption on a running Windows VM using an Azure AD client certificate thumbprint](#)

Prepare a pre-encrypted Windows VHD

The sections that follow are necessary to prepare a pre-encrypted Windows VHD for deployment as an encrypted VHD in Azure IaaS. Use the information to prepare and boot a fresh Windows VM (VHD) on Azure Site Recovery or Azure. For more information on how to prepare and upload a VHD, see [Upload a generalized VHD and use it to create new VMs in Azure](#).

Update group policy to allow non-TPM for OS protection

Configure the BitLocker Group Policy setting **BitLocker Drive Encryption**, which you'll find under **Local Computer Policy > Computer Configuration > Administrative Templates > Windows Components**. Change this setting to **Operating System Drives > Require additional authentication at startup > Allow BitLocker without a compatible TPM**, as shown in the following figure:



Install BitLocker feature components

For Windows Server 2012 and later, use the following command:

```
dism /online /Enable-Feature /all /FeatureName:BitLocker /quiet /norestart
```

For Windows Server 2008 R2, use the following command:

```
ServerManagerCmd -install BitLockers
```

Prepare the OS volume for BitLocker by using [bdehdcfg](#)

To compress the OS partition and prepare the machine for BitLocker, execute the [bdehdcfg](#) if needed:

```
bdehdcfg -target c: shrink -quiet
```

Protect the OS volume by using BitLocker

Use the [manage-bde](#) command to enable encryption on the boot volume using an external key protector. Also place the external key (.bek file) on the external drive or volume. Encryption is enabled on the system/boot volume after the next reboot.

```
manage-bde -on %systemdrive% -sk [ExternalDriveOrVolume]
reboot
```

NOTE

Prepare the VM with a separate data/resource VHD for getting the external key by using BitLocker.

Encrypting an OS drive on a running Linux VM

Prerequisites for OS disk encryption

- The VM must be using a distribution compatible with OS disk encryption as listed in the [Azure Disk Encryption FAQ](#).
- The VM must be created from the Marketplace image in Azure Resource Manager.
- Azure VM with at least 4 GB of RAM (recommended size is 7 GB).
- (For RHEL and CentOS) Disable SELinux. To disable SELinux, see "4.4.2. Disabling SELinux" in the [SELinux User's and Administrator's Guide](#) on the VM.
- After you disable SELinux, reboot the VM at least once.

Steps

1. Create a VM by using one of the distributions specified previously.

For CentOS 7.2, OS disk encryption is supported via a special image. To use this image, specify "7.2n" as the SKU when you create the VM:

```
Set-AzureRmVMSourceImage -VM $VirtualMachine -PublisherName "OpenLogic" -Offer "CentOS" -Skus "7.2n" -Version "latest"
```

2. Configure the VM according to your needs. If you're going to encrypt all the (OS + data) drives, the data drives need to be specified and mountable from /etc/fstab.

NOTE

Use UUID=... to specify data drives in /etc/fstab instead of specifying the block device name (for example, /dev/sdb1).

During encryption, the order of drives changes on the VM. If your VM relies on a specific order of block devices, it will fail to mount them after encryption.

3. Sign out of the SSH sessions.

4. To encrypt the OS, specify volumeType as **All** or **OS** when you enable encryption.

NOTE

All user-space processes that are not running as `systemd` services should be killed with a `SIGKILL`. Reboot the VM.

When you enable OS disk encryption on a running VM, plan on VM downtime.

5. Periodically monitor the progress of encryption by using the instructions in the [next section](#).

6. After Get-AzureRmVmDiskEncryptionStatus shows "VMRestartPending", restart your VM either by signing in to it or by using the portal, PowerShell, or CLI.

```
C:\> Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName $ResourceGroupName -VMName $VMName -ExtensionName $ExtensionName

OsVolumeEncrypted      : VMRestartPending
DataVolumesEncrypted   : NotMounted
OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage         : OS disk successfully encrypted, reboot the VM
```

Before you reboot, we recommend that you save [boot diagnostics](#) of the VM.

Monitoring OS encryption progress

You can monitor OS encryption progress in three ways:

- Use the `Get-AzureRmVmDiskEncryptionStatus` cmdlet and inspect the `ProgressMessage` field:

```
OsVolumeEncrypted      : EncryptionInProgress
DataVolumesEncrypted   : NotMounted
OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage        : OS disk encryption started
```

After the VM reaches "OS disk encryption started", it takes about 40 to 50 minutes on a Premium-storage backed VM.

Because of [issue #388](#) in WALinuxAgent, `OsVolumeEncrypted` and `DataVolumesEncrypted` show up as `Unknown` in some distributions. With WALinuxAgent version 2.1.5 and later, this issue is fixed automatically. If you see `Unknown` in the output, you can verify disk-encryption status by using the Azure Resource Explorer.

Go to [Azure Resource Explorer](#), and then expand this hierarchy in the selection panel on left:

```
-- subscriptions
  |-- [Your subscription]
    |-- resourceGroups
      |-- [Your resource group]
        |-- providers
          |-- Microsoft.Compute
            |-- virtualMachines
              |-- [Your virtual machine]
                |-- InstanceView
```

In the InstanceView, scroll down to see the encryption status of your drives.

```
60 -> {
  "code": "ProvisioningState/succeeded",
  "level": "Info",
  "displayStatus": "Provisioning succeeded",
  "time": "2016-09-22T02:19:41.4646766+00:00"
}
61 }
62 ]
63 ]
64 ]
65 ]
66 ]
67 ]
68 ],
69 -> "extensions": [
70 -> {
  "name": "AzureDiskEncryptionForLinux",
  "type": "Microsoft.Azure.Security.AzureDiskEncryptionForLinux",
  "typeHandlerVersion": "0.1.0.999190",
  "substatuses": [
71 -> {
  "code": "ComponentStatus/Microsoft.Azure.Security.AzureDiskEncryptionForLinux"
  "level": "Info",
  "displayStatus": "Provisioning succeeded",
  "message": "{\"os\": \"NotEncrypted\", \"data\": \"EncryptionInProgress\"}"
}
72 ]
73 ]
74 ]
75 ]
76 ]
77 ]
78 ]
79 ]
80 ]
```

- Look at [boot diagnostics](#). Messages from the ADE extension should be prefixed with `[AzureDiskEncryption]`.
- Sign in to the VM via SSH, and get the extension log from:

`/var/log/azure/Microsoft.Azure.Security.AzureDiskEncryptionForLinux`

We recommend that you don't sign-in to the VM while OS encryption is in progress. Copy the logs only when the other two methods have failed.

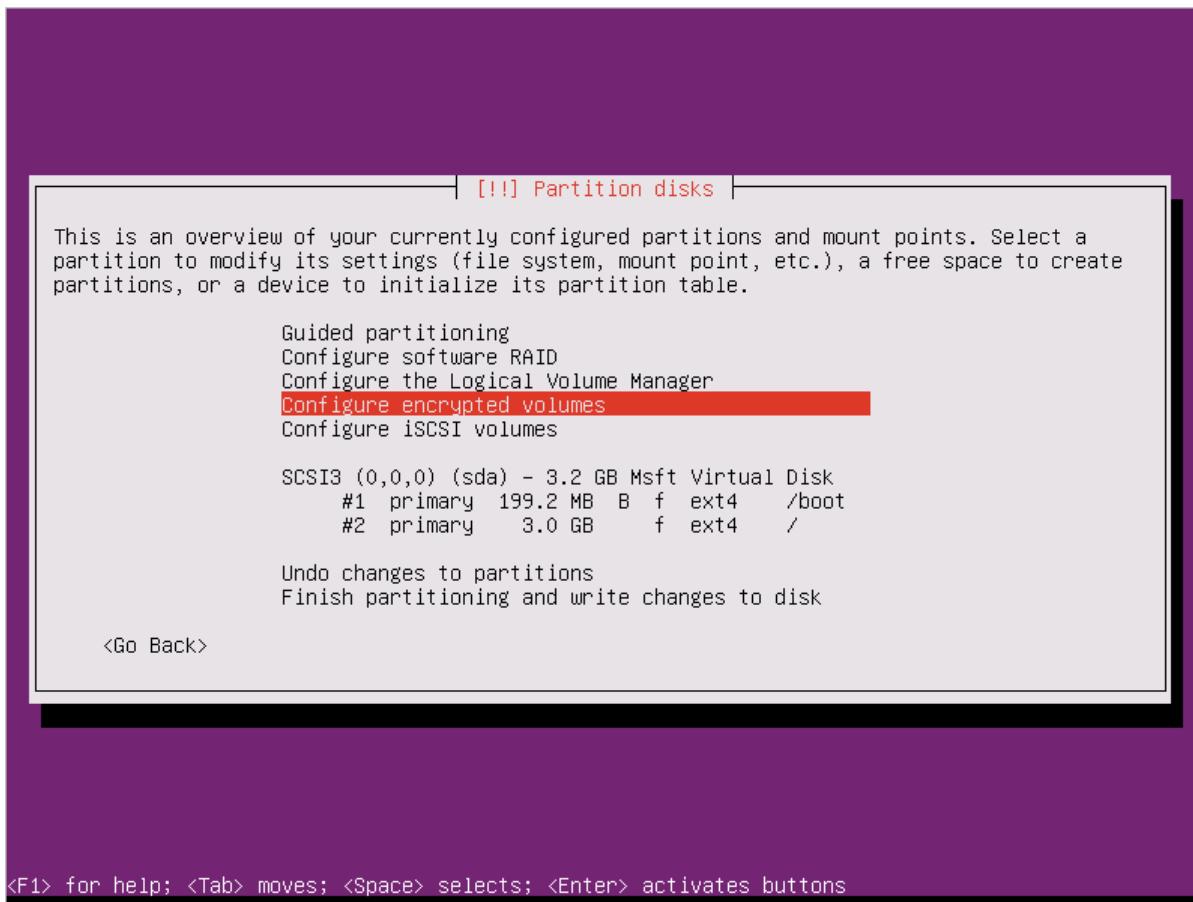
Prepare a pre-encrypted Linux VHD

The preparation for pre-encrypted VHDs can vary depending on the distribution. Examples on preparing [Ubuntu 16](#), [openSUSE 13.2](#), and [CentOS 7](#) are available.

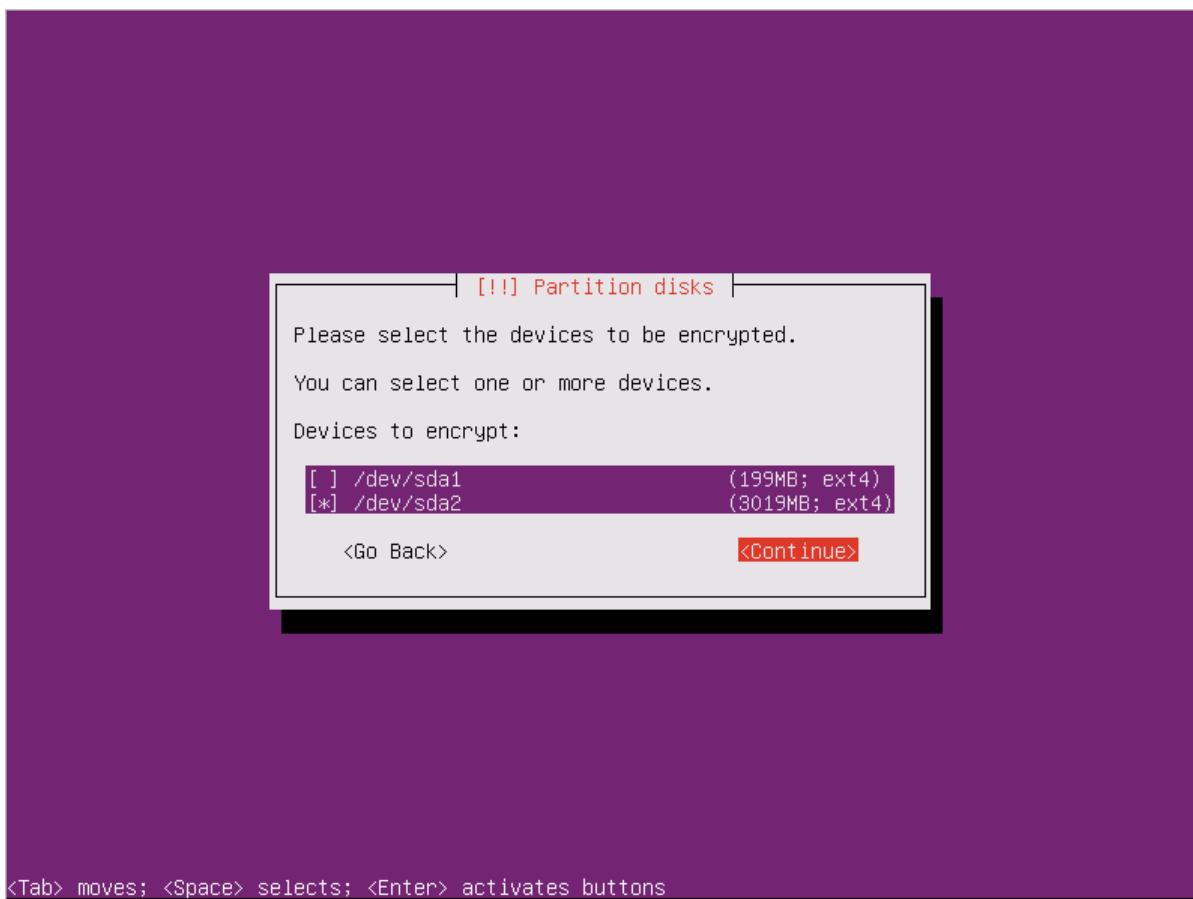
Ubuntu 16

Configure encryption during the distribution installation by doing the following steps:

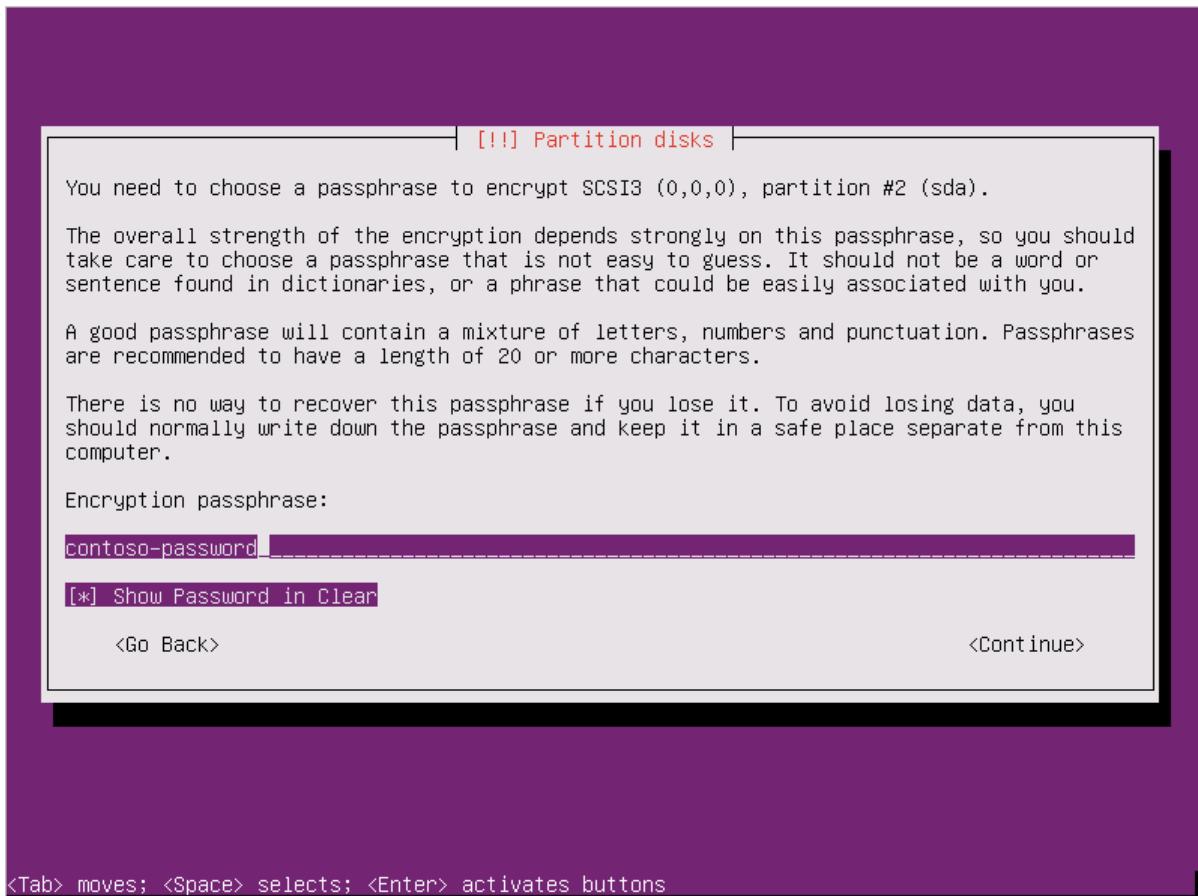
1. Select **Configure encrypted volumes** when you partition the disks.



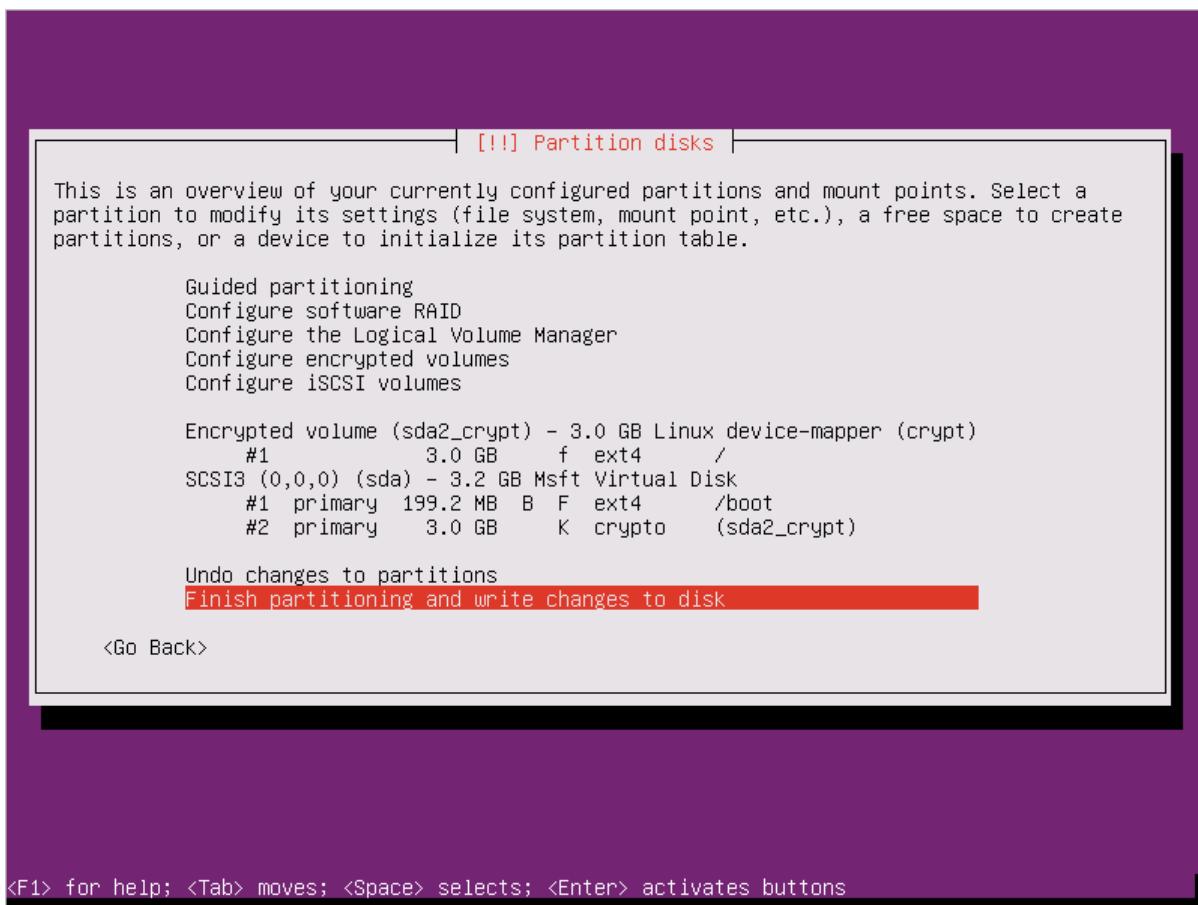
2. Create a separate boot drive, which must not be encrypted. Encrypt your root drive.



3. Provide a passphrase. This is the passphrase that you uploaded to the key vault.



4. Finish partitioning.



5. When you boot the VM and are asked for a passphrase, use the passphrase you provided in step 3.

```

[ 1.129797] input: Microsoft Umbus HID-compliant Mouse as /devices/0006:045E:0621.0001/input/input4
[ 1.132206]   sda: sda1 sda2
[ 1.133217] hid 0006:045E:0621.0001: input: <UNKNOWN> HID v0.01 Mouse [Microsoft Umbus HID-compliant Mouse] on
[ 1.134340] hv_netvsc: hv_netvsc channel opened successfully
[ 1.138418] sd 2:0:0:0: [sda] Attached SCSI disk
[ 1.265049] hv_netvsc umbus_15: Send section size: 6144, Section count:2560
[ 1.266137] hv_netvsc umbus_15: Device MAC 00:15:5d:05:34:01 link state up
[ 1.272596] scsi host3: storvsc_host_t
[ 1.436076] psmouse serio1: trackpoint: failed to get extended button data
Begin: Loading essential drivers ... [ 2.401782] md: linear personality registered for level -1
[ 2.404316] md: multipath personality registered for level -4
[ 2.407122] md: raid0 personality registered for level 0
[ 2.410610] md: raid1 personality registered for level 1
[ 2.480099] raid6: sse2x1 gen() 10995 MB/s
[ 2.548012] raid6: sse2x1 xor() 8467 MB/s
[ 2.616010] raid6: sse2x2 gen() 14312 MB/s
[ 2.684013] raid6: sse2x2 xor() 9555 MB/s
[ 2.752011] raid6: sse2x4 gen() 16205 MB/s
[ 2.820010] raid6: sse2x4 xor() 11594 MB/s
[ 2.888007] raid6: avx2x1 gen() 21995 MB/s
[ 2.956007] raid6: avx2x2 gen() 25959 MB/s
[ 3.024011] raid6: avx2x4 gen() 29505 MB/s
[ 3.024735] raid6: using algorithm avx2x4 gen() 29505 MB/s
[ 3.025038] raid6: using avx2x2 recovery algorithm
[ 3.027102] xor: automatically using best checksumming function:
[ 3.064003]   avx      : 35013.000 MB/sec
[ 3.065688] async_tx: api initialized (async)
[ 3.074685] md: raid6 personality registered for level 6
[ 3.075435] md: raid5 personality registered for level 5
[ 3.075746] md: raid4 personality registered for level 4
[ 3.079565] md: raid10 personality registered for level 10
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... Please unlock disk sda2_crypt: -

```

6. Prepare the VM for uploading into Azure using [these instructions](#). Don't run the last step (deprovisioning the VM) yet.

Configure encryption to work with Azure by doing the following steps:

1. Create a file under /usr/local/sbin/azure_crypt_key.sh, with the content in the following script. Pay attention to the KeyFileName, because it's the passphrase file name used by Azure.

```

#!/bin/sh
MountPoint=/tmp-keydisk-mount
KeyFileName=LinuxPassPhraseFileName
echo "Trying to get the key from disks ..." >&2
mkdir -p $MountPoint
modprobe vfat >/dev/null 2>&1
modprobe ntfs >/dev/null 2>&1
sleep 2
OPENED=0
cd /sys/block
for DEV in sd*; do

    echo "> Trying device: $DEV ..." >&2
    mount -t vfat -r /dev/${DEV}1 $MountPoint >/dev/null ||
    mount -t ntfs -r /dev/${DEV}1 $MountPoint >/dev/null
    if [ -f $MountPoint/$KeyFileName ]; then
        cat $MountPoint/$KeyFileName
        umount $MountPoint 2>/dev/null
        OPENED=1
        break
    fi
    umount $MountPoint 2>/dev/null
done

if [ $OPENED -eq 0 ]; then
    echo "FAILED to find suitable passphrase file ..." >&2
    echo -n "Try to enter your password: " >&2
    read -s -r A </dev/console
    echo -n "$A"
else
    echo "Success loading keyfile!" >&2
fi

```

2. Change the crypt config in `/etc/crypttab`. It should look like this:

```
xxx_crypt uuid=xxxxxxxxxxxxxxxxxxxxxx none luks,discard,keyscript=/usr/local/sbin/azure_crypt_key.sh
```

3. If you're editing `azure_crypt_key.sh` in Windows and you copied it to Linux, run

```
dos2unix /usr/local/sbin/azure_crypt_key.sh.
```

4. Add executable permissions to the script:

```
chmod +x /usr/local/sbin/azure_crypt_key.sh
```

5. Edit `/etc/initramfs-tools/modules` by appending lines: `vfat ntfs nls_cp437 nls_utf8 nls_iso8859-1`

6. Run `update-initramfs -u -k all` to update the initramfs to make the `keyscript` take effect.

7. Now you can deprovision the VM.

```

root@ubuntu-preencrypted:~# ls -l /usr/local/sbin/azure_crypt_key.sh
-rwxr-xr-x 1 root root 860 Sep 18 16:57 /usr/local/sbin/azure_crypt_key.sh
root@ubuntu-preencrypted:~# cat /etc/crypttab
sda2_crypt UUID=b0dee704-1f2a-4f02-9a13-289cc99dbb8 none luks,discard,keyscheme=/usr/local/sbin/azure_crypt_key.sh
root@ubuntu-preencrypted:~# cat /etc/initramfs-tools/modules
# List of modules that you want to include in your initramfs.
# They will be loaded at boot time in the order below.
#
# Syntax: module_name [args ...]
#
# You must run update-initramfs(8) to effect this change.
#
# Examples:
#
# raid1
# sd_mod
# vfat
# ntfs
# nls_cp437
# nls_utf8
# nls_iso859-1
root@ubuntu-preencrypted:~# update-initramfs -u -k all
update-initramfs: Generating /boot/initrd.img-4.4.0-36-generic
W: plymouth: The plugin label.so is missing, the selected theme might not work as expected.
W: plymouth: You might want to install the plymouth-themes and plymouth-label package to fix this.
W: mdadm: /etc/mdadm/mdadm.conf defines no arrays.
[ 6289.960173] blk_update_request: I/O error, dev fd0, sector 0
update-initramfs: Generating /boot/initrd.img-4.4.0-21-generic
W: plymouth: The plugin label.so is missing, the selected theme might not work as expected.
W: mdadm: /etc/mdadm/mdadm.conf defines no arrays.
[ 6297.592236] blk_update_request: I/O error, dev fd0, sector 0
root@ubuntu-preencrypted:~# waagent -force -deprovision
WARNING! The waagent service will be stopped.
WARNING! Cached DHCP leases will be deleted.
WARNING! root password will be disabled. You will not be able to login as root.
WARNING! Nameserver configuration in /etc/resolvconf/resolv.conf.d/tail,originial will be deleted.
2016/09/18 17:06:38.572398 INFO resolvconf is enabled; leaving /etc/resolv.conf intact
2016/09/18 17:06:38.572398 INFO resolvconf is enabled; leaving /etc/resolv.conf intact
root@ubuntu-preencrypted:~# export HISTSIZE=0
root@ubuntu-preencrypted:~# logout

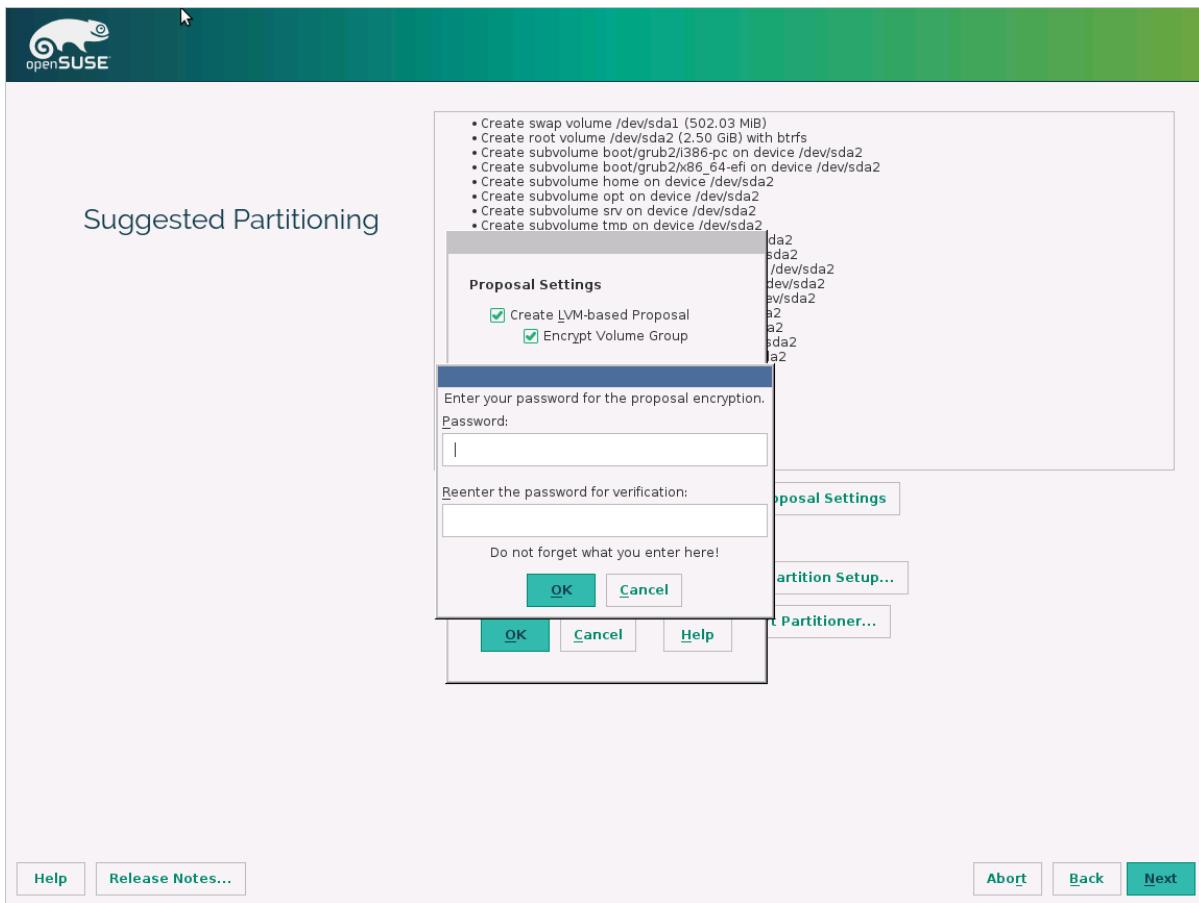
```

8. Continue to the next step and upload your VHD into Azure.

openSUSE 13.2

To configure encryption during the distribution installation, do the following steps:

1. When you partition the disks, select **Encrypt Volume Group**, and then enter a password. This is the password that you'll upload to your key vault.



2. Boot the VM using your password.

```
[ 0.000000] tsc: Fast TSC calibration failed
[ OK ] Found device Virtual_Disk.
[ OK ] Found device Virtual_Disk.
      Starting Cryptography Setup for cr_scsi-14d534654202020fd10f64360...278fd6327ec-part2...
      Starting Setup Virtual Console...
[ OK ] Started Setup Virtual Console.
      Starting Dispatch Password Requests to Console...
[ OK ] Started Dispatch Password Requests to Console.
Please enter passphrase for disk Virtual_Disk (cr_scsi-14d534654202020fd10f64360f5f14797052278fd63
27ec-part2)! ****
```

3. Prepare the VM for uploading to Azure by following the instructions in [Prepare a SLES or openSUSE virtual machine for Azure](#). Don't run the last step (deprovisioning the VM) yet.

To configure encryption to work with Azure, do the following steps:

1. Edit the /etc/dracut.conf, and add the following line: `add_drivers+=" vfat ntfs nls_cp437 nls_iso8859-1"`
2. Comment out these lines by the end of the file /usr/lib/dracut/modules.d/90crypt/module-setup.sh:

```
# inst_multiple -o \
# $systemdunitdir/system-generators/systemd-cryptsetup-generator \
# $systemdunitdir/systemd-cryptsetup \
# $systemdunitdir/systemd-ask-password-console.path \
# $systemdunitdir/systemd-ask-password-console.service \
# $systemdunitdir/cryptsetup.target \
# $systemdunitdir/sysinit.target.wants/cryptsetup.target \
# systemd-ask-password systemd-tty-ask-password-agent \
# inst_script "$moddir"/crypt-run-generator.sh /sbin/crypt-run-generator
```

3. Append the following line at the beginning of the file /usr/lib/dracut/modules.d/90crypt/parse-crypt.sh:

```
DRACUT_SYSTEMD=0
```

And change all occurrences of:

```
if [ -z "$DRACUT_SYSTEMD" ]; then
```

to:

```
if [ 1 ]; then
```

4. Edit /usr/lib/dracut/modules.d/90crypt/cryptroot-ask.sh and append it to "# Open LUKS device":

```
MountPoint=/tmp-keydisk-mount
KeyFileName=LinuxPassPhraseFileName
echo "Trying to get the key from disks ..." >&2
mkdir -p $MountPoint >&2
modprobe vfat >/dev/null >&2
modprobe ntfs >/dev/null >&2
for SFS in /dev/sd*; do
echo "> Trying device:$SFS..." >&2
mount ${SFS}1 $MountPoint -t vfat -r >&2 ||
mount ${SFS}1 $MountPoint -t ntfs -r >&2
if [ -f $MountPoint/$KeyFileName ]; then
echo "> Keyfile got..." >&2
cp $MountPoint/$KeyFileName /tmp-keyfile >&2
luksfile=/tmp-keyfile
umount $MountPoint >&2
break
fi
done
```

5. Run `/usr/sbin/dracut -f -v` to update the initrd.

6. Now you can deprovision the VM and upload your VHD into Azure.

CentOS 7

To configure encryption during the distribution installation, do the following steps:

1. Select **Encrypt my data** when you partition disks.

INSTALLATION DESTINATION

CENTOS 7 INSTALLATION

Done **us** **Help!**

Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

3072 MiB
 Msft Virtual Disk
sda / 3072 MiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

 Add a disk...

Disks left unselected here will not be touched.

Other Storage Options

Partitioning

Automatically configure partitioning. I will configure partitioning.

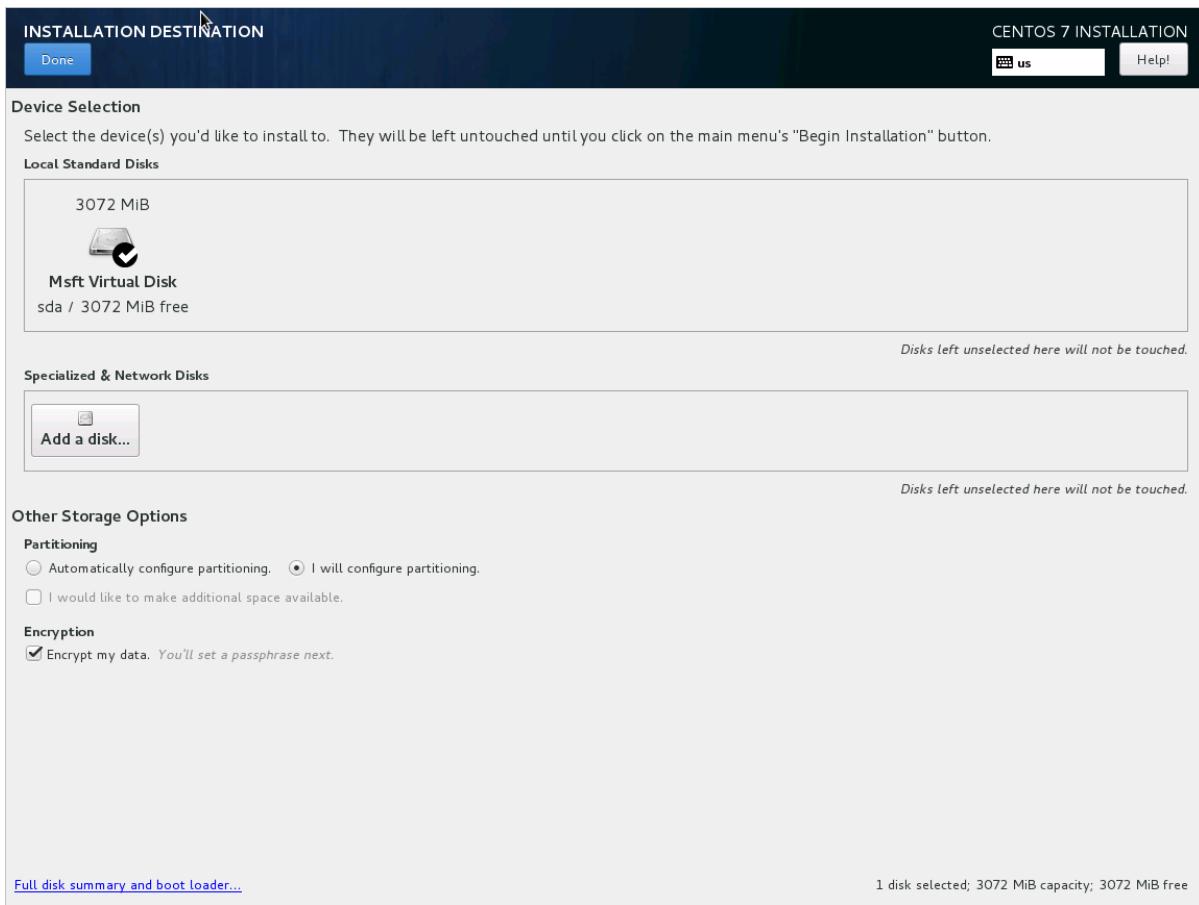
I would like to make additional space available.

Encryption

Encrypt my data. You'll set a passphrase next.

[Full disk summary and boot loader...](#)

1 disk selected; 3072 MiB capacity; 3072 MiB free



2. Make sure **Encrypt** is selected for root partition.

MANUAL PARTITIONING

CENTOS 7 INSTALLATION

Done **us** **Help!**

New CentOS 7 Installation

SYSTEM	
/boot	190 MiB
sda1	
/	2879 MiB >
luks-sda2	

luks-sda2

Mount Point:

Device(s):

Desired Capacity:

Modify...

Device Type: Standard Partition Encrypt

File System: xfs Reformat

Label:

Name: sda2

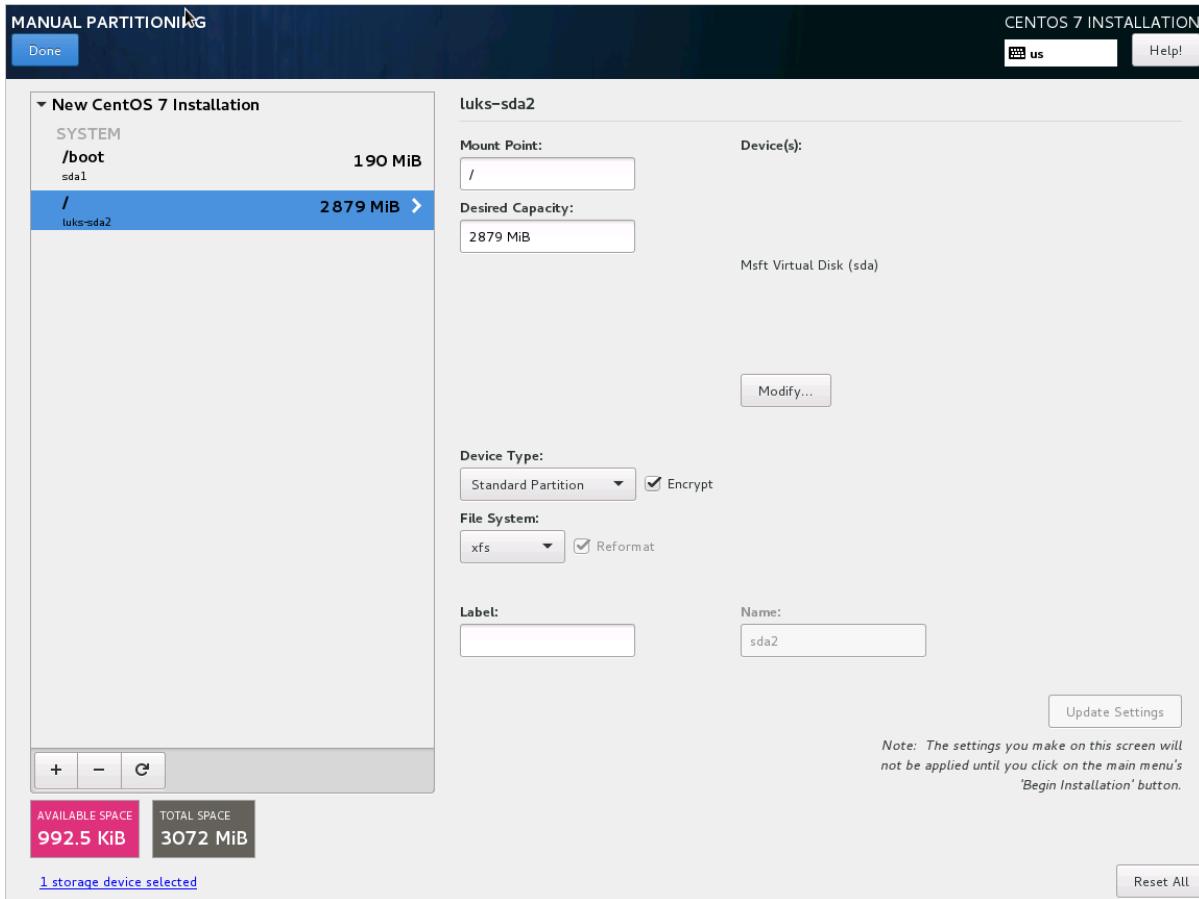
Update Settings

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

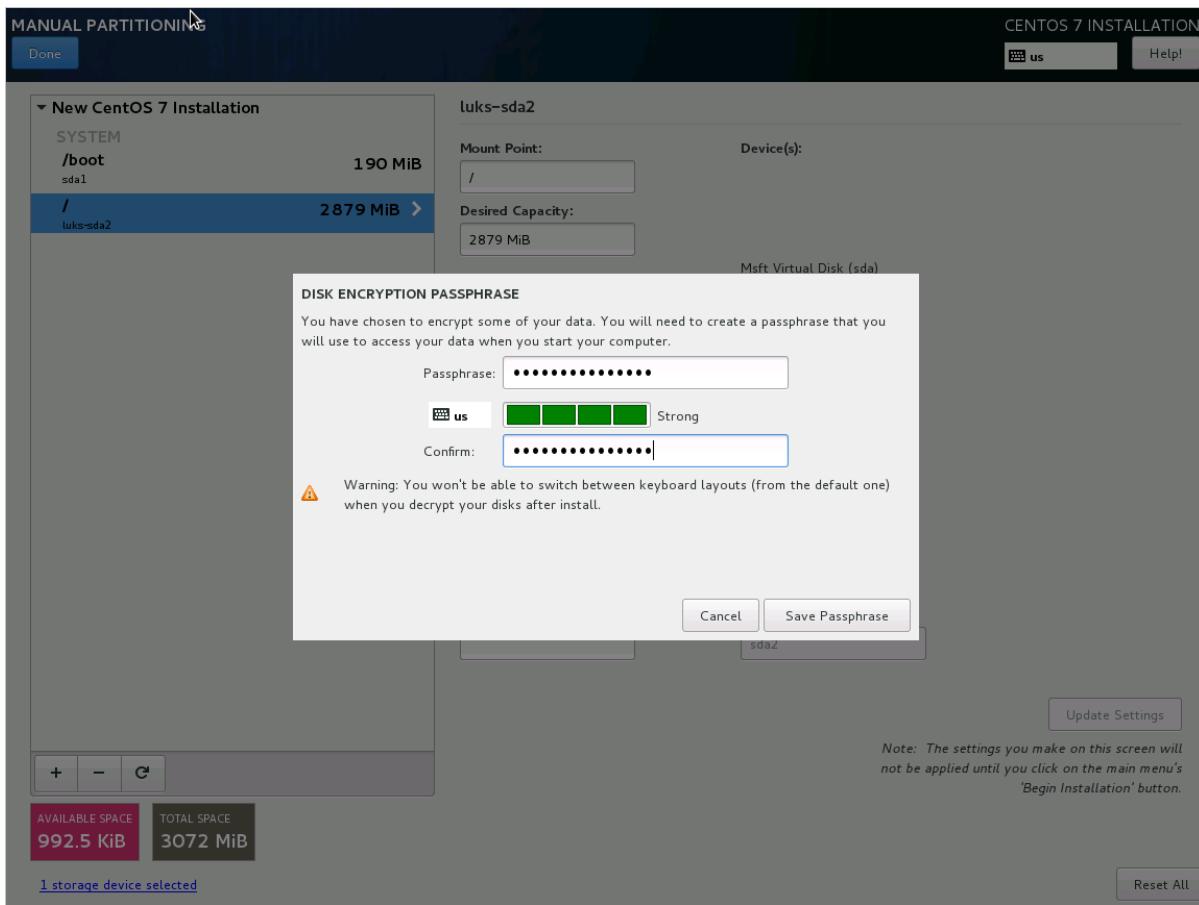
AVAILABLE SPACE **992.5 KiB** **TOTAL SPACE** **3072 MiB**

[1 storage device selected](#)

Reset All



3. Provide a passphrase. This is the passphrase that you'll upload to your key vault.



- When you boot the VM and are asked for a passphrase, use the passphrase you provided in step 3.



- Prepare the VM for uploading into Azure by using the "CentOS 7.0+" instructions in [Prepare a CentOS-based virtual machine for Azure](#). Don't run the last step (deprovisioning the VM) yet.
- Now you can deprovision the VM and upload your VHD into Azure.

To configure encryption to work with Azure, do the following steps:

1. Edit the /etc/dracut.conf, and add the following line:

```
add_drivers+=" vfat ntfs nls_cp437 nls_iso8859-1"
```

2. Comment out these lines by the end of the file /usr/lib/dracut/modules.d/90crypt/module-setup.sh:

```
# inst_multiple -o \
# $systemdunitdir/system-generators/systemd-cryptsetup-generator \
# $systemdunitdir/systemd-cryptsetup \
# $systemdunitdir/systemd-ask-password-console.path \
# $systemdunitdir/systemd-ask-password-console.service \
# $systemdunitdir/cryptsetup.target \
# $systemdunitdir/sysinit.target.wants/cryptsetup.target \
# systemd-ask-password systemd-tty-ask-password-agent
# inst_script "$moddir"/crypt-run-generator.sh /sbin/crypt-run-generator
```

3. Append the following line at the beginning of the file /usr/lib/dracut/modules.d/90crypt/parse-crypt.sh:

```
DRACUT_SYSTEMD=0
```

And change all occurrences of:

```
if [ -z "$DRACUT_SYSTEMD" ]; then
```

to

```
if [ 1 ]; then
```

4. Edit /usr/lib/dracut/modules.d/90crypt/cryptroot-ask.sh and append the following after the "# Open LUKS device":

```
MountPoint=/tmp-keydisk-mount KeyFileName=LinuxPassPhraseFileName echo "Trying to get the key from disks ..."
>&2 mkdir -p $MountPoint >&2 modprobe vfat >/dev/null >&2 modprobe ntfs >/dev/null >&2 for SFS in /dev/sd*;
do echo "> Trying device:$SFS..." >&2 mount ${SFS}1 $MountPoint -t vfat -r >&2 || mount ${SFS}1 $MountPoint -
t ntfs -r >&2 if [ -f $MountPoint/$KeyFileName ]; then echo "> keyfile got..." >&2 cp
$MountPoint/$KeyFileName /tmp-keyfile >&2 luksfile=/tmp-keyfile umount $MountPoint >&2 break fi done
```

5. Run the "/usr/sbin/dracut -f -v" to update the initrd.

```
[root@centos-preencrypted ~]# cat /etc/dracut.conf | grep add_drivers
add_drivers+=" vfat ntfs nls_cp437 nls_iso8859-1"
[root@centos-preencrypted ~]# cat /usr/lib/dracut/modules.d/90crypt/cryptroot-ask.sh | grep LinuxPassPhrasefileName -A 15 -B 1
MountPoint=/tmp-keydisk-mount
KeyFileName=LinuxPassPhraseFileName
echo "Trying to get the key from disks ..." >&2
mkdir -p $MountPoint >&2
modprobe vfat >/dev/null >&2
modprobe ntfs >/dev/null >&2
for SFS in /dev/sd*; do
echo "> Trying device:$SFS..." >&2
mount ${SFS}1 $MountPoint -t vfat -r >&2 ||
mount ${SFS}1 $MountPoint -t ntfs -r >&2
if [ ! -f $MountPoint/$KeyFileName ]; then
    echo "> keyfile got..." >&2
    cp $MountPoint/$KeyFileName /tmp-keyfile >&2
    luksfile=/tmp-keyfile
    umount $MountPoint >&2
    break
fi
[root@centos-preencrypted ~]# dracut -f -v_
```

Upload encrypted VHD to an Azure storage account

After BitLocker encryption or DM-Crypt encryption is enabled, the local encrypted VHD needs to be uploaded to your storage account.

```
Add-AzureRmVhd [-Destination] <Uri> [-LocalFilePath] <FileInfo> [[-NumberOfUploaderThreads] <Int32> ] [[-BaseImageUriToPatch] <Uri> ] [[-OverWrite]] [ <CommonParameters>]
```

Upload the secret for the pre-encrypted VM to your key vault

When encrypting using an Azure AD app (previous release), the disk-encryption secret that you obtained previously must be uploaded as a secret in your key vault. The key vault needs to have disk encryption and permissions enabled for your Azure AD client.

```
$AadClientId = "My-AAD-Client-Id"
$AadClientSecret = "My-AAD-Client-Secret"

$keyVault = New-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $ResourceGroupName -Location
$Location

Set-AzureRmKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $ResourceGroupName -
ServicePrincipalName $AadClientId -PermissionsToKeys all -PermissionsToSecrets all
Set-AzureRmKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $ResourceGroupName -
EnabledForDiskEncryption
```

Disk encryption secret not encrypted with a KEK

To set up the secret in your key vault, use [Set-AzureKeyVaultSecret](#). If you have a Windows virtual machine, the bek file is encoded as a base64 string and then uploaded to your key vault using the `Set-AzureKeyVaultSecret` cmdlet.

For Linux, the passphrase is encoded as a base64 string and then uploaded to the key vault. In addition, make sure that the following tags are set when you create the secret in the key vault.

Windows BEK file

```
# Change the VM Name, key vault name, and specify the path to the BEK file.
$VMName = "MySecureVM"
$BEKfilepath = "C:\test\BEK\12345678-90AB-CDEF-A1B2-C3D4E5F67890A.BEK"
$VeyVaultName = "MySecureVault"

# Get the name of the BEK file from the BEK file path. This will be a tag for the key vault secret.
$BEKfileName = Split-Path $BEKfilepath -Leaf

# These tags will be added to the key vault secret so you can easily see which BEK file belongs to which VM.
/tags = @{"MachineName" = "$VMName"; "DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP";
"DiskEncryptionKeyFileName" = "$BEKfileName"}

# Convert the BEK file to a Base64 string.
$FileContentEncoded = [System.convert]::ToBase64String((Get-Content -Path $BEKfilepath -Encoding Byte))

# Create a new secret in the vault from the converted BEK file.
# The file is converted to a secure string before import into the key vault

$SecretName = [guid]::NewGuid().ToString()
$SecureSecretValue = ConvertTo-SecureString $FileContentEncoded -AsPlainText -Force
$Secret = Set-AzureKeyVaultSecret -VaultName $VeyVaultName -Name $SecretName -SecretValue $SecureSecretValue -
tags $tags

# Show the secret's URL and store it as a variable. This is used as -DiskEncryptionKeyUrl in Set-
AzureRmVMOSDisk when you attach your OS disk.
$SecretUrl=$secret.Id
$SecretUrl
```

Linux

```
# This is the passphrase that was provided for encryption during the distribution installation
$passphrase = "contoso-password"

$tags = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP"; "DiskEncryptionKeyFileName" =
"LinuxPassPhraseFileName"}
$secretName = [guid]::NewGuid().ToString()
$secretValue = [Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($passphrase))
$secureSecretValue = ConvertTo-SecureString $secretValue -AsPlainText -Force

$secret = Set-AzureKeyVaultSecret -VaultName $KeyVaultName -Name $secretName -SecretValue $secureSecretValue -
tags $tags
$secretUrl = $secret.Id
```

Use the `$secretUrl` in the next step for [attaching the OS disk without using KEK](#).

Disk encryption secret encrypted with a KEK

Before you upload the secret to the key vault, you can optionally encrypt it by using a key encryption key. Use the wrap [API](#) to first encrypt the secret using the key encryption key. The output of this wrap operation is a base64 URL encoded string, which you can then upload as a secret by using the `Set-AzureKeyVaultSecret` cmdlet.

```
# This is the passphrase that was provided for encryption during the distribution installation
$passphrase = "contoso-password"

Add-AzureKeyVaultKey -VaultName $KeyVaultName -Name "keyencryptionkey" -Destination Software
$keyEncryptionKey = Get-AzureKeyVaultKey -VaultName $KeyVault.OriginalVault.Name -Name "keyencryptionkey"

$apiversion = "2015-06-01"
```

```

#####
# Get Auth URI
#####

$uri = $KeyVault.VaultUri + "/keys"
$headers = @{}

$response = try { Invoke-RestMethod -Method GET -Uri $uri -Headers $headers } catch { $_.Exception.Response
}

$authHeader = $response.Headers["www-authenticate"]
$authUri = [regex]::match($authHeader, 'authorization="(.?)"').Groups[1].Value

Write-Host "Got Auth URI successfully"

#####
# Get Auth Token
#####

$uri = $authUri + "/oauth2/token"
$body = "grant_type=client_credentials"
$body += "&client_id=" + $AadClientId
$body += "&client_secret=" + [Uri]::EscapeDataString($AadClientSecret)
$body += "&resource=" + [Uri]::EscapeDataString("https://vault.azure.net")
$headers = @{}

$response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body $body

$access_token = $response.access_token

Write-Host "Got Auth Token successfully"

#####
# Get KEK info
#####

$uri = $KeyEncryptionKey.Id + "?api-version=" + $apiversion
$headers = @{"Authorization" = "Bearer " + $access_token}

$response = Invoke-RestMethod -Method GET -Uri $uri -Headers $headers

$keyid = $response.key.kid

Write-Host "Got KEK info successfully"

#####
# Encrypt passphrase using KEK
#####

$passphraseB64 = [Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Passphrase))
$uri = $keyid + "/encrypt?api-version=" + $apiversion
$headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" = "application/json"}
$bodyObj = @{"alg" = "RSA-OAEP"; "value" = $passphraseB64}
$body = $bodyObj | ConvertTo-Json

$response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body $body

$wrappedSecret = $response.value

Write-Host "Encrypted passphrase successfully"

#####
# Store secret
#####

$secretName = [guid]::NewGuid().ToString()
$uri = $KeyVault.VaultUri + "/secrets/" + $secretName + "?api-version=" + $apiversion
$secretAttributes = @{"enabled" = $true}
$secretTags = @{
    "EncryptionAlgorithm" = "RSA-OAEP";
    "EncryptionKeyFileName" =

```

```

$secretTags = @{
    DISKENCRYPTIONKEYENCRIPTIONTAG1 = "RSA-HASH" ;
    DISKENCRYPTIONKEYENCRIPTIONTAG2 = "AES-CBC" ;
    DISKENCRYPTIONKEYENCRIPTIONTAG3 = "AES-GCM"
}

"LinuxPassPhraseFileName"} }

$headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" = "application/json"}
$bodyObj = @{$value" = $wrappedSecret; "attributes" = $secretAttributes; "tags" = $secretTags}
$body = $bodyObj | ConvertTo-Json

$response = Invoke-RestMethod -Method PUT -Uri $uri -Headers $headers -Body $body

Write-Host "Stored secret successfully"

$secretUrl = $response.id

```

Use `$keyEncryptionKey` and `$secretUrl` in the next step for [attaching the OS disk using KEK](#).

Specify a secret URL when you attach an OS disk

Without using a KEK

While you're attaching the OS disk, you need to pass `$secretUrl`. The URL was generated in the "Disk-encryption secret not encrypted with a KEK" section.

```

Set-AzureRmVMOSDisk ` 
    -VM $VirtualMachine ` 
    -Name $OSDiskName ` 
    -SourceImageUri $VhdUri ` 
    -VhdUri $OSDiskUri ` 
    -Linux ` 
    -CreateOption FromImage ` 
    -DiskEncryptionKeyVaultId $KeyVault.ResourceId ` 
    -DiskEncryptionKeyUrl $SecretUrl

```

Using a KEK

When you attach the OS disk, pass `$keyEncryptionKey` and `$secretUrl`. The URL was generated in the "Disk encryption secret encrypted with a KEK" section.

```

Set-AzureRmVMOSDisk ` 
    -VM $VirtualMachine ` 
    -Name $OSDiskName ` 
    -SourceImageUri $CopiedTemplateBlobUri ` 
    -VhdUri $OSDiskUri ` 
    -Linux ` 
    -CreateOption FromImage ` 
    -DiskEncryptionKeyVaultId $KeyVault.ResourceId ` 
    -DiskEncryptionKeyUrl $SecretUrl ` 
    -KeyEncryptionKeyVaultId $KeyVault.ResourceId ` 
    -KeyEncryptionKeyURL $KeyEncryptionKey.Id

```

Azure Disk Encryption for IaaS VMs FAQ

12/7/2018 • 7 minutes to read • [Edit Online](#)

This article provides answers to frequently asked questions (FAQ) about Azure Disk Encryption for Windows and Linux IaaS VMs. For more information about this service, see [Azure Disk Encryption for Windows and Linux IaaS VMs](#).

Where is Azure Disk Encryption in general availability (GA)?

Azure Disk Encryption for Windows and Linux IaaS VMs is in general availability in all Azure public regions.

What user experiences are available with Azure Disk Encryption?

Azure Disk Encryption GA supports Azure Resource Manager templates, Azure PowerShell, and Azure CLI. The different user experiences give you flexibility. You have three different options for enabling disk encryption for your IaaS VMs. For more information on the user experience and step-by-step guidance available in Azure Disk Encryption, see [Enable Azure Disk Encryption for Windows](#) and [Enable Azure Disk Encryption for Linux](#).

How much does Azure Disk Encryption cost?

There's no charge for encrypting VM disks with Azure Disk Encryption but there are charges associated with the use of Azure Key Vault. For more information on Azure Key Vault costs, see the [Key Vault pricing](#) page.

Which virtual machine tiers does Azure Disk Encryption support?

Azure Disk Encryption is available on standard tier VMs including [A, D, DS, G, GS, and F](#) series IaaS VMs. It's also available for VMs with premium storage. It isn't available on basic tier VMs.

What Linux distributions does Azure Disk Encryption support?

Azure Disk Encryption is supported on the following Linux server distributions and versions:

LINUX DISTRIBUTION	VERSION	VOLUME TYPE SUPPORTED FOR ENCRYPTION
Ubuntu	16.04-DAILY-LTS	OS and data disk
Ubuntu	14.04.5-DAILY-LTS	OS and data disk
RHEL	7.5	Data disk*
RHEL	7.4	Data disk*
RHEL	7.3	Data disk*
RHEL	7.2	Data disk*
RHEL	6.8	Data disk*
RHEL	6.7	Data disk*

LINUX DISTRIBUTION	VERSION	VOLUME TYPE SUPPORTED FOR ENCRYPTION
CentOS	7.4	OS and data disk
CentOS	7.3	OS and data disk
CentOS	7.2n	OS and data disk
CentOS	6.8	OS and data disk
CentOS	7.1	Data disk
CentOS	7.0	Data disk
CentOS	6.7	Data disk
CentOS	6.6	Data disk
CentOS	6.5	Data disk
openSUSE	13.2	Data disk
SLES	12 SP1	Data disk
SLES	Priority: 12-SP1	Data disk
SLES	HPC 12	Data disk
SLES	Priority: 11-SP4	Data disk
SLES	11 SP4	Data disk

***ADE is supported for RHEL for data disk. The current ADE implementation does work for OS disk but isn't currently jointly supported. Both Microsoft and Red Hat are working on a jointly supported solution. In the interim, you can reference the [Azure Disk Encryption for Linux](#) article.**

How can I start using Azure Disk Encryption?

To get started, read the [Azure Disk Encryption overview](#).

Can I encrypt both boot and data volumes with Azure Disk Encryption?

Yes, you can encrypt boot and data volumes for Windows and Linux IaaS VMs. For Windows VMs, you can't encrypt the data without first encrypting the OS volume. For Linux VMs, it's possible to encrypt the data volume without having to encrypt the OS volume first. After you've encrypted the OS volume for Linux, disabling encryption on an OS volume for Linux IaaS VMs isn't supported.

Does Azure Disk Encryption allow you to bring your own key (BYOK)?

Yes, you can supply your own key encryption keys. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption keys support scenarios, see [Azure Disk Encryption prerequisites](#).

Can I use an Azure-created key encryption key?

Yes, you can use Azure Key Vault to generate a key encryption key for Azure disk encryption use. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption key, see [Azure Disk Encryption prerequisites](#).

Can I use an on-premises key management service or HSM to safeguard the encryption keys?

You can't use the on-premises key management service or HSM to safeguard the encryption keys with Azure Disk Encryption. You can only use the Azure Key Vault service to safeguard the encryption keys. For more information on the key encryption key support scenarios, see [Azure Disk Encryption prerequisites](#).

What are the prerequisites to configure Azure Disk Encryption?

There are prerequisites for Azure Disk Encryption. See the [Azure Disk Encryption prerequisites](#) article to create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and keys. For more information on the key encryption key support scenarios, see [Azure Disk Encryption overview](#).

What are the prerequisites to configure Azure Disk Encryption with an Azure AD app (previous release)?

There are prerequisites for Azure Disk Encryption. See the [Azure Disk Encryption prerequisites](#) article to create an Azure Active Directory application, create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and keys. For more information on the key encryption key support scenarios, see [Azure Disk Encryption overview](#).

Is Azure Disk Encryption using an Azure AD app (previous release) still supported?

Yes. Disk encryption using an Azure AD app is still supported. However, when encrypting new VMs it's recommended that you use the new method rather than encrypting with an Azure AD app.

Can I migrate VMs that were encrypted with an Azure AD app to encryption without an Azure AD app?

Currently, there isn't a direct migration path for machines that were encrypted with an Azure AD app to encryption without an Azure AD app. Additionally, there isn't a direct path from encryption without an Azure AD app to encryption with an AD app.

What version of Azure PowerShell does Azure Disk Encryption support?

Use the latest version of the Azure PowerShell SDK to configure Azure Disk Encryption. Download the latest version of [Azure PowerShell](#). Azure Disk Encryption is *not* supported by Azure SDK version 1.1.0.

NOTE

The Linux Azure disk encryption preview extension is deprecated. For details, see [Deprecating Azure disk encryption preview extension for Linux IaaS VMs](#).

Can I apply Azure Disk Encryption on my custom Linux image?

You can't apply Azure Disk Encryption on your custom Linux image. Only the gallery Linux images for the supported distributions called out previously are supported. Custom Linux images aren't currently supported.

Can I apply updates to a Linux Red Hat VM that uses the yum update?

Yes, you can perform an update or patch a Red Hat Linux VM. For more information, see [Applying updates to an encrypted Azure IaaS Red Hat VM by using the yum update](#).

What is the recommended Azure disk encryption workflow for Linux?

The following workflow is recommended to have the best results on Linux:

- Start from the unmodified stock gallery image corresponding to the needed OS distro and version
- Back up any mounted drives that will be encrypted. This back up allows for recovery if there's a failure, for example if the VM is rebooted before encryption has completed.
- Encrypt (can take several hours or even days depending on VM characteristics and size of any attached data disks)
- Customize, and add software to the image as needed.

If this workflow isn't possible, relying on [Storage Service Encryption \(SSE\)](#) at the platform storage account layer may be an alternative to full disk encryption using dm-crypt.

What is the disk "Bek Volume" or "/mnt/azure_bek_disk"?

"Bek volume" for Windows or "/mnt/azure_bek_disk" for Linux is a local data volume that securely stores the encryption keys for Encrypted Azure IaaS VMs.

NOTE

Do not delete or edit any contents in this disk. Do not unmount the disk since the encryption key presence is needed for any encryption operations on the IaaS VM.

What encryption method does Azure Disk Encryption use?

On Windows, ADE uses the Bitlocker AES256 encryption method (AES256WithDiffuser on versions prior to Windows Server 2012). On Linux, ADE uses the dmcrypt default of aes-xts-plain64 with a 256-bit volume master key.

If I use EncryptFormatAll and specify all volume types, will it erase the data on the data drives that we already encrypted?

No, data won't be erased from data drives that are already encrypted using Azure Disk Encryption. Similar to how EncryptFormatAll didn't re-encrypt the OS drive, it won't re-encrypt the already encrypted data drive. For more information, see the [EncryptFormatAll criteria](#).

Where can I go to ask questions or provide feedback?

You can ask questions or provide feedback on the [Azure Disk Encryption forum](#).

Next steps

In this document, you learned more about the most frequent questions related to Azure Disk Encryption. For more information about this service, see the following articles:

- [Azure Disk Encryption Overview](#)
- [Apply disk encryption in Azure Security Center](#)
- [Azure data encryption at rest](#)

Azure Disk Encryption troubleshooting guide

12/7/2018 • 6 minutes to read • [Edit Online](#)

This guide is for IT professionals, information security analysts, and cloud administrators whose organizations use Azure Disk Encryption. This article is to help with troubleshooting disk-encryption-related problems.

Troubleshooting Linux OS disk encryption

Linux operating system (OS) disk encryption must unmount the OS drive before running it through the full disk encryption process. If it can't unmount the drive, an error message of "failed to unmount after ..." is likely to occur.

This error can occur when OS disk encryption is tried on a target VM environment that has been changed from the supported stock gallery image. Deviations from the supported image can interfere with the extension's ability to unmount the OS drive. Examples of deviations can include the following items:

- Customized images no longer match a supported file system or partitioning scheme.
- Large applications such as SAP, MongoDB, Apache Cassandra, and Docker aren't supported when they're installed and running in the OS before encryption. Azure Disk Encryption is unable to shut down these processes safely as required in preparation of the OS drive for disk encryption. If there are still active processes holding open file handles to the OS drive, the OS drive can't be unmounted, resulting in a failure to encrypt the OS drive.
- Custom scripts that run in close time proximity to the encryption being enabled, or if any other changes are being made on the VM during the encryption process. This conflict can happen when an Azure Resource Manager template defines multiple extensions to execute simultaneously, or when a custom script extension or other action runs simultaneously to disk encryption. Serializing and isolating such steps might resolve the issue.
- Security Enhanced Linux (SELinux) hasn't been disabled before enabling encryption, so the unmount step fails. SELinux can be reenabled after encryption is complete.
- The OS disk uses a Logical Volume Manager (LVM) scheme. Although limited LVM data disk support is available, an LVM OS disk isn't.
- Minimum memory requirements aren't met (7 GB is suggested for OS disk encryption).
- Data drives are recursively mounted under the /mnt/ directory, or each other (for example, /mnt/data1, /mnt/data2, /data3 + /data3/data4).
- Other Azure Disk Encryption [prerequisites](#) for Linux aren't met.

Unable to encrypt

In some cases, the Linux disk encryption appears to be stuck at "OS disk encryption started" and SSH is disabled. The encryption process can take between 3-16 hours to finish on a stock gallery image. If multi-terabyte-sized data disks are added, the process might take days.

The Linux OS disk encryption sequence unmounts the OS drive temporarily. It then performs block-by-block encryption of the entire OS disk, before it remounts it in its encrypted state. Unlike Azure Disk Encryption on Windows, Linux Disk Encryption doesn't allow for concurrent use of the VM while the encryption is in progress. The performance characteristics of the VM can make a significant difference in the time required to complete encryption. These characteristics include the size of the disk and whether the storage account is standard or premium (SSD) storage.

To check the encryption status, poll the **ProgressMessage** field returned from the [Get-AzureRmVmDiskEncryptionStatus](#) command. While the OS drive is being encrypted, the VM enters a servicing state, and disables SSH to prevent any disruption to the ongoing process. The **EncryptionInProgress** message

reports for the majority of the time while the encryption is in progress. Several hours later, a **VMRestartPending** message prompts you to restart the VM. For example:

```
PS > Get-AzureRmVMDiskEncryptionStatus -ResourceGroupName $resourceGroupName -VMName $vmName
OsVolumeEncrypted      : EncryptionInProgress
DataVolumesEncrypted   : EncryptionInProgress
OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage        : OS disk encryption started

PS > Get-AzureRmVMDiskEncryptionStatus -ResourceGroupName $resourceGroupName -VMName $vmName
OsVolumeEncrypted      : VMRestartPending
DataVolumesEncrypted   : Encrypted
OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage        : OS disk successfully encrypted, please reboot the VM
```

After you're prompted to reboot the VM, and after the VM restarts, you must wait 2-3 minutes for the reboot and for the final steps to be performed on the target. The status message changes when the encryption is finally complete. After this message is available, the encrypted OS drive is expected to be ready for use and the VM is ready to be used again.

In the following cases, we recommend that you restore the VM back to the snapshot or backup taken immediately before encryption:

- If the reboot sequence, described previously, doesn't happen.
- If the boot information, progress message, or other error indicators report that OS encryption has failed in the middle of this process. An example of a message is the "failed to unmount" error that is described in this guide.

Before the next attempt, reevaluate the characteristics of the VM and make sure that all of the prerequisites are satisfied.

Troubleshooting Azure Disk Encryption behind a firewall

When connectivity is restricted by a firewall, proxy requirement, or network security group (NSG) settings, the ability of the extension to perform needed tasks might be disrupted. This disruption can result in status messages such as "Extension status not available on the VM." In expected scenarios, the encryption fails to finish. The sections that follow have some common firewall problems that you might investigate.

Network security groups

Any network security group settings that are applied must still allow the endpoint to meet the documented network configuration [prerequisites](#) for disk encryption.

Azure Key Vault behind a firewall

The VM must be able to access a key vault. Refer to guidance on access to the key vault from behind a firewall that the [Azure Key Vault](#) team maintains.

Linux package management behind a firewall

At runtime, Azure Disk Encryption for Linux relies on the target distribution's package management system to install needed prerequisite components before enabling encryption. If the firewall settings prevent the VM from being able to download and install these components, then subsequent failures are expected. The steps to configure this package management system can vary by distribution. On Red Hat, when a proxy is required, you must make sure that the subscription-manager and yum are set up properly. For more information, see [How to troubleshoot subscription-manager and yum problems](#).

Troubleshooting Windows Server 2016 Server Core

On Windows Server 2016 Server Core, the bdehdcfg component isn't available by default. This component is

required by Azure Disk Encryption. It's used to split the system volume from OS volume, which is done only once for the life time of the VM. These binaries aren't required during later encryption operations.

To work around this issue, copy the following four files from a Windows Server 2016 Data Center VM to the same location on Server Core:

```
\windows\system32\bdehdcfg.exe  
\windows\system32\bdehdcfglib.dll  
\windows\system32\en-US\bdehdcfglib.dll.mui  
\windows\system32\en-US\bdehdcfg.exe.mui
```

1. Enter the following command:

```
bdehdcfg.exe -target default
```

2. This command creates a 550-MB system partition. Reboot the system.
3. Use DiskPart to check the volumes, and then proceed.

For example:

```
DISKPART> list vol

Volume ### Ltr Label Fs Type Size Status Info
----- -- -----
Volume 0 C NTFS Partition 126 GB Healthy Boot
Volume 1 NTFS Partition 550 MB Healthy System
Volume 2 D Temporary S NTFS Partition 13 GB Healthy Pagefile
```

Next steps

In this document, you learned more about some common problems in Azure Disk Encryption and how to troubleshoot those problems. For more information about this service and its capabilities, see the following articles:

- [Apply disk encryption in Azure Security Center](#)
- [Azure data encryption at rest](#)

Azure Disk Encryption prerequisites (previous release)

12/13/2018 • 21 minutes to read • [Edit Online](#)

The new release of Azure Disk Encryption eliminates the requirement for providing an Azure AD application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Azure AD credentials during the enable encryption step. All new VMs must be encrypted without the Azure AD application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see [Azure Disk Encryption prerequisites](#). VMs that were already encrypted with Azure AD application parameters are still supported and should continue to be maintained with the AAD syntax.

This article, Azure Disk Encryption Prerequisites, explains items that need to be in place before you can use Azure Disk Encryption. Along with general prerequisites, Azure Disk Encryption is integrated with [Azure Key Vault](#) and it uses an Azure AD application to provide authentication in order to manage encryption keys in the key vault. You may also wish to use [Azure PowerShell](#) or the [Azure CLI](#) to set up or configure Key Vault and the Azure AD application.

Before you enable Azure Disk Encryption on Azure IaaS VMs for the supported scenarios that were discussed in the [Azure Disk Encryption Overview](#) article, be sure to have the prerequisites in place.

NOTE

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.

Supported operating systems

Azure Disk Encryption is supported on the following operating systems:

- Windows Server versions: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016.
 - For Windows Server 2008 R2, you must have .NET Framework 4.5 installed before you enable encryption in Azure. 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 client versions: Windows 8 client and Windows 10 client.
- Azure Disk Encryption is only supported on specific Azure Gallery based Linux server distributions and versions. For the list of currently supported versions, refer to the [Azure Disk Encryption FAQ](#).
- Azure Disk Encryption requires that your key vault and VMs reside in the same Azure region and subscription. Configuring the resources in separate regions causes a failure in enabling the Azure Disk Encryption feature.

Additional prerequisites for Linux IaaS VMs

- Azure Disk Encryption for Linux requires 7 GB of RAM on the VM to enable OS disk encryption on [supported images](#). Once the OS disk encryption process is complete, the VM can be configured to run with less memory.
- Before enabling encryption, the data disks to be encrypted need to be properly listed in /etc/fstab. Use a persistent block device name for this entry, as device names in the "/dev/sdX" format can't be relied upon to be associated with the same disk across reboots, particularly after encryption is applied. 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 desired drive is still mounted.

- If the /etc/fstab file doesn't mount the drive properly prior to 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.
- After reboot, it will take time for the Azure Disk Encryption process to mount the newly encrypted disks. They won't immediately be available after a reboot. The process needs time to start, unlock, and then mount the encrypted drives prior to their being available for other processes to access. This process may take more than a minute after reboot depending on the system characteristics.

An example of commands that can be used to mount the data disks and create the necessary /etc/fstab entries can be found in [lines 197-205 of this script file](#).

Networking and Group Policy

To enable the Azure Disk Encryption feature using the older AAD parameter syntax, the IaaS VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the IaaS 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 IaaS VM must be able to connect to the key vault endpoint.
- The IaaS 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](#).
- On Windows, if TLS 1.0 has been explicitly disabled and the .NET version has not been updated to 4.6 or higher, the following registry change will enable ADE to select the more recent TLS version:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001
```

Group Policy:

- The Azure Disk Encryption solution uses the BitLocker external key protector for Windows IaaS 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, force the new policy to update (gpupdate.exe /force), and then restarting may be required.

Azure PowerShell

[Azure PowerShell](#) provides a set of cmdlets that uses the [Azure Resource Manager](#) model for managing your

Azure resources. You can use it in your browser with [Azure Cloud Shell](#), or you can install it on your local machine using the instructions below to use it in any PowerShell session. If you already have it installed locally, make sure you use the latest version of Azure PowerShell SDK version to configure Azure Disk Encryption. Download the latest version of [Azure PowerShell release](#).

Install Azure PowerShell for use on your local machine (optional):

1. Follow the instructions in the links for your operating system, then continue though the rest of the steps below.

- [Install and configure Azure PowerShell for Windows](#).
 - Install PowerShellGet, Azure PowerShell, and load the AzureRM module.
- [Install and configure Azure Powershell on macOS and Linux](#).
 - Install PowerShell Core, Azure PowerShell for .NET Core, and load the Az module.

2. Install the [Azure Active Directory PowerShell module](#).

```
Install-Module AzureAD
```

3. Verify the installed versions of the modules.

```
Get-Module AzureRM -ListAvailable | Select-Object -Property Name,Version,Path  
Get-Module AzureAD -ListAvailable | Select-Object -Property Name,Version,Path
```

4. Sign in to Azure using the [Connect-AzureRmAccount cmdlet](#).

```
Connect-AzureRmAccount  
# For specific instances of Azure, use the -Environment parameter.  
Connect-AzureRmAccount -Environment (Get-AzureRmEnvironment -Name AzureUSGovernment)  
  
<# If you have multiple subscriptions and want to specify a specific one,  
get your subscription list with Get-AzureRmSubscription and  
specify it with Set-AzureRmContext. #>  
Get-AzureRmSubscription  
Set-AzureRmContext -SubscriptionId "xxxx-xxxx-xxxx-xxxx"
```

5. Connect to Azure AD [Connect-AzureAD](#).

```
Connect-AzureAD
```

6. Review [Getting started with Azure PowerShell](#) and [AzureAD](#), if needed.

Azure CLI

The [Azure CLI 2.0](#) is a command-line tool for managing Azure resources. The CLI is designed to flexibly query data, support long-running operations as non-blocking processes, and make scripting easy. You can use it in your browser with [Azure Cloud Shell](#), or you can install it on your local machine and use it in any PowerShell session.

1. [Install Azure CLI](#) for use on your local machine (optional):

2. Verify the installed version.

```
az --version
```

3. Sign in to Azure using [az login](#).

```
az login

# If you would like to select a tenant, use:
az login --tenant "<tenant>"

# If you have multiple subscriptions, get your subscription list with az account list and specify with
az account set.
az account list
az account set --subscription "<subscription name or ID>"
```

4. Review [Get started with Azure CLI 2.0](#) if needed.

Prerequisite workflow for Key Vault and the Azure AD app

If you're already familiar with the Key Vault and Azure AD prerequisites for Azure Disk Encryption, you can use the [Azure Disk Encryption prerequisites PowerShell script](#). For more information on using the prerequisites script, see the [Encrypt a VM Quickstart](#) and the [Azure Disk Encryption Appendix](#).

1. Create a key vault.
2. Set up an Azure AD application and service principal.
3. Set the key vault access policy for the Azure AD app.
4. Set key vault advanced access policies.

Create a key vault

Azure Disk Encryption is integrated with [Azure Key Vault](#) to help you control and manage the disk-encryption keys and secrets in your key vault subscription. You can create a key vault or use an existing one for Azure Disk Encryption. For more information about key vaults, see [Get started with Azure Key Vault](#) and [Secure your key vault](#). You can use a Resource Manager template, Azure PowerShell, or the Azure CLI to create a key vault.

WARNING

In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

Create a key vault with PowerShell

You can create a key vault with Azure PowerShell using the [New-AzureRmKeyVault](#) cmdlet. For additional cmdlets for Key Vault, see [AzureRM.KeyVault](#).

1. If needed, [connect to your Azure subscription](#).
2. Create a new resource group, if needed, with [New-AzureRmResourceGroup](#). To list data center locations, use [Get-AzureRmLocation](#).

```
# Get-AzureRmLocation
New-AzureRmResourceGroup -Name 'MySecureRG' -Location 'East US'
```

3. Create a new key vault using [New-AzureRmKeyVault](#)

```
New-AzureRmKeyVault -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -Location 'East US'
```

4. Note the **Vault Name**, **Resource Group Name**, **Resource ID**, **Vault URI**, and the **Object ID** that are returned for later use when you encrypt the disks.

Create a key vault with Azure CLI

You can manage your key vault with Azure CLI using the [az keyvault](#) commands. To create a key vault, use [az keyvault create](#).

1. If needed, [connect to your Azure subscription](#).
2. Create a new resource group, if needed, with [az group create](#). To list locations, use [az account list-locations](#)

```
# To list locations: az account list-locations --output table  
az group create -n "MySecureRG" -l "East US"
```

3. Create a new key vault using [az keyvault create](#).

```
az keyvault create --name "MySecureVault" --resource-group "MySecureRG" --location "East US"
```

4. Note the **Vault Name** (name), **Resource Group Name**, **Resource ID** (ID), **Vault URI**, and the **Object ID** that are returned for use later.

Create a key vault with a Resource Manager template

You can create a key vault by using the [Resource Manager template](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, Key Vault name, Object ID, legal terms, and agreement, and then click **Purchase**.

Set up an Azure AD app and service principal

When you need encryption to be enabled on a running VM in Azure, Azure Disk Encryption generates and writes the encryption keys to your key vault. Managing encryption keys in your key vault requires Azure AD authentication. Create an Azure AD application for this purpose. For authentication purposes, you can use either client secret-based authentication or [client certificate-based Azure AD authentication](#).

Set up an Azure AD app and service principal with Azure PowerShell

To execute the following commands, get and use the [Azure AD PowerShell module](#).

1. If needed, [connect to your Azure subscription](#).
2. Use the [New-AzureRmADApplication](#) PowerShell cmdlet to create an Azure AD application.
MyApplicationHomePage and the MyApplicationUri can be any values you wish.

```
$aadClientSecret = "My AAD client secret"  
$aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -AsPlainText -Force  
$azureAdApplication = New-AzureRmADApplication -DisplayName "My Application Display Name" -HomePage  
"https://MyApplicationHomePage" -IdentifierUris "https://MyApplicationUri" -Password  
$aadClientSecretSec  
$servicePrincipal = New-AzureRmADServicePrincipal -ApplicationId $azureAdApplication.ApplicationId
```

3. The \$azureAdApplication.ApplicationId is the Azure AD ClientID and the \$aadClientSecret is the client secret that you will use later to enable Azure Disk Encryption. Safeguard the Azure AD client secret appropriately. Running `$azureAdApplication.ApplicationId` will show you the ApplicationID.

Set up an Azure AD app and service principal with Azure CLI

You can manage your service principals with Azure CLI using the [az ad sp](#) commands. For more information, see [Create an Azure service principal](#).

1. If needed, [connect to your Azure subscription](#).

2. Create a new service principal.

```
az ad sp create-for-rbac --name "ServicePrincipalName" --password "My-AAD-client-secret" --skip-assignment
```

3. The appId returned is the Azure AD ClientID used in other commands. It's also the SPN you'll use for az keyvault set-policy. The password is the client secret that you should use later to enable Azure Disk Encryption. Safeguard the Azure AD client secret appropriately.

Set up an Azure AD app and service principal though the Azure portal

Use the steps from the [Use portal to create an Azure Active Directory application and service principal that can access resources](#) article to create an Azure AD application. Each step listed below will take you directly to the article section to complete.

1. Verify required permissions

2. Create an Azure Active Directory application

- You can use any name and sign-on URL you would like when creating the application.

3. Get the application ID and the authentication key.

- The authentication key is the client secret and is used as the AadClientSecret for Set-AzureRmVMDiskEncryptionExtension.
 - The authentication key is used by the application as a credential to sign in to Azure AD. In the Azure portal, this secret is called keys, but has no relation to key vaults. Secure this secret appropriately.
- The application ID will be used later as the AadClientId for Set-AzureRmVMDiskEncryptionExtension and as the ServicePrincipalName for Set-AzureRmKeyVaultAccessPolicy.

Set the key vault access policy for the Azure AD app

To write encryption secrets to a specified Key Vault, Azure Disk Encryption needs the Client ID and the Client Secret of the Azure Active Directory application that has permissions to write secrets to the Key Vault.

NOTE

Azure Disk Encryption requires you to configure the following access policies to your Azure AD client application: *WrapKey* and *Set* permissions.

Set the key vault access policy for the Azure AD app with Azure PowerShell

Your Azure AD application needs rights to access the keys or secrets in the vault. Use the [Set-AzureKeyVaultAccessPolicy](#) cmdlet to grant permissions to the application, using the client ID (which was generated when the application was registered) as the *-ServicePrincipalName* parameter value. To learn more, see the blog post [Azure Key Vault - Step by Step](#).

1. If needed, [connect to your Azure subscription](#).
2. Set the key vault access policy for the AD application with PowerShell.

```
$keyVaultName = 'MySecureVault'  
$aadClientID = 'MyAadAppClientID'  
$rgname = 'MySecureRG'  
Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -  
PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $rgname
```

Set the key vault access policy for the Azure AD app with Azure CLI

Use [az keyvault set-policy](#) to set the access policy. For more information, see [Manage Key Vault using CLI 2.0](#).

1. If needed, [connect to your Azure subscription](#).
2. Give the service principal you created via the Azure CLI access to get secrets and wrap keys with the following command:

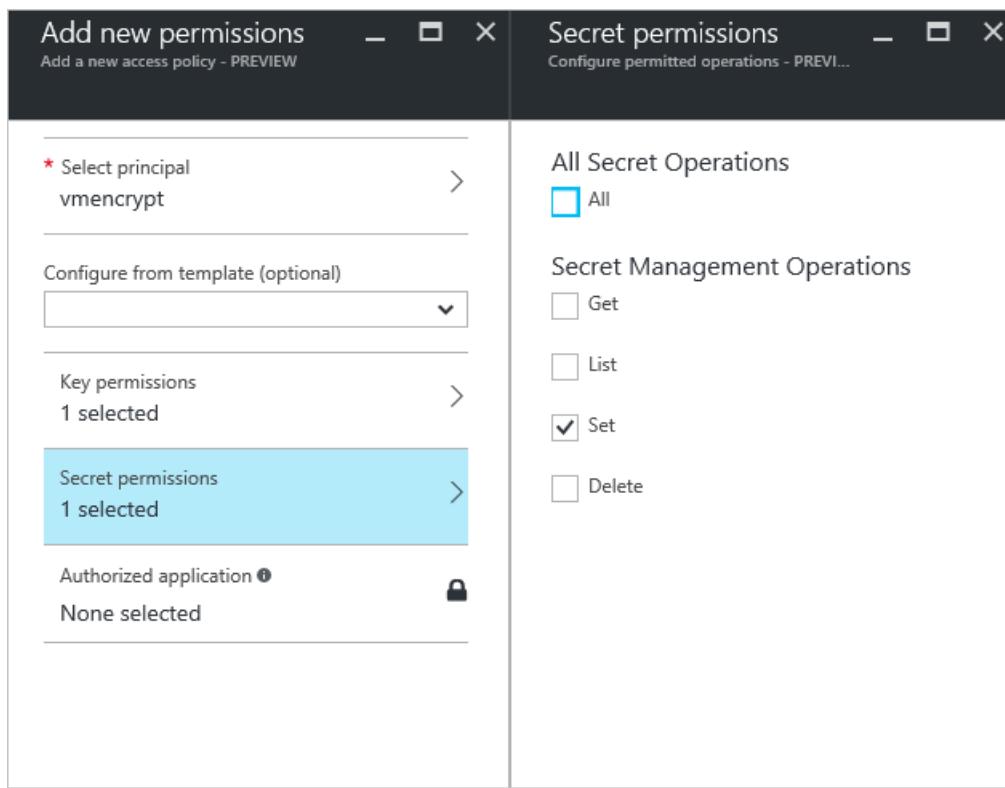
```
az keyvault set-policy --name "MySecureVault" --spn "<spn created with CLI/the Azure AD ClientID>" --key-permissions wrapKey --secret-permissions set
```

Set the key vault access policy for the Azure AD app with the portal

1. Open the resource group with your key vault.
2. Select your key vault, go to **Access Policies**, then click **Add new**.
3. Under **Select principal**, search for the Azure AD application you created and select it.
4. For **Key permissions**, check **Wrap Key** under **Cryptographic Operations**.
5. For **Secret permissions**, check **Set** under **Secret Management Operations**.
6. Click **OK** to save the access policy.

The screenshot shows two overlapping dialog boxes from the Azure portal:

- Add new permissions** (left):
 - Select principal**: A dropdown menu showing "vmencrypt".
 - Configure from template (optional)**: A dropdown menu.
 - Key permissions**: A section titled "1 selected".
 - Secret permissions**: A section titled "1 selected".
 - Authorized application**: A section titled "None selected".
- Key permissions** (right):
 - All Key Operations**: An expanded section showing "All" selected.
 - Key Management Operations**: A list of operations:
 - Get
 - List
 - Update
 - Create
 - Import
 - Delete
 - Backup
 - Restore
 - Cryptographic Operations**: A list of operations:
 - Decrypt
 - Encrypt
 - UnwrapKey
 - WrapKey
 - Verify
 - Sign



Set key vault advanced access policies

The Azure platform needs access to the encryption keys or secrets in your key vault to make them available to the VM for booting and decrypting the volumes. Enable disk encryption on the key vault or deployments will fail.

Set key vault advanced access policies with Azure PowerShell

Use the key vault PowerShell cmdlet [Set-AzureRmKeyVaultAccessPolicy](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** EnabledForDiskEncryption is required for Azure Disk encryption.

```
Set-AzureRmKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -  
EnabledForDiskEncryption
```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

```
Set-AzureRmKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -  
EnabledForDeployment
```

- **Enable Key Vault for template deployment, if needed:** Enables Azure Resource Manager to get secrets from this key vault when this key vault is referenced in a template deployment.

```
Set-AzureRmKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName 'MySecureRG' -  
EnabledForTemplateDeployment
```

Set key vault advanced access policies using the Azure CLI

Use [az keyvault update](#) to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** Enabled-for-disk-encryption is required.

```
az keyvault update --name "MySecureVault" --resource-group "MySecureRG" --enabled-for-disk-encryption "true"
```

- **Enable Key Vault for deployment, if needed:** Allow Virtual Machines to retrieve certificates stored as secrets from the vault.

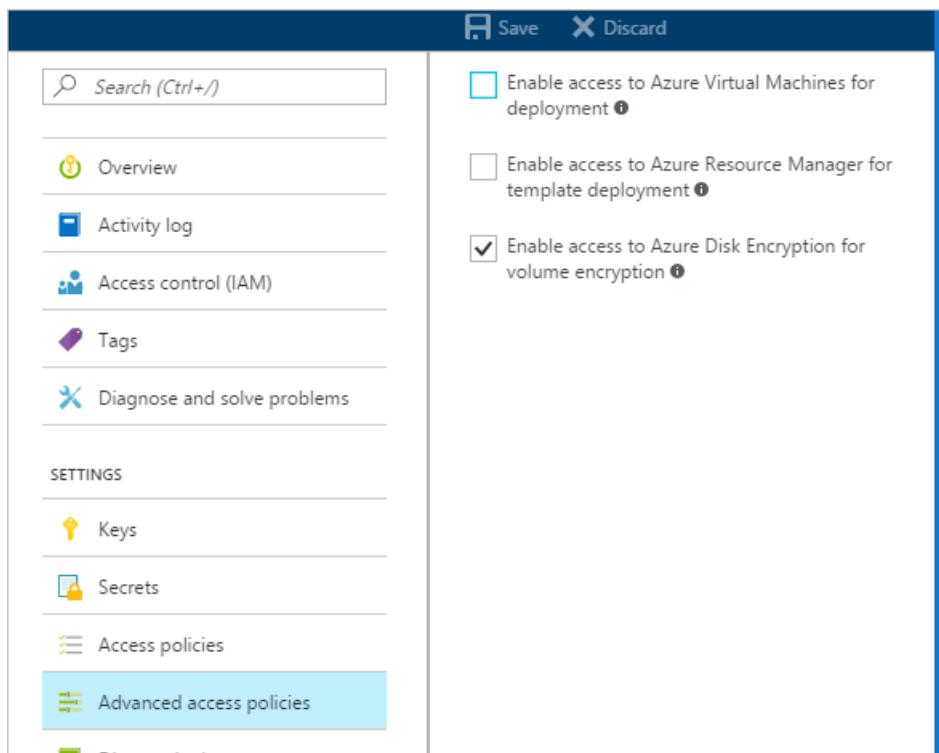
```
az keyvault update --name "MySecureVault" --resource-group "MySecureRG" --enabled-for-deployment "true"
```

- **Enable Key Vault for template deployment, if needed:** Allow Resource Manager to retrieve secrets from the vault.

```
az keyvault update --name "MySecureVault" --resource-group "MySecureRG" --enabled-for-template-deployment "true"
```

Set key vault advanced access policies through the Azure portal

1. Select your keyvault, go to **Access Policies**, and [Click to show advanced access policies](#).
2. Select the box labeled **Enable access to Azure Disk Encryption for volume encryption**.
3. Select **Enable access to Azure Virtual Machines for deployment** and/or **Enable Access to Azure Resource Manager for template deployment**, if needed.
4. Click **Save**.



Set up a key encryption key (optional)

If you want to use a key encryption key (KEK) for an additional layer of security for encryption keys, add a KEK to your key vault. Use the [Add-AzureKeyVaultKey](#) cmdlet to create a key encryption key in the key vault. You can also import a KEK from your on-premises key management HSM. For more information, see [Key Vault Documentation](#). When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

- Your key vault secret and KEK URLs must be versioned. Azure enforces this restriction of versioning. For valid secret and KEK URLs, see the following examples:

- Example of a valid secret URL:
<https://contosovault.vault.azure.net/secrets/EncryptionSecretWithKek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - Example of a valid KEK URL:
<https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
- Azure Disk Encryption doesn't support specifying port numbers as part of key vault secrets and KEK URLs. For examples of non-supported and supported key vault URLs, see the following examples:
 - Unacceptable key vault URL
<https://contosovault.vault.azure.net:443/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - Acceptable key vault URL:
<https://contosovault.vault.azure.net/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Set up a key encryption key with Azure PowerShell

Before using the PowerShell script, you should be familiar with the Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment. This script creates all Azure Disk Encryption prerequisites and encrypts an existing IaaS VM, wrapping the disk encryption key by using a key encryption key.

```

# Step 1: Create a new resource group and key vault in the same location.
# Fill in 'MyLocation', 'MySecureRG', and 'MySecureVault' with your values.
# Use Get-AzureRmLocation to get available locations and use the DisplayName.
# To use an existing resource group, comment out the line for New-AzureRmResourceGroup

$Loc = 'MyLocation';
$rgname = 'MySecureRG';
$keyVaultName = 'MySecureVault';
New-AzureRmResourceGroup -Name $rgname -Location $Loc;
New-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname -Location $Loc;
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$keyVaultResourceId = (Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName
$rgname).ResourceId;
$diskEncryptionKeyVaultUrl = (Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName
$rgname).VaultUri;

# Step 2: Create the AD application and service principal.
# Fill in 'MyAADClientSecret', "<My Application Display Name>", "<https://MyApplicationHomePage>", and "
<https://MyApplicationUri>" with your values.
# MyApplicationHomePage and the MyApplicationUri can be any values you wish.

$aadClientSecret = 'MyAADClientSecret';
$aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -AsPlainText -Force;
$azureAdApplication = New-AzureRmADApplication -DisplayName "<My Application Display Name>" -HomePage "
<https://MyApplicationHomePage>" -IdentifierUris '<https://MyApplicationUri>' -Password $aadClientSecretSec
$servicePrincipal = New-AzureRmADServicePrincipal -ApplicationId $azureAdApplication.ApplicationId;
$aadClientID = $azureAdApplication.ApplicationId;

#Step 3: Enable the vault for disk encryption and set the access policy for the Azure AD application.

Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ResourceGroupName $rgname -
EnabledForDiskEncryption;
Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -
PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $rgname;

#Step 4: Create a new key in the key vault with the Add-AzureKeyVaultKey cmdlet.
# Fill in 'MyKeyEncryptionKey' with your value.

$keyEncryptionKeyName = 'MyKeyEncryptionKey';
Add-AzureKeyVaultKey -VaultName $keyVaultName -Name $keyEncryptionKeyName -Destination 'Software';
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

#Step 5: Encrypt the disks of an existing IaaS VM
# Fill in 'MySecureVM' with your value.

$vmName = 'MySecureVM';
Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID $aadClientID
-AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyId $keyVaultResourceId;

```

Certificate-based authentication (optional)

If you would like to use certificate authentication, you can upload one to your key vault and deploy it to the client. Before using the PowerShell script, you should be familiar with the Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

```

# Fill in "MySecureRG", "MySecureVault", and 'MyLocation' ('My location' only if needed)

$rgname = 'MySecureRG'
$keyVaultName= 'MySecureVault'

```

```

# Create a key vault and set enabledForDiskEncryption property on it.
# Comment out the next three lines if you already have an existing key vault enabled for encryption. No need
to set 'My location' in this case.

$Loc = 'MyLocation'
New-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $rgname -Location $Loc
Set-AzureRmKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $rgname -
EnabledForDiskEncryption

#Setting some variables with the key vault information
$keyVault = Get-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $rgname
$DiskEncryptionKeyVaultUrl = $keyVault.VaultUri
$keyVaultResourceId = $keyVault.ResourceId

# Create the Azure AD application and associate the certificate with it.
# Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display Name>", "
<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with your values.
# MyApplicationHomePage and the MyApplicationUri can be any values you wish

$CertPath = "C:\certificates\mycert.pfx"
$CertPassword = "Password"
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath, $CertPassword)
$CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

$AzureAdApplication = New-AzureRmADApplication -DisplayName "<My Application Display Name>" -HomePage "
<https://MyApplicationHomePage>" -IdentifierUris "<https://MyApplicationUri>" -CertValue $CertValue
$ServicePrincipal = New-AzureRmADServicePrincipal -ApplicationId $AzureAdApplication.ApplicationId

$AADClientID = $AzureAdApplication.ApplicationId
$aadClientCertThumbprint= $cert.Thumbprint

Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -
PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $rgname

# Upload the pfx file to the key vault.
# Fill in "MyAADCert".

$keyVaultSecretName = "MyAADCert"
$fileContentBytes = get-content $CertPath -Encoding Byte
$fileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
$jsonObject = @{
    "data" : "$fileContentEncoded",
    "dataType" : "pfx",
    "password" : "$CertPassword"
}
@

$jsonObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
$jsonEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

#Set the secret and set the key vault policy for -EnabledForDeployment

$secret = ConvertTo-SecureString -String $jsonEncoded -AsPlainText -Force
Set-AzureKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName -SecretValue $secret
Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ResourceGroupName $rgname -EnabledForDeployment

# Deploy the certificate to the VM
# Fill in 'MySecureVM' with your value.

$vmName = 'MySecureVM'
$certUrl = (Get-AzureKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName).Id
$sourceVaultId = (Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname).ResourceId
$vm = Get-AzureRmVM -ResourceGroupName $rgname -Name $vmName
$vm = Add-AzureRmVMSecret -VM $vm -SourceVaultId $sourceVaultId -CertificateStore "My" -CertificateUrl
$certUrl
Update-AzureRmVM -VM $vm -ResourceGroupName $rgname

#Enable encryption on the VM using Azure AD client ID and the client certificate thumbprint

```

```
Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $VMName -AadClientID $AADClientID -  
AadClientCertThumbprint $AADClientCertThumbprint -DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -  
DiskEncryptionKeyId $KeyVaultResourceId
```

Certificate-based authentication and a KEK (optional)

If you would like to use certificate authentication and wrap the encryption key with a KEK, you can use the below script as an example. Before using the PowerShell script, you should be familiar with all of the previous Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

IMPORTANT

Azure AD certificate-based authentication is currently not supported on Linux VMs.

```
# Fill in 'MySecureRG', 'MySecureVault', and 'MyLocation' (if needed)  
  
$rgname = 'MySecureRG'  
$KeyVaultName= 'MySecureVault'  
  
# Create a key vault and set enabledForDiskEncryption property on it.  
# Comment out the next three lines if you already have an existing key vault enabled for encryption.  
  
$Loc = 'MyLocation'  
New-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $rgname -Location $Loc  
Set-AzureRmKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $rgname -  
EnabledForDiskEncryption  
  
# Create the Azure AD application and associate the certificate with it.  
# Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display Name>", "  
<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with your values.  
# MyApplicationHomePage and the MyApplicationUri can be any values you wish  
  
$CertPath = "C:\certificates\mycert.pfx"  
$CertPassword = "Password"  
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath, $CertPassword)  
$CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())  
  
$AzureAdApplication = New-AzureRmADApplication -DisplayName "<My Application Display Name>" -HomePage "  
<https://MyApplicationHomePage>" -IdentifierUris "<https://MyApplicationUri>" -CertValue $CertValue  
$ServicePrincipal = New-AzureRmADServicePrincipal -ApplicationId $AzureAdApplication.ApplicationId  
  
$AADClientID = $AzureAdApplication.ApplicationId  
$aadClientCertThumbprint= $cert.Thumbprint  
  
## Give access for setting secrets and wrapping keys  
Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -  
PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $rgname  
  
# Upload the pfx file to the key vault.  
# Fill in "MyAADCert".  
  
$KeyVaultSecretName = "MyAADCert"  
$FileContentBytes = get-content $CertPath -Encoding Byte  
$FileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)  
$JSONObject = @"  
{  
    "data" : "$filecontentencoded",  
    "dataType" : "pfx",  
    "password" : "$CertPassword"  
}  
"@
```

```

$JSONObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
$JSONEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

#Set the secret and set the key vault policy for deployment

$Secret = ConvertTo-SecureString -String $JSONEncoded -AsPlainText -Force
Set-AzureKeyVaultSecret -VaultName $KeyVaultName -Name $KeyVaultSecretName -SecretValue $Secret
Set-AzureRmKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $rgname -EnabledForDeployment

#Setting some variables with the key vault information and generating a KEK
# Fill in 'KEKName'

$KEKName = 'KEKName'
$keyVault = Get-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $rgname
$DiskEncryptionKeyVaultUrl = $keyVault.VaultUri
$keyVaultResourceId = $keyVault.ResourceId
$kek = Add-AzureKeyVaultKey -VaultName $KeyVaultName -Name $KEKName -Destination "Software"
$keyEncryptionKeyId = $kek.Key.kid

# Deploy the certificate to the VM
# Fill in 'MySecureVM' with your value.

$VMName = 'MySecureVM'
$certUrl = (Get-AzureKeyVaultSecret -VaultName $KeyVaultName -Name $KeyVaultSecretName).Id
$sourceVaultId = (Get-AzureRmKeyVault -VaultName $KeyVaultName -ResourceGroupName $rgname).ResourceId
$VM = Get-AzureRmVM -ResourceGroupName $rgname -Name $VMName
$VM = Add-AzureRmVMSecret -VM $VM -SourceVaultId $sourceVaultId -CertificateStore "My" -CertificateUrl
$certUrl
Update-AzureRmVM -VM $VM -ResourceGroupName $rgname

#Enable encryption on the VM using Azure AD client ID and the client certificate thumbprint

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $VMName -AadClientID $AADClientID -
AadClientCertThumbprint $AADClientCertThumbprint -DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyId $keyEncryptionKeyIdUrl -
KeyEncryptionKeyId $keyVaultResourceId

```

Next steps

[Enable Azure Disk Encryption for Windows](#)

[Enable Azure Disk Encryption for Linux](#)

Enable Azure Disk Encryption for Windows IaaS VMs (previous release)

12/7/2018 • 16 minutes to read • [Edit Online](#)

The new release of Azure Disk Encryption eliminates the requirement for providing an Azure AD application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Azure AD credentials during the enable encryption step. All new VMs must be encrypted without the Azure AD application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see [Azure Disk Encryption for Windows VMs](#). VMs that were already encrypted with Azure AD application parameters are still supported and should continue to be maintained with the AAD syntax.

You can enable many disk-encryption scenarios, and the steps may vary according to the scenario. The following sections cover the scenarios in greater detail for Windows IaaS VMs. Before you can use disk encryption, the [Azure Disk Encryption prerequisites](#) need to be completed.

Take a [snapshot](#) and/or back up before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the `Set-AzureRmVmDiskEncryptionExtension` cmdlet to encrypt managed disks by specifying the `-skipVmBackup` parameter. For more information about how to back up and restore encrypted VMs, see the [Azure Backup](#) article.

WARNING

In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

Enable encryption on new IaaS VMs created from the Marketplace

You can enable disk encryption on new IaaS Windows VM from the Marketplace in Azure using a Resource Manager template. The template creates a new encrypted Windows VM using the Windows Server 2012 gallery image.

1. On the [Resource Manager template](#), click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Purchase** to deploy a new IaaS VM where encryption is enabled.
3. After you deploy the template, verify the VM encryption status using your preferred method:
 - Verify with the Azure CLI by using the `az vm encryption show` command.

```
az vm encryption show --name "MySecureVM" --resource-group "MySecureRg"
```

- Verify with Azure PowerShell by using the `Get-AzureRmVmDiskEncryptionStatus` cmdlet.

```
Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName 'MySecureRg' -VMName 'MySecureVM'
```

- Select the VM, then click on **Disks** under the **Settings** heading to verify encryption status in the portal. In the chart under **Encryption**, you'll see if it's enabled.

NAME	STATUS	LOCATION	SIZE	DISK ENCRYPTION
ADEDemoCAT	Running	Australia East	Standard_D1	Enabled
ADEPreDemoCAT	Running	Australia East	Standard_D1	Enabled
at-east	Running	East US	Standard_A1	Enabled
at-prevm10	Running	Australia East	Standard_D2	Enabled

The following table lists the Resource Manager template parameters for new VMs from the Marketplace scenario using Azure AD client ID:

PARAMETER	DESCRIPTION
adminUserName	Admin user name for the virtual machine.
adminPassword	Admin user password for the virtual machine.
newStorageAccountName	Name of the storage account to store OS and data VHDs.
vmSize	Size of the VM. Currently, only Standard A, D, and G series are supported.
virtualNetworkName	Name of the VNet that the VM NIC should belong to.
subnetName	Name of the subnet in the VNet that the VM NIC should belong to.
AADClientID	Client ID of the Azure AD application that has permissions to write secrets to your key vault.
AADClientSecret	Client secret of the Azure AD application that has permissions to write secrets to your key vault.
keyVaultURL	<p>URL of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet</p> <pre>(Get-AzureRmKeyVault -VaultName, -ResourceGroupName).VaultURI</pre> <p>or the Azure CLI</p> <pre>az keyvault show --name "MySecureVault" --query properties.vaultUri</pre>
keyEncryptionKeyURL	<p>URL of the key encryption key that's used to encrypt the generated BitLocker key (optional).</p> <p>KeyEncryptionKeyURL is an optional parameter. You can bring your own KEK to further safeguard the data encryption key (Passphrase secret) in your key vault.</p>
keyVaultResourceGroup	Resource group of the key vault.

PARAMETER	DESCRIPTION
vmName	Name of the VM that the encryption operation is to be performed on.

Enable encryption on existing or running IaaS Windows VMs

In this scenario, you can enable encryption by using a template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail how to enable Azure Disk Encryption.

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the `Set-AzureRmVMDiskEncryptionExtension` cmdlet can be used to encrypt managed disks by specifying the `-skipVmBackup` parameter. The `Set-AzureRmVMDiskEncryptionExtension` command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Enable encryption on existing or running VMs with Azure PowerShell

Use the `Set-AzureRmVMDiskEncryptionExtension` cmdlet to enable encryption on a running IaaS virtual machine in Azure. For information about enabling encryption with Azure Disk Encryption by using PowerShell cmdlets, see the blog posts [Explore Azure Disk Encryption with Azure PowerShell - Part 1](#) and [Explore Azure Disk Encryption with Azure PowerShell - Part 2](#).

- **Encrypt a running VM using a client secret:** The script below initializes your variables and runs the `Set-AzureRmVMDiskEncryptionExtension` cmdlet. The resource group, VM, key vault, AAD app, and client secret should have already been created as prerequisites. Replace `MySecureRg`, `MySecureVM`, `MySecureVault`, `My-AAD-client-ID`, and `My-AAD-client-secret` with your values.

```
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyId $keyVaultResourceId;
```

- **Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

```

$rgName = 'MySecureRg';
$vmName = 'MyExtraSecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [Get-AzureRmVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName 'MySecureRg' -VMName 'MySecureVM'
```

- Disable disk encryption:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM'
```

Enable encryption on existing or running VMs with Azure CLI

Use the [az vm encryption enable](#) command to enable encryption on a running IaaS virtual machine in Azure.

- Encrypt a running VM using a client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --volume-type [All|OS|Data]
```

- Encrypt a running VM using KEK to wrap the client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI which is the Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [az vm encryption show](#) command.

```
az vm encryption show --name "MySecureVM" --resource-group "MySecureRg"
```

- **Disable encryption:** To disable encryption, use the [az vm encryption disable](#) command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type [ALL, DATA, OS]
```

NOTE

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzureRmVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. This command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Using the Resource Manager template

You can enable disk encryption on existing or running IaaS Windows VMs in Azure by using the [Resource Manager template to encrypt a running Windows VM](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Purchase** to enable encryption on the existing or running IaaS VM.

The following table lists the Resource Manager template parameters for existing or running VMs that use an Azure AD client ID:

PARAMETER	DESCRIPTION
AADClientID	Client ID of the Azure AD application that has permissions to write secrets to the key vault.
AADClientSecret	Client secret of the Azure AD application that has permissions to write secrets to the key vault.
keyVaultName	Name of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet <pre>(Get-AzureRmKeyVault -ResourceGroupName <MyResourceGroupName>).Vaultname</pre> or the Azure CLI command `az keyvault list --resource-group "MySecureGroup"

PARAMETER	DESCRIPTION
keyEncryptionKeyURL	URL of the key encryption key that's used to encrypt the generated BitLocker key. This parameter is optional if you select nokek in the UseExistingKek drop-down list. If you select kek in the UseExistingKek drop-down list, you must enter the <i>keyEncryptionKeyURL</i> value.
volumeType	Type of volume that the encryption operation is performed on. Valid values are <i>OS</i> , <i>Data</i> , and <i>All</i> .
sequenceVersion	Sequence version of the BitLocker operation. Increment this version number every time a disk-encryption operation is performed on the same VM.
vmName	Name of the VM that the encryption operation is to be performed on.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using the Resource Manager template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail the Resource Manager template and CLI commands.

Use the instructions in the appendix for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- [Prepare a pre-encrypted Windows VHD](#)
- [Prepare a pre-encrypted Linux VHD](#)

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzureRmVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. The Set-AzureRmVMDiskEncryptionExtension command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Encrypt VMs with pre-encrypted VHDS with Azure PowerShell

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet [Set-AzureRmVMOSDisk](#). The example below gives you some common parameters.

```
$VirtualMachine = New-AzureRmVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzureRmVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri "os.vhd" Caching
ReadWrite -Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa" -DiskEncryptionKeyVaultId
"/subscriptions/00000000-0000-0000-0000-
00000000/resourceGroups/myresourcegroup/providers/Microsoft.KeyVault/vaults/mytestvault"
New-AzureRmVM -VM $VirtualMachine -ResourceGroupName "MySecureRG"
```

Encrypt IaaS VMs with pre-encrypted VHDS with a Resource Manager template

You can enable disk encryption on your encrypted VHD by using the [Resource Manager template](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Create** to enable encryption on the new IaaS VM.

The following table lists the Resource Manager template parameters for your encrypted VHD:

PARAMETER	DESCRIPTION
newStorageAccountName	Name of the storage account to store the encrypted OS VHD. This storage account should already have been created in the same resource group and same location as the VM.
osVhdUri	URI of the OS VHD from the storage account.
osType	OS product type (Windows/Linux).
virtualNetworkName	Name of the VNet that the VM NIC should belong to. The name should already have been created in the same resource group and same location as the VM.
subnetName	Name of the subnet on the VNet that the VM NIC should belong to.
vmSize	Size of the VM. Currently, only Standard A, D, and G series are supported.
keyVaultResourceId	<p>The ResourceID that identifies the key vault resource in Azure Resource Manager. You can get it by using the PowerShell cmdlet</p> <pre>(Get-AzureRmKeyVault -VaultName &lt;MyKeyVaultName&gt; -ResourceGroupName &lt;MyResourceGroupName&gt;).ResourceId</pre> <p>or using the Azure CLI command</p> <pre>az keyvault show --name "MySecureVault" --query id</pre>
keyVaultSecretUrl	URL of the disk-encryption key that's set up in the key vault.
keyVaultKekUrl	URL of the key encryption key for encrypting the generated disk-encryption key.
vmName	Name of the IaaS VM.

Enable encryption on a newly added data disk

You can [add a new disk to a Windows VM using PowerShell](#), or [through the Azure portal](#).

Enable encryption on a newly added disk with Azure PowerShell

When using Powershell to encrypt a new disk for Windows VMs, a new sequence version should be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version. In some cases, a newly added data disk might be encrypted automatically by the Azure Disk Encryption extension. Auto encryption usually occurs when the VM reboots after the new disk comes online. This is typically caused because "All" was specified for the volume type when disk encryption previously ran on the VM. If auto encryption occurs on a newly added data disk, we recommend running the Set-AzureRmVmDiskEncryptionExtension cmdlet again with new sequence version. If your new data disk is auto encrypted and you do not wish to be encrypted, decrypt all drives first then re-encrypt with a new sequence version specifying OS for the volume type.

- Encrypt a running VM using a client secret:** The script below initializes your variables and runs the Set-AzureRmVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, AAD app, and client secret should have already been created as prerequisites. Replace MySecureRg, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values. This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

```
$sequenceVersion = [Guid]::.NewGuid();
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -VolumeType 'all' -SequenceVersion $sequenceVersion;
```

- Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault. This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

```
$sequenceVersion = [Guid]::.NewGuid();
$rgName = 'MySecureRg';
$vmName = 'MyExtraSecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId -VolumeType 'all' -SequenceVersion $sequenceVersion;
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-
name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: [https://\[keyvault-name\].vault.azure.net/keys/\[kekname\]/\[kek-unique-id\]](https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id])

Enable encryption on a newly added disk with Azure CLI

The Azure CLI command will automatically provide a new sequence version for you when you run the command to enable encryption. Acceptable values for the volume-type parameter are All, OS, and Data. You may need to

change the volume-type parameter to OS or Data if you're only encrypting one type of disk for the VM. The examples use "All" for the volume-type parameter.

- **Encrypt a running VM using a client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --volume-type "All"
```

- **Encrypt a running VM using KEK to wrap the client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI which is the Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type "all"
```

Enable encryption using Azure AD client certificate-based authentication.

You can use client certificate authentication with or without KEK. The scripts require that [Azure Disk Encryption prerequisites](#) are complete. Before using the PowerShell scripts, you should already have the certificate uploaded to the key vault and deployed to the VM. If you're using KEK too, the KEK should already exist. For more information, see the [Certificate-based authentication for Azure AD](#) section of the prerequisites article.

Enable encryption using certificate-based authentication with Azure PowerShell

```
## Fill in 'MySecureRg', 'My-AAD-client-ID', 'MySecureVault', and 'MySecureVM' .  
  
$rgName = 'MySecureRg';  
$AADClientID ='My-AAD-client-ID';  
$KeyVaultName = 'MySecureVault';  
$VMName = 'MySecureVM';  
$diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;  
$KeyVaultResourceId = $KeyVault.ResourceId;  
  
# Fill in the certificate path and the password so the thumbprint can be set as a variable.  
  
$certPath = '$CertPath = "C:\certificates\mycert.pfx";  
$CertPassword = 'Password'  
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath, $CertPassword)  
$aadClientCertThumbprint = $cert.Thumbprint;  
  
# Enable disk encryption using the client certificate thumbprint  
  
Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $VMName -AadClientID $AADClientID -  
AadClientCertThumbprint $aadClientCertThumbprint -DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -  
DiskEncryptionKeyId $KeyVaultResourceId
```

Enable encryption using certificate-based authentication and a KEK with Azure PowerShell

```

# Fill in 'MySecureRg', 'My-AAD-client-ID', 'MySecureVault,, 'MySecureVM', and "KEKName.

$rgName = 'MySecureRg';
$AADClientID ='My-AAD-client-ID';
$keyVaultName = 'MySecureVault';
$VMName = 'MySecureVM';
$keyEncryptionKeyName ='KEKName';
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name $keyEncryptionKeyName).Key.kid;

## Fill in the certificate path and the password so the thumbprint can be read and set as a variable.

$certPath = '$CertPath = "C:\certificates\mycert.pfx";
$CertPassword = 'Password'
$Cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($certPath, $CertPassword)
$aadClientCertThumbprint = $cert.Thumbprint;

# Enable disk encryption using the client certificate thumbprint and a KEK

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $VMName -AadClientID $AADClientID -
AadClientCertThumbprint $aadClientCertThumbprint -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId

```

Disable encryption

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template.

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM'
```

- **Disable encryption with the Azure CLI:** To disable encryption, use the [az vm encryption disable](#) command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type [ALL, DATA, OS]
```

- **Disable encryption with a Resource Manager Template:**

1. Click **Deploy to Azure** from the [Disable disk encryption on running Windows VM](#) template.
2. Select the subscription, resource group, location, VM, legal terms, and agreement.
3. Click **Purchase** to disable disk encryption on a running Windows VM.

Next steps

[Enable Azure Disk Encryption for Linux](#)

Enable Azure Disk Encryption for Linux IaaS VMs (previous release)

12/7/2018 • 18 minutes to read • [Edit Online](#)

The new release of Azure Disk Encryption eliminates the requirement for providing an Azure AD application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Azure AD credentials during the enable encryption step. All new VMs must be encrypted without the Azure AD application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see [Azure Disk Encryption for Linux VMs](#). VMs that were already encrypted with Azure AD application parameters are still supported and should continue to be maintained with the AAD syntax.

You can enable many disk-encryption scenarios, and the steps may vary according to the scenario. The following sections cover the scenarios in greater detail for Linux IaaS VMs. Before you can use disk encryption, the [Azure Disk Encryption prerequisites](#) need to be completed and the [Additional prerequisites for Linux IaaS VMs](#) section should be reviewed.

Take a [snapshot](#) and/or back up before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the Set-AzureRmVMDiskEncryptionExtension cmdlet to encrypt managed disks by specifying the -skipVmBackup parameter. For more information about how to back up and restore encrypted VMs, see the [Azure Backup](#) article.

WARNING

In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

When encrypting Linux OS volumes, the process can take a few hours. It is normal for Linux OS volumes to take longer than data volumes to encrypt.

Disabling encryption on Linux VMs is only supported for data volumes. It is not supported on data or OS volumes if the OS volume has been encrypted.

Deploy a new Linux IaaS VM with disk encryption enabled

1. Use the [Resource Manager template](#) to create a new encrypted Linux IaaS VM. The template will create a new RedHat Linux 7.2 VM with a 200-GB RAID-0 array and full disk encryption using managed disks. On the [FAQ](#) article, you'll notice that some Linux distributions only support encryption for data disks. However, this template provides an opportunity to become familiar with deploying templates and verifying encryption status with multiple methods.
2. Click **Deploy to Azure** on the Azure Resource Manager template.
3. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Create** to enable encryption on the existing or running IaaS VM.
4. After you deploy the template, verify the VM encryption status using your preferred method:
 - Verify with the Azure CLI by using the `az vm encryption show` command.

```
az vm encryption show --name "MySecureVM" --resource-group "MySecureRg"
```

- Verify with Azure PowerShell by using the [Get-AzureRmVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName 'MySecureRg' -VMName 'MySecureVM'
```

- Select the VM, then click on **Disks** under the **Settings** heading to verify encryption status in the portal. In the chart under **Encryption**, you'll see if it's enabled.

PARAMETER	DESCRIPTION
AAD Client ID	Client ID of the Azure AD application that has permissions to write secrets to the key vault.
AAD Client Secret	Client secret of the Azure AD application that has permissions to write secrets to your key vault.
Key Vault Name	Name of the key vault that the key should be placed.
Key Vault Resource Group	Resource group of the key vault.

Enable encryption on an existing or running IaaS Linux VM

In this scenario, you can enable encryption by using the Resource Manager template, PowerShell cmdlets, or CLI commands.

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the `Set-AzureRmVMDiskEncryptionExtension` cmdlet can be used to encrypt managed disks by specifying the `-skipVmBackup` parameter. The `Set-AzureRmVMDiskEncryptionExtension` command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Enable encryption on an existing or running Linux VM using Azure CLI

You can enable disk encryption on your encrypted VHD by installing and using The [Azure CLI 2.0](#) command-line tool. You can use it in your browser with [Azure Cloud Shell](#), or you can install it on your local machine and use it in any PowerShell session. To enable encryption on existing or running IaaS Linux VMs in Azure, use the following CLI commands:

Use the `az vm encryption enable` command to enable encryption on a running IaaS virtual machine in Azure.

- Encrypt a running VM using a client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --volume-type [All|OS|Data]
```

- Encrypt a running VM using KEK to wrap the client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI which is the Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [az vm encryption show](#) command.

```
az vm encryption show --name "MySecureVM" --resource-group "MySecureRg"
```

- **Disable encryption:** To disable encryption, use the [az vm encryption disable](#) command. Disabling encryption is only allowed on data volumes for Linux VMs.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type DATA
```

Enable encryption on an existing or running Linux VM using PowerShell

Use the [Set-AzureRmVMDiskEncryptionExtension](#) cmdlet to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM using a client secret:** The script below initializes your variables and runs the Set-AzureRmVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, AAD app, and client secret should have already been created as prerequisites. Replace MySecureRg, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values. You may need to add the -VolumeType parameter if you're encrypting data disks and not the OS disk.

```
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyId $keyVaultResourceId;
```

- **Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault. You may need to add the -VolumeType parameter if you're encrypting data disks and not the OS disk.

```

$rgName = 'MySecureRg';
$vmName = 'MyExtraSecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId;

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string:

```
/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-
name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
```

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-
name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- Verify the disks are encrypted:** To check on the encryption status of a IaaS VM, use the [Get-AzureRmVmDiskEncryptionStatus](#) cmdlet.

```
Get-AzureRmVmDiskEncryptionStatus -ResourceGroupName MySecureRg -VMName MySecureVM
```

- Disable disk encryption:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet. Disabling encryption is only allowed on data volumes for Linux VMs.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM'
```

Enable encryption on an existing or running IaaS Linux VM with a template

You can enable disk encryption on an existing or running IaaS Linux VM in Azure by using the [Resource Manager template](#).

- Click **Deploy to Azure** on the Azure quickstart template.
- Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Create** to enable encryption on the existing or running IaaS VM.

The following table lists Resource Manager template parameters for existing or running VMs that use an Azure AD client ID:

PARAMETER	DESCRIPTION
AADClientID	Client ID of the Azure AD application that has permissions to write secrets to the key vault.
AADClientSecret	Client secret of the Azure AD application that has permissions to write secrets to your key vault.

PARAMETER	DESCRIPTION
keyVaultName	<p>Name of the key vault that the key should be uploaded to. You can get it by using the Azure CLI command</p> <pre>az keyvault show --name "MySecureVault" --query resourceGroup</pre>
keyEncryptionKeyURL	URL of the key encryption key that's used to encrypt the generated key. This parameter is optional if you select nokek in the UseExistingKek drop-down list. If you select kek in the UseExistingKek drop-down list, you must enter the <i>keyEncryptionKeyURL</i> value.
volumeType	Type of volume that the encryption operation is performed on. Valid supported values are <i>OS</i> or <i>All</i> (see supported Linux distros and their versions for OS and data disks in prerequisite section earlier).
sequenceVersion	Sequence version of the BitLocker operation. Increment this version number every time a disk-encryption operation is performed on the same VM.
vmName	Name of the VM that the encryption operation is to be performed on.
passphrase	Type a strong passphrase as the data encryption key.

Use EncryptFormatAll feature for data disks on Linux IaaS VMs

The **EncryptFormatAll** parameter reduces the time for Linux data disks to be encrypted. Partitions meeting certain criteria will be formatted (with its current file system). Then they'll be remounted back to where it was before command execution. If you wish to exclude a data disk that meets the criteria, you can unmount it before running the command.

After running this command, any drives that were mounted previously will be reformatted. Then the encryption layer will be started on top of the now empty drive. When this option is selected, the ephemeral resource disk attached to the VM will also be encrypted. If the ephemeral drive is reset, it will be reformatted and re-encrypted for the VM by the Azure Disk Encryption solution at the next opportunity.

WARNING

EncryptFormatAll shouldn't be used when there is needed data on a VM's data volumes. You may exclude disks from encryption by unmounting them. You should first try out the EncryptFormatAll first on a test VM, understand the feature parameter and its implication before trying it on the production VM. The EncryptFormatAll option formats the data disk and all the data on it will be lost. Before proceeding, verify that disks you wish to exclude are properly unmounted.

If you're setting this parameter while updating encryption settings, it might lead to a reboot before the actual encryption. In this case, you will also want to remove the disk you don't want formatted from the fstab file. Similarly, you should add the partition you want encrypt-formatted to the fstab file before initiating the encryption operation.

EncryptFormatAll criteria

The parameter goes through all partitions and encrypts them as long as they meet **all** of the criteria below:

- Is not a root/OS/boot partition
- Is not already encrypted

- Is not a BEK volume
- Is not a RAID volume
- Is not a LVM volume
- Is mounted

Encrypt the disks that compose the RAID or LVM volume rather than the RAID or LVM volume.

Use the EncryptFormatAll parameter with a template

To use the EncryptFormatAll option, use any pre-existing Azure Resource Manager template that encrypts a Linux VM and change the **EncryptionOperation** field for the AzureDiskEncryption resource.

1. As an example, use the [Resource Manager template to encrypt a running Linux IaaS VM](#).
2. Click **Deploy to Azure** on the Azure quickstart template.
3. Change the **EncryptionOperation** from **EnableEncryption** to **EnableEncryptionFormatAll**
4. Select the subscription, resource group, resource group location, other parameters, legal terms, and agreement. Click **Create** to enable encryption on the existing or running IaaS VM.

Use the EncryptFormatAll parameter with a PowerShell cmdlet

Use the [Set-AzureRmVMDiskEncryptionExtension](#) cmdlet with the **EncryptFormatAll** parameter.

Encrypt a running VM using a client secret and EncryptFormatAll: As an example, the script below initializes your variables and runs the Set-AzureRmVMDiskEncryptionExtension cmdlet with the EncryptFormatAll parameter. The resource group, VM, key vault, AAD app, and client secret should have already been created as prerequisites. Replace MySecureRg, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values.

```
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID $aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId $keyVaultResourceId -EncryptFormatAll
```

Use the EncryptFormatAll parameter with Logical Volume Manager (LVM)

We recommend an LVM-on-crypt setup. For all the following examples, replace the device-path and mountpoints with whatever suits your use-case. This setup can be done as follows:

- Add the data disks that will compose the VM.
- Format, mount, and add these disks to the fstab file.

1. Format the newly added disk. We use symlinks generated by Azure here. Using symlinks avoids problems related to device names changing. For more information, see the [Troubleshoot Device Names problems](#) article.

```
mkfs -t ext4 /dev/disk/azure/scsi1/lun0
```

2. Mount the disks.

```
mount /dev/disk/azure/scsi1/lun0 /mnt/mountpoint
```

3. Add to fstab.

```
echo "/dev/disk/azure/scsi1/lun0 /mnt/mountpoint ext4 defaults,nofail 1 2" >> /etc/fstab
```

4. Run the Set-AzureRmVMDiskEncryptionExtension PowerShell cmdlet with -EncryptFormatAll to encrypt these disks.

```
Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName "MySecureGroup" -VMName "MySecureVM" -DiskEncryptionKeyVaultUrl "https://mykeyvault.vault.azure.net/" -EncryptFormatAll
```

5. Set up LVM on top of these new disks. Note the encrypted drives are unlocked after the VM has finished booting. So, the LVM mounting will also have to be subsequently delayed.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using the Resource Manager template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail the Resource Manager template and CLI commands.

Use the instructions in the appendix for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- [Prepare a pre-encrypted Windows VHD](#)
- [Prepare a pre-encrypted Linux VHD](#)

IMPORTANT

It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and prior to enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or [Azure Backup](#) can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzureRmVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. The Set-AzureRmVMDiskEncryptionExtension command will fail against managed disk based VMs until a backup has been made and this parameter has been specified.

Encrypting or disabling encryption may cause the VM to reboot.

Use Azure PowerShell to encrypt IaaS VMs with pre-encrypted VHDS

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet [Set-AzureRmVMDiskEncryptionExtension](#). The example below gives you some common parameters.

```
$VirtualMachine = New-AzureRmVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzureRmVMDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri "os.vhd" Caching
ReadWrite -Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa" -DiskEncryptionKeyVaultId
"/subscriptions/00000000-0000-0000-0000-
00000000/resourceGroups/myresourcegroup/providers/Microsoft.KeyVault/vaults/mytestvault"
New-AzureRmVM -VM $VirtualMachine -ResourceGroupName "MySecureRG"
```

Use the Resource Manager template to encrypt IaaS VMs with pre-encrypted VHDS

You can enable disk encryption on your encrypted VHD by using the [Resource Manager template](#).

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Create** to enable encryption on the new IaaS VM.

The following table lists the Resource Manager template parameters for your encrypted VHD:

PARAMETER	DESCRIPTION
newStorageAccountName	Name of the storage account to store the encrypted OS VHD. This storage account should already have been created in the same resource group and same location as the VM.
osVhdUri	URI of the OS VHD from the storage account.
osType	OS product type (Windows/Linux).
virtualNetworkName	Name of the VNet that the VM NIC should belong to. The name should already have been created in the same resource group and same location as the VM.
subnetName	Name of the subnet on the VNet that the VM NIC should belong to.
vmSize	Size of the VM. Currently, only Standard A, D, and G series are supported.
keyVaultResourceId	The ResourceID that identifies the key vault resource in Azure Resource Manager. You can get it by using the PowerShell cmdlet <div style="border: 1px solid #ccc; padding: 5px;"> <pre>(Get-AzureRmKeyVault -VaultName &lt;MyKeyVaultName&ampgt -ResourceGroupName &lt;MyResourceGroupName&ampgt).ResourceId</pre> </div> or using the Azure CLI command <div style="border: 1px solid #ccc; padding: 5px;"> <pre>az keyvault show --name "MySecureVault" --query id</pre> </div>
keyVaultSecretUrl	URL of the disk-encryption key that's set up in the key vault.
keyVaultKekUrl	URL of the key encryption key for encrypting the generated disk-encryption key.
vmName	Name of the IaaS VM.

Enable encryption on a newly added data disk

You can add a new data disk using [az vm disk attach](#), or [through the Azure portal](#). Before you can encrypt, you need to mount the newly attached data disk first. You must request encryption of the data drive since the drive will be unusable while encryption is in progress.

Enable encryption on a newly added disk with Azure CLI

If the VM was previously encrypted with "All" then the --volume-type parameter should remain All. All includes both OS and data disks. If the VM was previously encrypted with a volume type of "OS", then the --volume-type parameter should be changed to All so that both the OS and the new data disk will be included. If the VM was encrypted with only the volume type of "Data", then it can remain "Data" as demonstrated below. Adding and attaching a new data disk to a VM is not sufficient preparation for encryption. The newly attached disk must also be formatted and properly mounted within the VM prior to enabling encryption. On Linux the disk must be mounted in /etc/fstab with a [persistent block device name](#).

In contrast to Powershell syntax, the CLI does not require the user to provide a unique sequence version when enabling encryption. The CLI automatically generates and uses its own unique sequence version value.

- **Encrypt a running VM using a client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --volume-type "Data"
```

- **Encrypt a running VM using KEK to wrap the client secret:**

```
az vm encryption enable --resource-group "MySecureRg" --name "MySecureVM" --aad-client-id "<my spn created with CLI which is the Azure AD ClientID>" --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type "Data"
```

Enable encryption on a newly added disk with Azure PowerShell

When using Powershell to encrypt a new disk for Linux, a new sequence version needs to be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version.

- **Encrypt a running VM using a client secret:** The script below initializes your variables and runs the Set-AzureRmVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, AAD app, and client secret should have already been created as prerequisites. Replace MySecureRg, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values. The -VolumeType parameter is set to data disks and not the OS disk. If the VM was previously encrypted with a volume type of "OS" or "All", then the -VolumeType parameter should be changed to All so that both the OS and the new data disk will be included.

```
$sequenceVersion = [Guid]::NewGuid();
$rgName = 'MySecureRg';
$vmName = 'MySecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID $aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl - DiskEncryptionKeyId $keyVaultResourceId -VolumeType 'data' -SequenceVersion $sequenceVersion;
```

- **Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault. The -VolumeType parameter is set to data disks and not the OS disk. If the VM was previously encrypted with a volume type of "OS" or "All", then the -VolumeType parameter should be changed to All so that both the OS and the new data disk will be included.

```

$rgName = 'MySecureRg';
$vmName = 'MyExtraSecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$keyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$keyVault = Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $rgname;
$diskEncryptionKeyVaultUrl = $keyVault.VaultUri;
$keyVaultResourceId = $keyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzureKeyVaultKey -VaultName $keyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $rgname -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $keyVaultResourceId -VolumeType 'data';

```

NOTE

The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

Disable encryption for Linux VMs

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template.

IMPORTANT

Disabling encryption with Azure Disk Encryption on Linux VMs is only supported for data volumes. It is not supported on data or OS volumes if the OS volume has been encrypted.

- Disable disk encryption with Azure PowerShell:** To disable the encryption, use the [Disable-AzureRmVMDiskEncryption](#) cmdlet.

```
Disable-AzureRmVMDiskEncryption -ResourceGroupName 'MySecureRG' -VMName 'MySecureVM' [--volume-type {ALL, DATA, OS}]
```

- Disable encryption with the Azure CLI:** To disable encryption, use the [az vm encryption disable](#) command.

```
az vm encryption disable --name "MySecureVM" --resource-group "MySecureRg" --volume-type [ALL, DATA, OS]
```

- Disable encryption with a Resource Manager Template:** Use the [Disable encryption on a running Linux VM](#) template to disable encryption.
 - Click **Deploy to Azure**.
 - Select the subscription, resource group, location, VM, legal terms, and agreement.
 - Click **Purchase** to disable disk encryption on a running Windows VM.

Next steps

Enable Azure Disk Encryption for Windows

Azure Storage security overview

11/7/2018 • 5 minutes to read • [Edit Online](#)

Azure Storage is the cloud storage solution for modern applications that rely on durability, availability, and scalability to meet the needs of their customers. Azure Storage provides a comprehensive set of security capabilities. You can:

- Secure the storage account by using Role-Based Access Control (RBAC) and Azure Active Directory.
- Secure data in transit between an application and Azure by using client-side encryption, HTTPS, or SMB 3.0.
- Set data to be automatically encrypted when it's written to Azure Storage by using Storage Service Encryption.
- Set OS and data disks used by virtual machines (VMs) to be encrypted by using Azure Disk Encryption.
- Grant delegated access to the data objects in Azure Storage by using shared access signatures (SASs).
- Use analytics to track the authentication method that someone is using when they access Storage.

For a more detailed look at security in Azure Storage, see the [Azure Storage security guide](#). This guide provides a deep dive into the security features of Azure Storage. These features include storage account keys, data encryption in transit and at rest, and storage analytics.

This article provides an overview of Azure security features that you can use with Azure Storage. Links to articles give details of each feature so you can learn more.

Role-Based Access Control

You can help secure your storage account by using Role-Based Access Control. Restricting access based on the [need to know](#) and [least privilege](#) security principles is imperative for organizations that want to enforce security policies for data access. These access rights are granted by assigning the appropriate RBAC role to groups and applications at a certain scope. You can use [built-in RBAC roles](#), such as Storage Account Contributor, to assign privileges to users.

Learn more:

- [Azure Active Directory Role-Based Access Control](#)

Delegated access to storage objects

A shared access signature provides delegated access to resources in your storage account. The SAS means that you can grant a client limited permissions to objects in your storage account for a specified period and with a specified set of permissions. You can grant these limited permissions without having to share your account access keys.

The SAS is a URI that encompasses in its query parameters all the information necessary for authenticated access to a storage resource. To access storage resources with the SAS, the client only needs to provide the SAS to the appropriate constructor or method.

Learn more:

- [Understanding the SAS model](#)
- [Create and use an SAS with Blob storage](#)

Encryption in transit

Encryption in transit is a mechanism of protecting data when it's transmitted across networks. With Azure Storage,

you can secure data by using:

- [Transport-level encryption](#), such as HTTPS, when you transfer data into or out of Azure Storage.
- [Wire encryption](#), such as SMB 3.0 encryption, for Azure file shares.
- [Client-side encryption](#), to encrypt the data before it's transferred into Storage and to decrypt the data after it is transferred out of Storage.

Learn more about client-side encryption:

- [Client-Side Encryption for Microsoft Azure Storage](#)
- [Cloud security controls series: Encrypting Data in Transit](#)

Encryption at rest

For many organizations, [data encryption at rest](#) is a mandatory step toward data privacy, compliance, and data sovereignty. Three Azure features provide encryption of data that's at rest:

- [Storage Service Encryption](#) enables you to request that the storage service automatically encrypt data when writing it to Azure Storage.
- [Client-side encryption](#) also provides the feature of encryption at rest.
- [Azure Disk Encryption](#) enables you to encrypt the OS disks and data disks that an IaaS virtual machine uses.

Learn more about Storage Service Encryption:

- [Azure Storage Service Encryption](#) is available for [Azure Blob storage](#). For details on other Azure storage types, see [Azure Files](#), [Disk \(Premium Storage\)](#), [Table storage](#), and [Queue storage](#).
- [Azure Storage Service Encryption for Data at Rest](#)

Azure Disk Encryption

Azure Disk Encryption for virtual machines helps you address organizational security and compliance requirements. It encrypts your VM disks (including boot and data disks) by using keys and policies that you control in [Azure Key Vault](#).

Disk Encryption for VMs works for Linux and Windows operating systems. It also uses Key Vault to help you safeguard, manage, and audit use of your disk encryption keys. All the data in your VM disks is encrypted at rest by using industry-standard encryption technology in your Azure storage accounts. The Disk Encryption solution for Windows is based on [Microsoft BitLocker Drive Encryption](#), and the Linux solution is based on [dm-crypt](#).

Learn more

- [Azure Disk Encryption for Windows and Linux IaaS Virtual Machines](#)

Firewalls and Virtual networks

Azure storage allows you to enable firewall rules for your storage accounts. Once enabled they will block incoming requests for data, including requests from other Azure services. You can configure exceptions to allow traffic. Firewall rules may be enabled on existing storage accounts or during creation time.

You should use this functionality to secure your storage accounts to a specific set of allowed networks.

For more information on Azure storage firewalls and virtual networks review the article [Configure Azure Storage Firewalls and Virtual Networks](#)

Azure Data Box

Data Box, Data Box Disk, and Data Box Heavy devices help you transfer large amounts of data to Azure when the

network isn't an option. These offline data transfer devices are shipped between your organization and the Azure data center. They use AES encryption to help protect your data in transit, and they undergo a thorough post-upload sanitization process to delete your data from the device.

Data Box Edge and Data Box Gateway are online data transfer products that act as network storage gateways to manage data between your site and Azure. Data Box Edge, an on-premises network device, transfers data to and from Azure and uses artificial intelligence (AI)-enabled edge compute to process data. Data Box Gateway is a virtual appliance with storage gateway capabilities.

Learn more:

- [Azure Data Box](#)
- [Azure Data Box Edge](#)
- [Azure Data Box Gateway](#)

Advanced Threat Protection

Azure Storage provides Advanced Threat Protection for an additional layer of security intelligence that detects unusual and potentially harmful attempts to access or exploit your storage account. Advanced Threat Protection monitors Azure Storage diagnostic logs for suspicious read, write, or delete requests to Blob storage.

Advanced Threat Protection alerts can be viewed from [Azure Security Center](#). Azure Security Center provides details on any suspicious activity detected and recommends actions to investigate and remediate the potential threat.

Learn more:

- [Azure Storage Advanced Threat Protection Overview](#)

Azure Key Vault

Azure Disk Encryption uses [Azure Key Vault](#) to help you control and manage disk encryption keys and secrets in your key vault subscription. It also ensures that all data in the virtual machine disks are encrypted at rest in Azure Storage. You should use Key Vault to audit keys and policy usage.

Learn more

- [What is Azure Key Vault?](#)
- [Get started with Azure Key Vault](#)

Azure Storage security guide

11/7/2018 • 42 minutes to read • [Edit Online](#)

Azure Storage provides a comprehensive set of security capabilities that together enable developers to build secure applications:

- All data written to Azure Storage is automatically encrypted using [Storage Service Encryption \(SSE\)](#). For more information, see [Announcing Default Encryption for Azure Blobs, Files, Table and Queue Storage](#).
- Azure Active Directory (Azure AD) and Role-Based Access Control (RBAC) are supported for Azure Storage for both resource management operations and data operations, as follows:
 - You can assign RBAC roles scoped to the storage account to security principals and use Azure AD to authorize resource management operations such as key management.
 - Azure AD integration is supported in preview for data operations on the Blob and Queue services. You can assign RBAC roles scoped to a subscription, resource group, storage account, or an individual container or queue to a security principal or a managed identity for Azure resources. For more information, see [Authenticate access to Azure Storage using Azure Active Directory \(Preview\)](#).
- Data can be secured in transit between an application and Azure by using [Client-Side Encryption](#), HTTPS, or SMB 3.0.
- OS and data disks used by Azure virtual machines can be encrypted using [Azure Disk Encryption](#).
- Delegated access to the data objects in Azure Storage can be granted using [Shared Access Signatures](#).

This article provides an overview of each of these security features that can be used with Azure Storage. Links are provided to articles that will give details of each feature so you can easily do further investigation on each topic.

Here are the topics to be covered in this article:

- [Management Plane Security](#) – Securing your Storage Account

The management plane consists of the resources used to manage your storage account. This section covers the Azure Resource Manager deployment model and how to use Role-Based Access Control (RBAC) to control access to your storage accounts. It also addresses managing your storage account keys and how to regenerate them.

- [Data Plane Security](#) – Securing Access to Your Data

In this section, we'll look at allowing access to the actual data objects in your Storage account, such as blobs, files, queues, and tables, using Shared Access Signatures and Stored Access Policies. We will cover both service-level SAS and account-level SAS. We'll also see how to limit access to a specific IP address (or range of IP addresses), how to limit the protocol used to HTTPS, and how to revoke a Shared Access Signature without waiting for it to expire.

- [Encryption in Transit](#)

This section discusses how to secure data when you transfer it into or out of Azure Storage. We'll talk about the recommended use of HTTPS and the encryption used by SMB 3.0 for Azure file shares. We will also take a look at Client-side Encryption, which enables you to encrypt the data before it is transferred into Storage in a client application, and to decrypt the data after it is transferred out of Storage.

- [Encryption at Rest](#)

We will talk about Storage Service Encryption (SSE), which is now automatically enabled for new and existing storage accounts. We will also look at how you can use Azure Disk Encryption and explore the basic

differences and cases of Disk Encryption versus SSE versus Client-Side Encryption. We will briefly look at FIPS compliance for U.S. Government computers.

- Using [Storage Analytics](#) to audit access of Azure Storage

This section discusses how to find information in the storage analytics logs for a request. We'll take a look at real storage analytics log data and see how to discern whether a request is made with the Storage account key, with a Shared Access signature, or anonymously, and whether it succeeded or failed.

- [Enabling Browser-Based Clients using CORS](#)

This section talks about how to allow cross-origin resource sharing (CORS). We'll talk about cross-domain access, and how to handle it with the CORS capabilities built into Azure Storage.

Management Plane Security

The management plane consists of operations that affect the storage account itself. For example, you can create or delete a storage account, get a list of storage accounts in a subscription, retrieve the storage account keys, or regenerate the storage account keys.

When you create a new storage account, you select a deployment model of Classic or Resource Manager. The Classic model of creating resources in Azure only allows all-or-nothing access to the subscription, and in turn, the storage account.

This guide focuses on the Resource Manager model that is the recommended means for creating storage accounts. With the Resource Manager storage accounts, rather than giving access to the entire subscription, you can control access on a more finite level to the management plane using Role-Based Access Control (RBAC).

How to secure your storage account with Role-Based Access Control (RBAC)

Let's talk about what RBAC is, and how you can use it. Each Azure subscription has an Azure Active Directory. Users, groups, and applications from that directory can be granted access to manage resources in the Azure subscription that use the Resource Manager deployment model. This type of security is referred to as Role-Based Access Control (RBAC). To manage this access, you can use the [Azure portal](#), the [Azure CLI tools](#), [PowerShell](#), or the [Azure Storage Resource Provider REST APIs](#).

With the Resource Manager model, you put the storage account in a resource group and control access to the management plane of that specific storage account using Azure Active Directory. For example, you can give specific users the ability to access the storage account keys, while other users can view information about the storage account, but cannot access the storage account keys.

Granting Access

Access is granted by assigning the appropriate RBAC role to users, groups, and applications, at the right scope. To grant access to the entire subscription, you assign a role at the subscription level. You can grant access to all of the resources in a resource group by granting permissions to the resource group itself. You can also assign specific roles to specific resources, such as storage accounts.

Here are the main points that you need to know about using RBAC to access the management operations of an Azure Storage account:

- When you assign access, you basically assign a role to the account that you want to have access. You can control access to the operations used to manage that storage account, but not to the data objects in the account. For example, you can grant permission to retrieve the properties of the storage account (such as redundancy), but not to a container or data within a container inside Blob Storage.
- For someone to have permission to access the data objects in the storage account, you can give them permission to read the storage account keys, and that user can then use those keys to access the blobs, queues, tables, and files.
- Roles can be assigned to a specific user account, a group of users, or to a specific application.

- Each role has a list of Actions and Not Actions. For example, the Virtual Machine Contributor role has an Action of "listKeys" that allows the storage account keys to be read. The Contributor has "Not Actions" such as updating the access for users in the Active Directory.
- Roles for storage include (but are not limited to) the following roles:
 - Owner – They can manage everything, including access.
 - Contributor – They can do anything the owner can do except assign access. Someone with this role can view and regenerate the storage account keys. With the storage account keys, they can access the data objects.
 - Reader – They can view information about the storage account, except secrets. For example, if you assign a role with reader permissions on the storage account to someone, they can view the properties of the storage account, but they can't make any changes to the properties or view the storage account keys.
 - Storage Account Contributor – They can manage the storage account – they can read the subscription's resource groups and resources, and create and manage subscription resource group deployments. They can also access the storage account keys, which in turn means they can access the data plane.
 - User Access Administrator – They can manage user access to the storage account. For example, they can grant Reader access to a specific user.
 - Virtual Machine Contributor – They can manage virtual machines but not the storage account to which they are connected. This role can list the storage account keys, which means that the user to whom you assign this role can update the data plane.

In order for a user to create a virtual machine, they have to be able to create the corresponding VHD file in a storage account. To do that, they need to be able to retrieve the storage account key and pass it to the API creating the VM. Therefore, they must have this permission so they can list the storage account keys.

- The ability to define custom roles is a feature that allows you to compose a set of actions from a list of available actions that can be performed on Azure resources.
- The user must be set up in your Azure Active Directory before you can assign a role to them.
- You can create a report of who granted/revoked what kind of access to/from whom and on what scope using PowerShell or the Azure CLI.

Resources

- [Azure Active Directory Role-based Access Control](#)

This article explains the Azure Active Directory Role-based Access Control and how it works.

- [RBAC: Built in Roles](#)

This article details all of the built-in roles available in RBAC.

- [Understanding Resource Manager deployment and classic deployment](#)

This article explains the Resource Manager deployment and classic deployment models, and explains the benefits of using the Resource Manager and resource groups. It explains how the Azure Compute, Network, and Storage Providers work under the Resource Manager model.

- [Managing Role-Based Access Control with the REST API](#)

This article shows how to use the REST API to manage RBAC.

- [Azure Storage Resource Provider REST API Reference](#)

This API reference describes the APIs you can use to manage your storage account programmatically.

- [Use Resource Manager authentication API to access subscriptions](#)

This article shows how to authenticate using the Resource Manager APIs.

- [Role-Based Access Control for Microsoft Azure from Ignite](#)

This is a link to a video on Channel 9 from the 2015 MS Ignite conference. In this session, they talk about access management and reporting capabilities in Azure, and explore best practices around securing access to Azure subscriptions using Azure Active Directory.

Managing Your Storage Account Keys

Storage account keys are 512-bit strings created by Azure that, along with the storage account name, can be used to access the data objects stored in the storage account, for example, blobs, entities within a table, queue messages, and files on an Azure file share. Controlling access to the storage account keys controls access to the data plane for that storage account.

Each storage account has two keys referred to as "Key 1" and "Key 2" in the [Azure portal](#) and in the PowerShell cmdlets. These can be regenerated manually using one of several methods, including, but not limited to using the [Azure portal](#), PowerShell, the Azure CLI, or programmatically using the .NET Storage Client Library or the Azure Storage Services REST API.

There are any number of reasons to regenerate your storage account keys.

- You might regenerate them on a regular basis for security reasons.
- You would regenerate your storage account keys if someone managed to hack into an application and retrieve the key that was hardcoded or saved in a configuration file, giving them full access to your storage account.
- Another case for key regeneration is if your team is using a Storage Explorer application that retains the storage account key, and one of the team members leaves. The application would continue to work, giving them access to your storage account after they're gone. This is actually the primary reason they created account-level Shared Access Signatures – you can use an account-level SAS instead of storing the access keys in a configuration file.

Key regeneration plan

You don't want to just regenerate the key you are using without some planning. If you do that, you could cut off all access to that storage account, which can cause major disruption. This is why there are two keys. You should regenerate one key at a time.

Before you regenerate your keys, be sure you have a list of all of your applications that are dependent on the storage account, as well as any other services you are using in Azure. For example, if you are using Azure Media Services that are dependent on your storage account, you must resync the access keys with your media service after you regenerate the key. If you are using any applications such as a storage explorer, you will need to provide the new keys to those applications as well. If you have VMs whose VHD files are stored in the storage account, they will not be affected by regenerating the storage account keys.

You can regenerate your keys in the Azure portal. Once keys are regenerated, they can take up to 10 minutes to be synchronized across Storage Services.

When you're ready, here's the general process detailing how you should change your key. In this case, the assumption is that you are currently using Key 1 and you are going to change everything to use Key 2 instead.

1. Regenerate Key 2 to ensure that it is secure. You can do this in the Azure portal.
2. In all of the applications where the storage key is stored, change the storage key to use Key 2's new value. Test and publish the application.
3. After all of the applications and services are up and running successfully, regenerate Key 1. This ensures that anybody to whom you have not expressly given the new key will no longer have access to the storage account.

If you are currently using Key 2, you can use the same process, but reverse the key names.

You can migrate over a couple of days, changing each application to use the new key and publishing it. After all of them are done, you should then go back and regenerate the old key so it no longer works.

Another option is to put the storage account key in an [Azure Key Vault](#) as a secret and have your applications retrieve the key from there. Then when you regenerate the key and update the Azure Key Vault, the applications will not need to be redeployed because they will pick up the new key from the Azure Key Vault automatically. Note that you can have the application read the key each time you need it, or you can cache it in memory and if it fails when using it, retrieve the key again from the Azure Key Vault.

Using Azure Key Vault also adds another level of security for your storage keys. If you use this method, you will never have the storage key hardcoded in a configuration file, which removes that avenue of somebody getting access to the keys without specific permission.

Another advantage of using Azure Key Vault is you can also control access to your keys using Azure Active Directory. This means you can grant access to the handful of applications that need to retrieve the keys from Azure Key Vault, and know that other applications will not be able to access the keys without granting them permission specifically.

NOTE

Microsoft recommends using only one of the keys in all of your applications at the same time. If you use Key 1 in some places and Key 2 in others, you will not be able to rotate your keys without some application losing access.

Resources

- [Manage storage account settings in the Azure portal](#)
- [Azure Storage Resource Provider REST API Reference](#)

Data Plane Security

Data Plane Security refers to the methods used to secure the data objects stored in Azure Storage – the blobs, queues, tables, and files. We've seen methods to encrypt the data and security during transit of the data, but how do you go about controlling access to the objects?

You have three options for authorizing access to data objects in Azure Storage, including:

- Using Azure AD to authorize access to containers and queues (Preview). Azure AD provides advantages over other approaches to authorization, including removing the need to store secrets in your code. For more information, see [Authenticate access to Azure Storage using Azure Active Directory \(Preview\)](#).
- Using your storage account keys to authorize access via Shared Key. Authorizing via Shared Key requires storing your storage account keys in your application, so Microsoft recommends using Azure AD instead where possible. For production applications, or for authorizing access to Azure tables and files, continue using Shared Key while Azure AD integration is in preview.
- Using Shared Access Signatures to grant controlled permissions to specific data objects for a specific amount of time.

In addition, for Blob Storage, you can allow public access to your blobs by setting the access level for the container that holds the blobs accordingly. If you set access for a container to Blob or Container, it will allow public read access for the blobs in that container. This means anyone with a URL pointing to a blob in that container can open it in a browser without using a Shared Access Signature or having the storage account keys.

In addition to limiting access through authorization, you can also use [Firewalls and Virtual Networks](#) to limit access to the storage account based on network rules. This approach enables you deny access to public internet traffic, and to grant access only to specific Azure Virtual Networks or public internet IP address ranges.

Storage Account Keys

Storage account keys are 512-bit strings created by Azure that, along with the storage account name, can be used to access the data objects stored in the storage account.

For example, you can read blobs, write to queues, create tables, and modify files. Many of these actions can be performed through the Azure portal, or using one of many Storage Explorer applications. You can also write code to use the REST API or one of the Storage Client Libraries to perform these operations.

As discussed in the section on the [Management Plane Security](#), access to the storage keys for a Classic storage account can be granted by giving full access to the Azure subscription. Access to the storage keys for a storage account using the Azure Resource Manager model can be controlled through Role-Based Access Control (RBAC).

How to delegate access to objects in your account using Shared Access Signatures and Stored Access Policies

A Shared Access Signature is a string containing a security token that can be attached to a URI that allows you to delegate access to storage objects and specify constraints such as the permissions and the date/time range of access.

You can grant access to blobs, containers, queue messages, files, and tables. With tables, you can actually grant permission to access a range of entities in the table by specifying the partition and row key ranges to which you want the user to have access. For example, if you have data stored with a partition key of geographical state, you could give someone access to just the data for California.

In another example, you might give a web application a SAS token that enables it to write entries to a queue, and give a worker role application a SAS token to get messages from the queue and process them. Or you could give one customer a SAS token they can use to upload pictures to a container in Blob Storage, and give a web application permission to read those pictures. In both cases, there is a separation of concerns – each application can be given just the access that they require in order to perform their task. This is possible through the use of Shared Access Signatures.

Why you want to use Shared Access Signatures

Why would you want to use an SAS instead of just giving out your storage account key, which is so much easier? Giving out your storage account key is like sharing the keys of your storage kingdom. It grants complete access. Someone could use your keys and upload their entire music library to your storage account. They could also replace your files with virus-infected versions, or steal your data. Giving away unlimited access to your storage account is something that should not be taken lightly.

With Shared Access Signatures, you can give a client just the permissions required for a limited amount of time. For example, if someone is uploading a blob to your account, you can grant them write access for just enough time to upload the blob (depending on the size of the blob, of course). And if you change your mind, you can revoke that access.

Additionally, you can specify that requests made using a SAS are restricted to a certain IP address or IP address range external to Azure. You can also require that requests are made using a specific protocol (HTTPS or HTTP/HTTPS). This means if you only want to allow HTTPS traffic, you can set the required protocol to HTTPS only, and HTTP traffic will be blocked.

Definition of a Shared Access Signature

A Shared Access Signature is a set of query parameters appended to the URL pointing at the resource that provides information about the access allowed and the length of time for which the access is permitted. Here is an example; this URI provides read access to a blob for five minutes. Note that SAS query parameters must be URL Encoded, such as %3A for colon (:) or %20 for a space.

```
http://mystorage.blob.core.windows.net/mycontainer/myblob.txt (URL to the blob)
?sv=2015-04-05 (storage service version)
&st=2015-12-10T22%3A18%3A26Z (start time, in UTC time and URL encoded)
&se=2015-12-10T22%3A23%3A26Z (end time, in UTC time and URL encoded)
&sr=b (resource is a blob)
&sp=r (read access)
&sip=168.1.5.60-168.1.5.70 (requests can only come from this range of IP addresses)
&spr=https (only allow HTTPS requests)
&sig=Z%2FRHIX5Xcg0Mq2rqI301WTjEg2tYkboXr1P9ZUXDtkk%3D (signature used for the authentication of the SAS)
```

How the Shared Access Signature is authorized by the Azure Storage Service

When the storage service receives the request, it takes the input query parameters and creates a signature using the same method as the calling program. It then compares the two signatures. If they agree, then the storage service can check the storage service version to make sure it's valid, verify that the current date and time are within the specified window, make sure the access requested corresponds to the request made, etc.

For example, with our URL above, if the URL was pointing to a file instead of a blob, this request would fail because it specifies that the Shared Access Signature is for a blob. If the REST command being called was to update a blob, it would fail because the Shared Access Signature specifies that only read access is permitted.

Types of Shared Access Signatures

- A service-level SAS can be used to access specific resources in a storage account. Some examples of this are retrieving a list of blobs in a container, downloading a blob, updating an entity in a table, adding messages to a queue, or uploading a file to a file share.
- An account-level SAS can be used to access anything that a service-level SAS can be used for. Additionally, it can give options to resources that are not permitted with a service-level SAS, such as the ability to create containers, tables, queues, and file shares. You can also specify access to multiple services at once. For example, you might give someone access to both blobs and files in your storage account.

Creating a SAS URI

1. You can create a URI on demand, defining all of the query parameters each time.

This approach is flexible, but if you have a logical set of parameters that are similar each time, using a Stored Access Policy is a better idea.

2. You can create a Stored Access Policy for an entire container, file share, table, or queue. Then you can use this as the basis for the SAS URIs you create. Permissions based on Stored Access Policies can be easily revoked. You can have up to five policies defined on each container, queue, table, or file share.

For example, if you were going to have many people read the blobs in a specific container, you could create a Stored Access Policy that says "give read access" and any other settings that will be the same each time. Then you can create an SAS URI using the settings of the Stored Access Policy and specifying the expiration date/time. The advantage of this is that you don't have to specify all of the query parameters every time.

Revocation

Suppose your SAS has been compromised, or you want to change it because of corporate security or regulatory compliance requirements. How do you revoke access to a resource using that SAS? It depends on how you created the SAS URI.

If you are using ad hoc URIs, you have three options. You can issue SAS tokens with short expiration policies and wait for the SAS to expire. You can rename or delete the resource (assuming the token was scoped to a single object). You can change the storage account keys. This last option can have a significant impact, depending on how many services are using that storage account, and probably isn't something you want to do without some planning.

If you are using a SAS derived from a Stored Access Policy, you can remove access by revoking the Stored Access Policy – you can just change it so it has already expired, or you can remove it altogether. This takes effect

immediately, and invalidates every SAS created using that Stored Access Policy. Updating or removing the Stored Access Policy may impact people accessing that specific container, file share, table, or queue via SAS, but if the clients are written so they request a new SAS when the old one becomes invalid, this will work fine.

Because using a SAS derived from a Stored Access Policy gives you the ability to revoke that SAS immediately, it is the recommended best practice to always use Stored Access Policies when possible.

Resources

For more detailed information on using Shared Access Signatures and Stored Access Policies, complete with examples, refer to the following articles:

- These are the reference articles.

- [Service SAS](#)

This article provides examples of using a service-level SAS with blobs, queue messages, table ranges, and files.

- [Constructing a service SAS](#)
 - [Constructing an account SAS](#)

- These are tutorials for using the .NET client library to create Shared Access Signatures and Stored Access Policies.

- [Using Shared Access Signatures \(SAS\)](#)
 - [Shared Access Signatures, Part 2: Create and Use a SAS with the Blob Service](#)

This article includes an explanation of the SAS model, examples of Shared Access Signatures, and recommendations for the best practice use of SAS. Also discussed is the revocation of the permission granted.

- Authentication
 - [Authentication for the Azure Storage Services](#)
- Shared Access Signatures Getting Started Tutorial
 - [SAS Getting Started Tutorial](#)

Encryption in Transit

Transport-Level Encryption – Using HTTPS

Another step you should take to ensure the security of your Azure Storage data is to encrypt the data between the client and Azure Storage. The first recommendation is to always use the [HTTPS](#) protocol, which ensures secure communication over the public Internet.

To have a secure communication channel, you should always use HTTPS when calling the REST APIs or accessing objects in storage. Also, **Shared Access Signatures**, which can be used to delegate access to Azure Storage objects, include an option to specify that only the HTTPS protocol can be used when using Shared Access Signatures, ensuring that anybody sending out links with SAS tokens will use the proper protocol.

You can enforce the use of HTTPS when calling the REST APIs to access objects in storage accounts by enabling [Secure transfer required](#) for the storage account. Connections using HTTP will be refused once this is enabled.

Using encryption during transit with Azure file shares

[Azure Files](#) supports encryption via SMB 3.0 and with HTTPS when using the File REST API. When mounting outside of the Azure region the Azure file share is located in, such as on-premises or in another Azure region, SMB 3.0 with encryption is always required. SMB 2.1 does not support encryption, so by default connections are only allowed within the same region in Azure, but SMB 3.0 with encryption can be enforced by [requiring secure](#)

[transfer](#) for the storage account.

SMB 3.0 with encryption is available in [all supported Windows and Windows Server operating systems](#) except Windows 7 and Windows Server 2008 R2, which only support SMB 2.1. SMB 3.0 is also supported on [macOS](#) and on distributions of [Linux](#) using Linux kernel 4.11 and above. Encryption support for SMB 3.0 has also been backported to older versions of the Linux kernel by several Linux distributions, consult [Understanding SMB client requirements](#).

Using Client-side encryption to secure data that you send to storage

Another option that helps you ensure that your data is secure while being transferred between a client application and Storage is Client-side Encryption. The data is encrypted before being transferred into Azure Storage. When retrieving the data from Azure Storage, the data is decrypted after it is received on the client side. Even though the data is encrypted going across the wire, we recommend that you also use HTTPS, as it has data integrity checks built in which help mitigate network errors affecting the integrity of the data.

Client-side encryption is also a method for encrypting your data at rest, as the data is stored in its encrypted form. We'll talk about this in more detail in the section on [Encryption at Rest](#).

Encryption at Rest

There are three Azure features that provide encryption at rest. Azure Disk Encryption is used to encrypt the OS and data disks in IaaS Virtual Machines. Client-side Encryption and SSE are both used to encrypt data in Azure Storage.

While you can use Client-side Encryption to encrypt the data in transit (which is also stored in its encrypted form in Storage), you may prefer to use HTTPS during the transfer, and have some way for the data to be automatically encrypted when it is stored. There are two ways to do this -- Azure Disk Encryption and SSE. One is used to directly encrypt the data on OS and data disks used by VMs, and the other is used to encrypt data written to Azure Blob Storage.

Storage Service Encryption (SSE)

SSE is enabled for all storage accounts and cannot be disabled. SSE automatically encrypts your data when writing it to Azure Storage. When you read data from Azure Storage, it is decrypted by Azure Storage before being returned. SSE enables you to secure your data without having to modify code or add code to any applications.

You can use either Microsoft-managed keys or your own custom keys. Microsoft generates managed keys and handles their secure storage as well as their regular rotation, as defined by internal Microsoft policy. For more information about using custom keys, see [Storage Service Encryption using customer-managed keys in Azure Key Vault](#).

SSE automatically encrypts data in all performance tiers (Standard and Premium), all deployment models (Azure Resource Manager and Classic), and all of the Azure Storage services (Blob, Queue, Table, and File).

Client-side Encryption

We mentioned client-side encryption when discussing the encryption of the data in transit. This feature allows you to programmatically encrypt your data in a client application before sending it across the wire to be written to Azure Storage, and to programmatically decrypt your data after retrieving it from Azure Storage.

This does provide encryption in transit, but it also provides the feature of Encryption at Rest. Although the data is encrypted in transit, we still recommend using HTTPS to take advantage of the built-in data integrity checks that help mitigate network errors affecting the integrity of the data.

An example of where you might use this is if you have a web application that stores blobs and retrieves blobs, and you want the application and data to be as secure as possible. In that case, you would use client-side encryption. The traffic between the client and the Azure Blob Service contains the encrypted resource, and nobody can interpret the data in transit and reconstitute it into your private blobs.

Client-side encryption is built into the Java and the .NET storage client libraries, which in turn use the Azure Key Vault APIs, making it easy for you to implement. The process of encrypting and decrypting the data uses the envelope technique, and stores metadata used by the encryption in each storage object. For example, for blobs, it stores it in the blob metadata, while for queues, it adds it to each queue message.

For the encryption itself, you can generate and manage your own encryption keys. You can also use keys generated by the Azure Storage Client Library, or you can have the Azure Key Vault generate the keys. You can store your encryption keys in your on-premises key storage, or you can store them in an Azure Key Vault. Azure Key Vault allows you to grant access to the secrets in Azure Key Vault to specific users using Azure Active Directory. This means that not just anybody can read the Azure Key Vault and retrieve the keys you're using for client-side encryption.

Resources

- [Encrypt and decrypt blobs in Microsoft Azure Storage using Azure Key Vault](#)

This article shows how to use client-side encryption with Azure Key Vault, including how to create the KEK and store it in the vault using PowerShell.

- [Client-Side Encryption and Azure Key Vault for Microsoft Azure Storage](#)

This article gives an explanation of client-side encryption, and provides examples of using the storage client library to encrypt and decrypt resources from the four storage services. It also talks about Azure Key Vault.

Using Azure Disk Encryption to encrypt disks used by your virtual machines

Azure Disk Encryption is a new feature. This feature allows you to encrypt the OS disks and Data disks used by an IaaS Virtual Machine. For Windows, the drives are encrypted using industry-standard BitLocker encryption technology. For Linux, the disks are encrypted using the DM-Crypt technology. This is integrated with Azure Key Vault to allow you to control and manage the disk encryption keys.

The solution supports the following scenarios for IaaS VMs when they are enabled in Microsoft Azure:

- Integration with Azure Key Vault
- Standard tier VMs: [A, D, DS, G, GS, and so forth series IaaS VMs](#)
- Enabling encryption on Windows and Linux IaaS VMs
- Disabling encryption on OS and data drives for Windows IaaS VMs
- Disabling encryption on data drives for Linux IaaS VMs
- Enabling encryption on IaaS VMs that are running Windows client OS
- Enabling encryption on volumes with mount paths
- Enabling encryption on Linux VMs that are configured with disk striping (RAID) by using mdadm
- Enabling encryption on Linux VMs by using LVM for data disks
- Enabling encryption on Windows VMs that are configured by using storage spaces
- All Azure public regions are supported

The solution does not support the following scenarios, features, and technology in the release:

- Basic tier IaaS VMs
- Disabling encryption on an OS drive for Linux IaaS VMs
- IaaS VMs that are created by using the classic VM creation method
- Integration with your on-premises Key Management Service
- Azure Files (shared file system), Network File System (NFS), dynamic volumes, and Windows VMs that are configured with software-based RAID systems

NOTE

Linux OS disk encryption is currently supported on the following Linux distributions: RHEL 7.2, CentOS 7.2n, and Ubuntu 16.04.

This feature ensures that all data on your virtual machine disks is encrypted at rest in Azure Storage.

Resources

- [Azure Disk Encryption for Windows and Linux IaaS VMs](#)

Comparison of Azure Disk Encryption, SSE, and Client-Side Encryption

IaaS VMs and their VHD files

For data disks used by IaaS VMs, Azure Disk Encryption is recommended. If you create a VM with unmanaged disks using an image from the Azure Marketplace, Azure performs a [shallow copy](#) of the image to your storage account in Azure Storage, and it is not encrypted even if you have SSE enabled. After it creates the VM and starts updating the image, SSE will start encrypting the data. For this reason, it's best to use Azure Disk Encryption on VMs with unmanaged disks created from images in the Azure Marketplace if you want them fully encrypted. If you create a VM with Managed Disks, SSE encrypts all the data by default using platform managed keys.

If you bring a pre-encrypted VM into Azure from on-premises, you will be able to upload the encryption keys to Azure Key Vault, and continue using the encryption for that VM that you were using on-premises. Azure Disk Encryption is enabled to handle this scenario.

If you have non-encrypted VHD from on-premises, you can upload it into the gallery as a custom image and provision a VM from it. If you do this using the Resource Manager templates, you can ask it to turn on Azure Disk Encryption when it boots up the VM.

When you add a data disk and mount it on the VM, you can turn on Azure Disk Encryption on that data disk. It will encrypt that data disk locally first, and then the classic deployment model layer will do a lazy write against storage so the storage content is encrypted.

Client-side encryption

Client-side encryption is the most secure method of encrypting your data, because it encrypts data prior to transit. However, it does require that you add code to your applications using storage, which you may not want to do. In those cases, you can use HTTPS to secure your data in transit. Once data reaches Azure Storage, it is encrypted by SSE.

With client-side encryption, you can encrypt table entities, queue messages, and blobs.

Client-side encryption is managed entirely by the application. This is the most secure approach, but does require you to make programmatic changes to your application and put key management processes in place. You would use this when you want the extra security during transit, and you want your stored data to be encrypted.

Client-side encryption is more load on the client, and you have to account for this in your scalability plans, especially if you are encrypting and transferring a large amount of data.

Storage Service Encryption (SSE)

SSE is managed by Azure Storage. SSE does not provide for the security of the data in transit, but it does encrypt the data as it is written to Azure Storage. SSE does not affect Azure Storage performance.

You can encrypt any kind of data of the storage account using SSE (block blobs, append blobs, page blobs, table data, queue data, and files).

If you have an archive or library of VHD files that you use as a basis for creating new virtual machines, you can create a new storage account and then upload the VHD files to that account. Those VHD files will be encrypted by Azure Storage.

If you have Azure Disk Encryption enabled for the disks in a VM, then any newly written data is encrypted both by SSE and by Azure Disk Encryption.

Storage Analytics

Using Storage Analytics to monitor authorization type

For each storage account, you can enable Azure Storage Analytics to perform logging and store metrics data. This is a great tool to use when you want to check the performance metrics of a storage account, or need to troubleshoot a storage account because you are having performance problems.

Another piece of data you can see in the storage analytics logs is the authentication method used by someone when they access storage. For example, with Blob Storage, you can see if they used a Shared Access Signature or the storage account keys, or if the blob accessed was public.

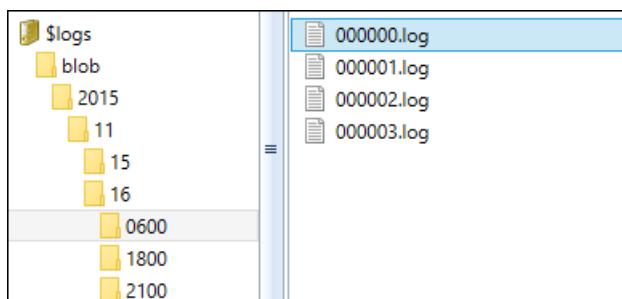
This can be helpful if you are tightly guarding access to storage. For example, in Blob Storage you can set all of the containers to private and implement the use of an SAS service throughout your applications. Then you can check the logs regularly to see if your blobs are accessed using the storage account keys, which may indicate a breach of security, or if the blobs are public but they shouldn't be.

What do the logs look like?

After you enable the storage account metrics and logging through the Azure portal, analytics data will start to accumulate quickly. The logging and metrics for each service is separate; the logging is only written when there is activity in that storage account, while the metrics will be logged every minute, every hour, or every day, depending on how you configure it.

The logs are stored in block blobs in a container named \$logs in the storage account. This container is automatically created when Storage Analytics is enabled. Once this container is created, you can't delete it, although you can delete its contents.

Under the \$logs container, there is a folder for each service, and then there are subfolders for the year/month/day/hour. Under hour, the logs are numbered. This is what the directory structure will look like:



Every request to Azure Storage is logged. Here's a snapshot of a log file, showing the first few fields.

```
1.0;2015-11-16T06:13:26.9046078Z;GetBlobServiceProperties;Success;200;3;3;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:13:27.2588724Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:14:28.0166751Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:14:29.2558837Z;GetBlobServiceProperties;Success;200;3;3;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:14:43.4307865Z;BlobPreflightRequest;AnonymousSuccess;200;2;2;anonymous;;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:14:43.4528051Z;GetBlobServiceProperties;Success;200;1;1;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:15:30.3567270Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:15:29.2735098Z;GetBlobServiceProperties;Success;200;5;5;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:16:32.9445742Z;GetBlobServiceProperties;Success;200;4;3;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:16:44.2766486Z;ListContainers;Success;200;4;4;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:16:56.0216743Z;CreateContainer;Success;201;10;10;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:16:56.0517020Z;ListContainers;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:16:59.9423538Z;ListContainers;Success;200;3;3;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:16:59.9984102Z;ListBlobs;Success;200;3;3;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:17:23.7717291Z;GetBlobProperties;ClientOtherError;404;3;3;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:17:23.8347867Z;PutBlob;Success;201;71;8;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:17:23.9549008Z;GetBlobProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:17:31.9243814Z;GetBlobProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:17:31.9554107Z;GetBlob;Success;206;81;5;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:17:46.1437305Z;GetContainerACL;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";1.0;2015-11-16T06:16:30.3890982Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorage.blob.core.windows.net/";"https://mystorage.blob.core.windows.net/";
```

You can see that you can use the logs to track any kind of calls to a storage account.

What are all of those fields for?

There is an article listed in the resources below that provides the list of the many fields in the logs and what they are used for. Here is the list of fields in order:

```
<version-number>;<request-start-time>;<operation-type>;<request-status>;<http-status-code>;<end-to-end-latency-in-ms>;<server-latency-in-ms>;<authentication-type>;<requester-account-name>;<owner-account-name>;<service-type>;<request-url>;<requested-object-key>;<request-id-header>;<operation-count>;<requester-ip-address>;<request-version-header>;<request-header-size>;<request-packet-size>;<response-header-size>;<response-packet-size>;<request-content-length>;<request-md5>;<server-md5>;<etag-identifier>;<last-modified-time>;<conditions-used>;<user-agent-header>;<referrer-header>;<client-request-id>
```

We're interested in the entries for GetBlob, and how they are authorized, so we need to look for entries with operation-type "Get-Blob", and check the request-status (fourth column) and the authorization-type (eighth column).

For example, in the first few rows in the listing above, the request-status is "Success" and the authorization-type is "authenticated". This means the request was authorized using the storage account key.

How is access to my blobs being authorized?

We have three cases that we are interested in.

1. The blob is public and it is accessed using a URL without a Shared Access Signature. In this case, the request-status is "AnonymousSuccess" and the authorization-type is "anonymous".

```
1.0;2015-11-17T02:01:29.0488963Z;GetBlob;AnonymousSuccess;200;124;37;anonymous;;mystorage...
```

2. The blob is private and was used with a Shared Access Signature. In this case, the request-status is "SASuccess" and the authorization-type is "sas".

```
1.0;2015-11-16T18:30:05.6556115Z;GetBlob;SASuccess;200;416;64;sas;;mystorage...
```

3. The blob is private and the storage key was used to access it. In this case, the request-status is "**Success**" and the authorization-type is "**authenticated**".

```
1.0;2015-11-16T18:32:24.3174537Z;GetBlob;Success;206;59;22;authenticated;;mystorage...
```

You can use the Microsoft Message Analyzer to view and analyze these logs. It includes search and filter capabilities. For example, you might want to search for instances of GetBlob to see if the usage is what you expect, that is, to make sure someone is not accessing your storage account inappropriately.

Resources

- [Storage Analytics](#)

This article is an overview of storage analytics and how to enable them.

- [Storage Analytics Log Format](#)

This article illustrates the Storage Analytics Log Format, and details the fields available therein, including authentication-type, which indicates the type of authentication used for the request.

- [Monitor a Storage Account in the Azure portal](#)

This article shows how to configure monitoring of metrics and logging for a storage account.

- [End-to-End Troubleshooting using Azure Storage Metrics and Logging, AzCopy, and Message Analyzer](#)

This article talks about troubleshooting using the Storage Analytics and shows how to use the Microsoft Message Analyzer.

- [Microsoft Message Analyzer Operating Guide](#)

This article is the reference for the Microsoft Message Analyzer and includes links to a tutorial, quickstart, and feature summary.

Cross-Origin Resource Sharing (CORS)

Cross-domain access of resources

When a web browser running in one domain makes an HTTP request for a resource from a different domain, this is called a cross-origin HTTP request. For example, an HTML page served from contoso.com makes a request for a jpeg hosted on fabrikam.blob.core.windows.net. For security reasons, browsers restrict cross-origin HTTP requests initiated from within scripts, such as JavaScript. This means that when some JavaScript code on a web page on contoso.com requests that jpeg on fabrikam.blob.core.windows.net, the browser will not allow the request.

What does this have to do with Azure Storage? Well, if you are storing static assets such as JSON or XML data files in Blob Storage using a storage account called Fabrikam, the domain for the assets will be fabrikam.blob.core.windows.net, and the contoso.com web application will not be able to access them using JavaScript because the domains are different. This is also true if you're trying to call one of the Azure Storage Services – such as Table Storage – that return JSON data to be processed by the JavaScript client.

Possible solutions

One way to resolve this is to assign a custom domain like "storage.contoso.com" to fabrikam.blob.core.windows.net. The problem is that you can only assign that custom domain to one storage account. What if the assets are stored in multiple storage accounts?

Another way to resolve this is to have the web application act as a proxy for the storage calls. This means if you are uploading a file to Blob Storage, the web application would either write it locally and then copy it to Blob Storage, or it would read all of it into memory and then write it to Blob Storage. Alternately, you could write a dedicated web application (such as a Web API) that uploads the files locally and writes them to Blob Storage. Either way, you have to account for that function when determining the scalability needs.

How can CORS help?

Azure Storage allows you to enable CORS – Cross Origin Resource Sharing. For each storage account, you can specify domains that can access the resources in that storage account. For example, in our case outlined above, we can enable CORS on the fabrikam.blob.core.windows.net storage account and configure it to allow access to contoso.com. Then the web application contoso.com can directly access the resources in fabrikam.blob.core.windows.net.

One thing to note is that CORS allows access, but it does not provide authentication, which is required for all non-public access of storage resources. This means you can only access blobs if they are public or you include a Shared Access Signature giving you the appropriate permission. Tables, queues, and files have no public access, and require a SAS.

By default, CORS is disabled on all services. You can enable CORS by using the REST API or the storage client library to call one of the methods to set the service policies. When you do that, you include a CORS rule, which is in XML. Here's an example of a CORS rule that has been set using the Set Service Properties operation for the Blob Service for a storage account. You can perform that operation using the storage client library or the REST APIs for Azure Storage.

```
<Cors>
  <CorsRule>
    <AllowedOrigins>http://www.contoso.com, http://www.fabrikam.com</AllowedOrigins>
    <AllowedMethods>PUT,GET</AllowedMethods>
    <AllowedHeaders>x-ms-meta-data*,x-ms-meta-target*,x-ms-meta-abc</AllowedHeaders>
    <ExposedHeaders>x-ms-meta-*</ExposedHeaders>
    <MaxAgeInSeconds>200</MaxAgeInSeconds>
  </CorsRule>
</Cors>
```

Here's what each row means:

- **AllowedOrigins** This tells which non-matching domains can request and receive data from the storage service. This says that both contoso.com and fabrikam.com can request data from Blob Storage for a specific storage account. You can also set this to a wildcard (*) to allow all domains to access requests.
- **AllowedMethods** This is the list of methods (HTTP request verbs) that can be used when making the request. In this example, only PUT and GET are allowed. You can set this to a wildcard (*) to allow all methods to be used.
- **AllowedHeaders** This is the request headers that the origin domain can specify when making the request. In this example, all metadata headers starting with x-ms-meta-data, x-ms-meta-target, and x-ms-meta-abc are permitted. The wildcard character (*) indicates that any header beginning with the specified prefix is allowed.
- **ExposedHeaders** This tells which response headers should be exposed by the browser to the request issuer. In this example, any header starting with "x-ms-meta-" will be exposed.
- **MaxAgeInSeconds** This is the maximum amount of time that a browser will cache the preflight OPTIONS request. (For more information about the preflight request, check the first article below.)

Resources

For more information about CORS and how to enable it, check out these resources.

- [Cross-Origin Resource Sharing \(CORS\) Support for the Azure Storage Services on Azure.com](#)

This article provides an overview of CORS and how to set the rules for the different storage services.

- [Cross-Origin Resource Sharing \(CORS\) Support for the Azure Storage Services on MSDN](#)

This is the reference documentation for CORS support for the Azure Storage Services. This has links to articles applying to each storage service, and shows an example and explains each element in the CORS file.

- [Microsoft Azure Storage: Introducing CORS](#)

This is a link to the initial blog article announcing CORS and showing how to use it.

Frequently asked questions about Azure Storage security

1. How can I verify the integrity of the blobs I'm transferring into or out of Azure Storage if I can't use the HTTPS protocol?

If for any reason you need to use HTTP instead of HTTPS and you are working with block blobs, you can use MD5 checking to help verify the integrity of the blobs being transferred. This will help with protection from network/transport layer errors, but not necessarily with intermediary attacks.

If you can use HTTPS, which provides transport level security, then using MD5 checking is redundant and unnecessary.

For more information, please check out the [Azure Blob MD5 Overview](#).

2. What about FIPS-Compliance for the U.S. Government?

The United States Federal Information Processing Standard (FIPS) defines cryptographic algorithms approved for use by U.S. Federal government computer systems for the protection of sensitive data. Enabling FIPS mode on a Windows server or desktop tells the OS that only FIPS-validated cryptographic algorithms should be used. If an application uses non-compliant algorithms, the applications will break. With .NET Framework versions 4.5.2 or higher, the application automatically switches the cryptography algorithms to use FIPS-compliant algorithms when the computer is in FIPS mode.

Microsoft leaves it up to each customer to decide whether to enable FIPS mode. We believe there is no compelling reason for customers who are not subject to government regulations to enable FIPS mode by default.

Resources

- [Why We're Not Recommending "FIPS Mode" Anymore](#)

This blog article gives an overview of FIPS and explains why they don't enable FIPS mode by default.

- [FIPS 140 Validation](#)

This article provides information on how Microsoft products and cryptographic modules comply with the FIPS standard for the U.S. Federal government.

- ["System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing" security settings effects in Windows XP and in later versions of Windows](#)

This article talks about the use of FIPS mode in older Windows computers.

Azure infrastructure security

7/20/2018 • 2 minutes to read • [Edit Online](#)

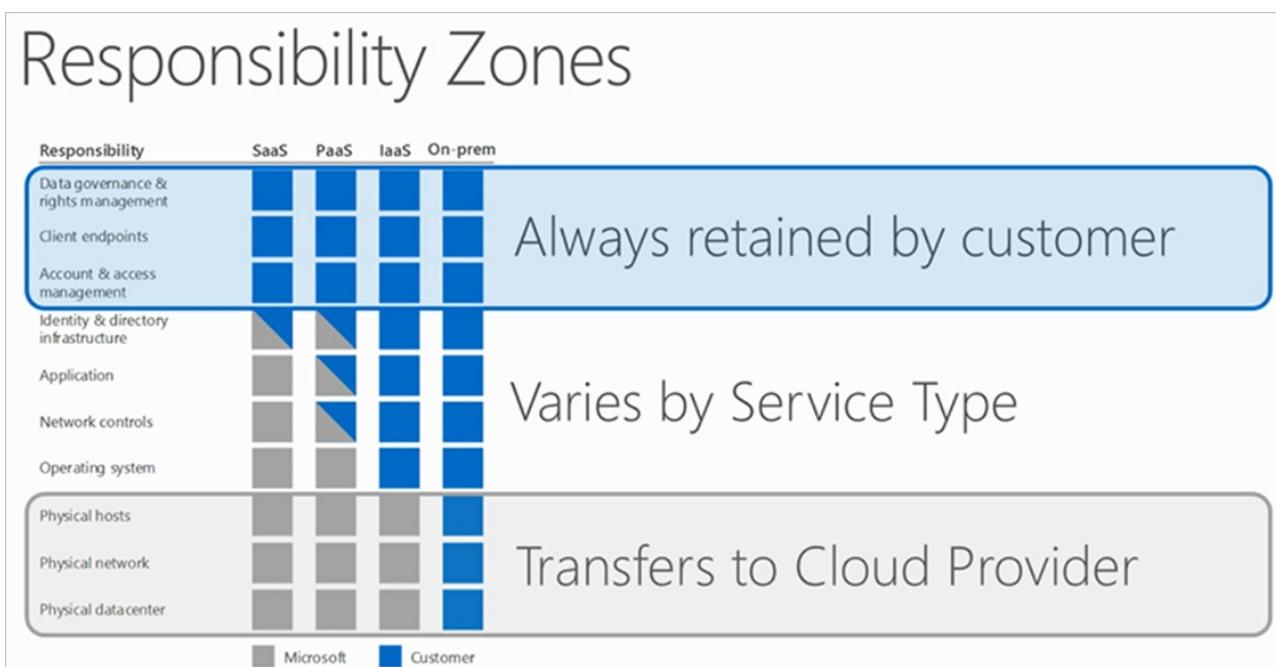
Microsoft Azure runs in datacenters managed and operated by Microsoft. These geographically dispersed datacenters comply with key industry standards, such as ISO/IEC 27001:2013 and NIST SP 800-53, for security and reliability. The datacenters are managed, monitored, and administered by Microsoft operations staff. The operations staff has years of experience in delivering the world's largest online services with 24 x 7 continuity.

This series of articles provides information about what Microsoft does to secure the Azure infrastructure. The articles address:

- [Physical security](#)
- [Availability](#)
- [Components and boundaries](#)
- [Network architecture](#)
- [Production network](#)
- [SQL Database](#)
- [Operations](#)
- [Monitoring](#)
- [Integrity](#)
- [Data protection](#)

Shared responsibility model

It's important to understand the division of responsibility between you and Microsoft. On-premises, you own the whole stack, but as you move to the cloud, some responsibilities transfer to Microsoft. The following graphic illustrates the areas of responsibility, according to the type of deployment of your stack (software as a service [SaaS], platform as a service [PaaS], infrastructure as a service [IaaS], and on-premises).



You are always responsible for the following, regardless of the type of deployment:

- Data

- Endpoints
- Account
- Access management

Be sure that you understand the division of responsibility between you and Microsoft in a SaaS, PaaS, and IaaS deployment. For more information, see [Shared responsibilities for cloud computing](#).

Next steps

To learn more about what Microsoft does to help secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure information system components and boundaries](#)
- [Azure network architecture](#)
- [Azure production network](#)
- [Azure SQL Database security features](#)
- [Azure production operations and management](#)
- [Azure infrastructure monitoring](#)
- [Azure infrastructure integrity](#)
- [Azure customer data protection](#)

Azure facilities, premises, and physical security

7/20/2018 • 5 minutes to read • [Edit Online](#)

Azure is composed of a [globally distributed datacenter infrastructure](#), supporting thousands of online services and spanning more than 100 highly secure facilities worldwide.

The infrastructure is designed to bring applications closer to users around the world, preserving data residency, and offering comprehensive compliance and resiliency options for customers. Azure has 52 regions worldwide, and is available in 140 countries.

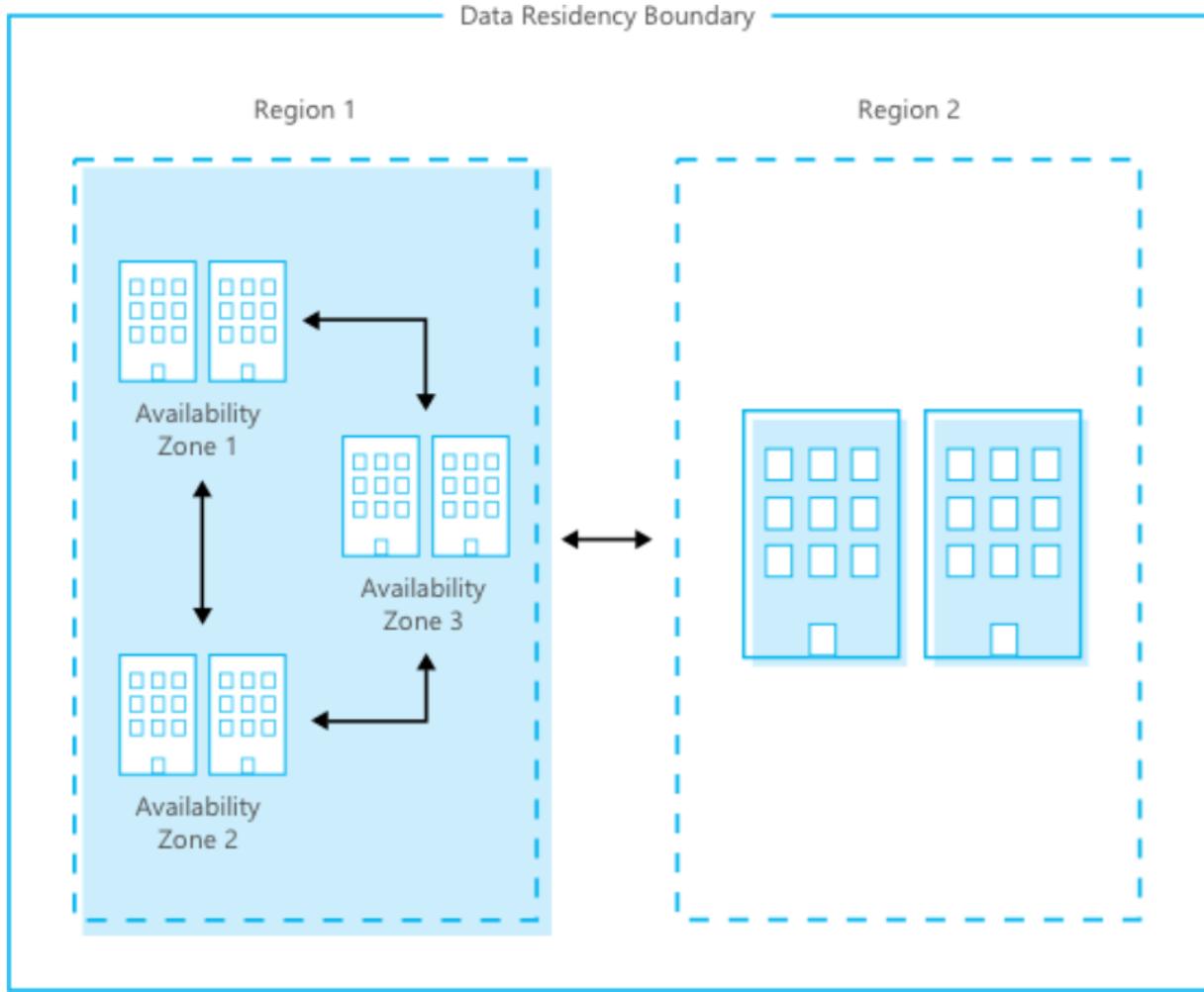
A region is a set of datacenters that is interconnected via a massive and resilient network. The network includes content distribution, load balancing, redundancy, and encryption by default. With more global regions than any other cloud provider, Azure gives you the flexibility to deploy applications where you need them.

Azure regions are organized into geographies. An Azure geography ensures that data residency, sovereignty, compliance, and resiliency requirements are honored within geographical boundaries.

Geographies allow customers with specific data-residency and compliance needs to keep their data and applications close. Geographies are fault-tolerant to withstand complete region failure, through their connection to the dedicated, high-capacity networking infrastructure.

Availability zones are physically separate locations within an Azure region. Each availability zone is made up of one or more datacenters equipped with independent power, cooling, and networking. Availability zones allow you to run mission-critical applications with high availability and low-latency replication.

The following figure shows how the Azure global infrastructure pairs region and availability zones within the same data residency boundary for high availability, disaster recovery, and backup.



Geographically distributed datacenters enables Microsoft to be close to customers, to reduce network latency and allow for geo-redundant backup and failover.

Physical security

Microsoft designs, builds, and operates datacenters in a way that strictly controls physical access to the areas where your data is stored. Microsoft understands the importance of protecting your data, and is committed to helping secure the datacenters that contain your data. We have an entire division at Microsoft devoted to designing, building, and operating the physical facilities supporting Azure. This team is invested in maintaining state-of-the-art physical security.

Microsoft takes a layered approach to physical security, to reduce the risk of unauthorized users gaining physical access to data and the datacenter resources. Datacenters managed by Microsoft have extensive layers of protection: access approval at the facility's perimeter, at the building's perimeter, inside the building, and on the datacenter floor. Layers of physical security are:

- **Access request and approval.** You must request access prior to arriving at the datacenter. You're required to provide a valid business justification for your visit, such as compliance or auditing purposes. All requests are approved on a need-to-access basis by Microsoft employees. A need-to-access basis helps keep the number of individuals needed to complete a task in the datacenters to the bare minimum. After Microsoft grants permission, an individual only has access to the discrete area of the datacenter required, based on the approved business justification. Permissions are limited to a certain period of time, and then expire.
- **Facility's perimeter.** When you arrive at a datacenter, you're required to go through a well-defined access point. Typically, tall fences made of steel and concrete encompass every inch of the perimeter. There are cameras around the datacenters, with a security team monitoring their videos at all times.

- **Building entrance.** The datacenter entrance is staffed with professional security officers who have undergone rigorous training and background checks. These security officers also routinely patrol the datacenter, and monitor the videos of cameras inside the datacenter at all times.
- **Inside the building.** After you enter the building, you must pass two-factor authentication with biometrics to continue moving through the datacenter. If your identity is validated, you can enter only the portion of the datacenter that you have approved access to. You can stay there only for the duration of the time approved.
- **Datacenter floor.** You are only allowed onto the floor that you're approved to enter. You are required to pass a full body metal detection screening. To reduce the risk of unauthorized data entering or leaving the datacenter without our knowledge, only approved devices can make their way into the datacenter floor. Additionally, video cameras monitor the front and back of every server rack. When you exit the datacenter floor, you again must pass through full body metal detection screening. To leave the datacenter, you're required to pass through an additional security scan.

Microsoft requires visitors to surrender badges upon departure from any Microsoft facility.

Physical security reviews

Periodically, we conduct physical security reviews of the facilities, to ensure the datacenters properly address Azure security requirements. The datacenter hosting provider personnel do not provide Azure service management. Personnel can't sign in to Azure systems and don't have physical access to the Azure collocation room and cages.

Data bearing devices

Microsoft uses best practice procedures and a wiping solution that is [NIST 800-88 compliant](#). For hard drives that can't be wiped, we use a destruction process that destroys it and renders the recovery of information impossible. This destruction process can be to disintegrate, shred, pulverize, or incinerate. We determine the means of disposal according to the asset type. We retain records of the destruction.

Equipment disposal

Upon a system's end-of-life, Microsoft operational personnel follow rigorous data handling and hardware disposal procedures to assure that hardware containing your data is not made available to untrusted parties. We use a secure erase approach for hard drives that support it. For hard drives that can't be wiped, we use a destruction process that destroys the drive and renders the recovery of information impossible. This destruction process can be to disintegrate, shred, pulverize, or incinerate. We determine the means of disposal according to the asset type. We retain records of the destruction. All Azure services use approved media storage and disposal management services.

Compliance

We design and manage the Azure infrastructure to meet a broad set of international and industry-specific compliance standards, such as ISO 27001, HIPAA, FedRAMP, SOC 1, and SOC 2. We also meet country-specific standards, including Australia IRAP, UK G-Cloud, and Singapore MTCS. Rigorous third-party audits, such as those done by the British Standards Institute, verify adherence to the strict security controls these standards mandate.

For a full list of compliance standards that Azure adheres to, see the [Compliance offerings](#).

Next steps

To learn more about what Microsoft does to help secure the Azure infrastructure, see:

- Azure infrastructure availability
- Azure information system components and boundaries
- Azure network architecture
- Azure production network
- Azure SQL Database security features
- Azure production operations and management
- Azure infrastructure monitoring
- Azure infrastructure integrity
- Azure customer data protection

Azure infrastructure availability

7/20/2018 • 2 minutes to read • [Edit Online](#)

Azure provides robust availability, based on extensive redundancy achieved with virtualization technology. Azure provides numerous levels of redundancy to provide maximum availability of customers' data.

Temporary outages and natural disaster

The Microsoft Cloud Infrastructure and Operations team designs, builds, operates, and improves the security of the cloud infrastructure. This team ensures that the Azure infrastructure is delivering high availability and reliability, high efficiency, and smart scalability. The team provides a more secure, private, and trusted cloud.

Uninterruptible power supplies and vast banks of batteries ensure that electricity remains continuous if a short-term power disruption occurs. Emergency generators provide backup power for extended outages and planned maintenance. If a natural disaster occurs, the datacenter can use onsite fuel reserves.

High-speed and robust fiber optic networks connect datacenters with other major hubs and internet users. Compute nodes host workloads closer to users to reduce latency, provide geo-redundancy, and increase overall service resiliency. A team of engineers works around the clock to ensure services are persistently available.

Microsoft ensures high availability through advanced monitoring and incident response, service support, and backup failover capability. Geographically distributed Microsoft operations centers operate 24/7/365. The Azure network is one of the largest in the world. The fiber optic and content distribution network connects datacenters and edge nodes to ensure high performance and reliability.

Disaster recovery

Azure keeps your data durable in two locations. You can choose the location of the backup site. In both locations, Azure constantly maintains three healthy replicas of your data.

Database availability

Azure ensures that a database is internet accessible through an internet gateway with sustained database availability. Monitoring assesses the health and state of the active databases at five-minute time intervals.

Storage availability

Azure delivers storage through a highly scalable and durable storage service, which provides connectivity endpoints. This means that an application can access the storage service directly. The storage service processes incoming storage requests efficiently, with transactional integrity.

Next steps

To learn more about what Microsoft does to help secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure information system components and boundaries](#)
- [Azure network architecture](#)
- [Azure production network](#)
- [Azure SQL Database security features](#)
- [Azure production operations and management](#)

- Azure infrastructure monitoring
- Azure infrastructure integrity
- Azure customer data protection

Azure information system components and boundaries

7/20/2018 • 8 minutes to read • [Edit Online](#)

This article provides a general description of the Azure architecture and management. The Azure system environment is made up of the following networks:

- Microsoft Azure production network (Azure network)
- Microsoft corporate network (corpnet)

Separate IT teams are responsible for operations and maintenance of these networks.

Azure architecture

Azure is a cloud computing platform and infrastructure for building, deploying, and managing applications and services through a network of datacenters. Microsoft manages these datacenters. Based on the number of resources you specify, Azure creates virtual machines (VMs) based on resource need. These VMs run on an Azure hypervisor, which is designed for use in the cloud and is not accessible to the public.

On each Azure physical server node, there is a hypervisor that runs directly over the hardware. The hypervisor divides a node into a variable number of guest VMs. Each node also has one root VM, which runs the host operating system. Windows Firewall is enabled on each VM. You define which ports are addressable by configuring the service definition file. These ports are the only ones open and addressable, internally or externally. All traffic and access to the disk and network is mediated by the hypervisor and root operating system.

At the host layer, Azure VMs run a customized and hardened version of the latest Windows Server. Azure uses a version of Windows Server that includes only those components necessary to host VMs. This improves performance and reduces attack surface. Machine boundaries are enforced by the hypervisor, which doesn't depend on the operating system security.

Azure management by fabric controllers

In Azure, VMs running on physical servers (blades/nodes) are grouped into clusters of about 1000. The VMs are independently managed by a scaled-out and redundant platform software component called the fabric controller (FC).

Each FC manages the lifecycle of applications running in its cluster, and provisions and monitors the health of the hardware under its control. It runs autonomic operations, such as reincarnating VM instances on healthy servers when it determines that a server has failed. The FC also performs application-management operations, such as deploying, updating, and scaling out applications.

The datacenter is divided into clusters. Clusters isolate faults at the FC level, and prevent certain classes of errors from affecting servers beyond the cluster in which they occur. FCs that serve a particular Azure cluster are grouped into an FC cluster.

Hardware inventory

The FC prepares an inventory of Azure hardware and network devices during the bootstrap configuration process. Any new hardware and network components entering the Azure production environment must follow the bootstrap configuration process. The FC is responsible for managing the entire inventory listed in the datacenter.xml configuration file.

FC-managed operating system images

The operating system team provides images, in the form of Virtual Hard Disks, deployed on all host and guest VMs in the Azure production environment. The team constructs these base images through an automated offline build process. The base image is a version of the operating system in which the kernel and other core components have been modified and optimized to support the Azure environment.

There are three types of fabric-managed operating system images:

- Host: A customized operating system that runs on host VMs.
- Native: A native operating system that runs on tenants (for example, Azure Storage). This operating system does not have any hypervisor.
- Guest: A guest operating system that runs on guest VMs.

The host and native FC-managed operating systems are designed for use in the cloud, and are not publicly accessible.

Host and native operating systems

Host and native are hardened operating system images that host the fabric agents, and run on a compute node (runs as first VM on the node) and storage nodes. The benefit of using optimized base images of host and native is that it reduces the surface area exposed by APIs or unused components. These can present high security risks and increase the footprint of the operating system. Reduced-footprint operating systems only include the components necessary to Azure.

Guest operating system

Azure internal components running on guest operating system VMs have no opportunity to run Remote Desktop Protocol. Any changes to baseline configuration settings must go through the change and release management process.

Azure datacenters

The Microsoft Cloud Infrastructure and Operations (MCIO) team manages the physical infrastructure and datacenter facilities for all Microsoft online services. MCIO is primarily responsible for managing the physical and environmental controls within the datacenters, as well as managing and supporting outer perimeter network devices (such as edge routers and datacenter routers). MCIO is also responsible for setting up the bare minimum server hardware on racks in the datacenter. Customers have no direct interaction with Azure.

Service management and service teams

Various engineering groups, known as service teams, manage the support of the Azure service. Each service team is responsible for an area of support for Azure. Each service team must make an engineer available 24x7 to investigate and resolve failures in the service. Service teams do not, by default, have physical access to the hardware operating in Azure.

The service teams are:

- Application Platform
- Azure Active Directory
- Azure Compute
- Azure Net
- Cloud Engineering Services
- ISSD: Security
- Multifactor Authentication
- SQL Database
- Storage

Types of users

Employees (or contractors) of Microsoft are considered to be internal users. All other users are considered to be external users. All Azure internal users have their employee status categorized with a sensitivity level that defines their access to customer data (access or no access). User privileges to Azure (authorization permission after authentication takes place) are described in the following table:

ROLE	INTERNAL OR EXTERNAL	SENSITIVITY LEVEL	AUTHORIZED PRIVILEGES AND FUNCTIONS PERFORMED	ACCESS TYPE
Azure datacenter engineer	Internal	No access to customer data	Manage the physical security of the premises. Conduct patrols in and out of the datacenter, and monitor all entry points. Escort into and out of the datacenter certain non-cleared personnel who provide general services (such as dining or cleaning) or IT work within the datacenter. Conduct routine monitoring and maintenance of network hardware. Perform incident management and break-fix work by using a variety of tools. Conduct routine monitoring and maintenance of the physical hardware in the datacenters. Access to environment on demand from property owners. Capable to perform forensic investigations, log incident reports, and require mandatory security training and policy requirements. Operational ownership and maintenance of critical security tools, such as scanners and log collection.	Persistent access to the environment.

Role	Internal or External	Sensitivity Level	Authorized Privileges and Functions Performed	Access Type
Azure incident triage (rapid response engineers)	Internal	Access to customer data	Manage communications among MCIO, support, and engineering teams. Triage platform incidents, deployment issues, and service requests.	Just-in-time access to the environment, with limited persistent access to non-customer systems.
Azure deployment engineers	Internal	Access to customer data	Deploy and upgrade platform components, software, and scheduled configuration changes in support of Azure.	Just-in-time access to the environment, with limited persistent access to non-customer systems.
Azure customer outage support (tenant)	Internal	Access to customer data	Debug and diagnose platform outages and faults for individual compute tenants and Azure accounts. Analyze faults. Drive critical fixes to the platform or customer, and drive technical improvements across support.	Just-in-time access to the environment, with limited persistent access to non-customer systems.
Azure live site engineers (monitoring engineers) and incident	Internal	Access to customer data	Diagnose and mitigate platform health by using diagnostic tools. Drive fixes for volume drivers, repair items resulting from outages, and assist outage restoration actions.	Just-in-time access to the environment, with limited persistent access to non-customer systems.
Azure customers	External	N/A	N/A	N/A

Azure uses unique identifiers to authenticate organizational users and customers (or processes acting on behalf of organizational users). This applies to all assets and devices that are part of the Azure environment.

Azure internal authentication

Communications between Azure internal components are protected with TLS encryption. In most cases, the X.509 certificates are self-signed. Certificates with connections that can be accessed from outside the Azure network are an exception, as are certificates for the FCs. FCs have certificates issued by a Microsoft Certificate of Authority (CA) that is backed by a trusted root CA. This allows FC public keys to be rolled over easily. Additionally, Microsoft developer tools use FC public keys. When developers submit new application images, the images are encrypted with an FC public key in order to protect any embedded secrets.

Azure hardware device authentication

The FC maintains a set of credentials (keys and/or passwords) used to authenticate itself to various hardware devices under its control. Microsoft uses a system to prevent access to these credentials. Specifically, the transport, persistence, and use of these credentials is designed to prevent Azure developers, administrators, and backup services and personnel access to sensitive, confidential, or private information.

Microsoft uses encryption based on the FC's master identity public key. This occurs at FC setup and FC reconfiguration times, to transfer the credentials used to access networking hardware devices. When the FC needs the credentials, the FC retrieves and decrypts them.

Network devices

The Azure networking team configures network service accounts to enable an Azure client to authenticate to network devices (routers, switches, and load balancers).

Secure service administration

Azure operations personnel are required to use secure admin workstations (SAWs). Customers can implement similar controls by using privileged access workstations. With SAWs, administrative personnel use an individually assigned administrative account that is separate from the user's standard user account. The SAW builds on that account separation practice by providing a trustworthy workstation for those sensitive accounts.

Next steps

To learn more about what Microsoft does to help secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure network architecture](#)
- [Azure production network](#)
- [Azure SQL Database security features](#)
- [Azure production operations and management](#)
- [Azure infrastructure monitoring](#)
- [Azure infrastructure integrity](#)
- [Azure customer data protection](#)

Azure network architecture

8/3/2018 • 6 minutes to read • [Edit Online](#)

The Azure network architecture follows a modified version of the industry standard core/distribution/access model, with distinct hardware layers. The layers include:

- Core (datacenter routers)
- Distribution (access routers and L2 aggregation). The distribution layer separates L3 routing from L2 switching.
- Access (L2 host switches)

The network architecture has two levels of layer 2 switches. One layer aggregates traffic from the other layer. The second layer loops to incorporate redundancy. This provides a more flexible VLAN footprint, and improves port scaling. The architecture keeps L2 and L3 distinct, which allows the use of hardware in each of the distinct layers in the network, and minimizes fault in one layer from affecting the other layer(s). The use of trunks allows for resource sharing, such as the connectivity to the L3 infrastructure.

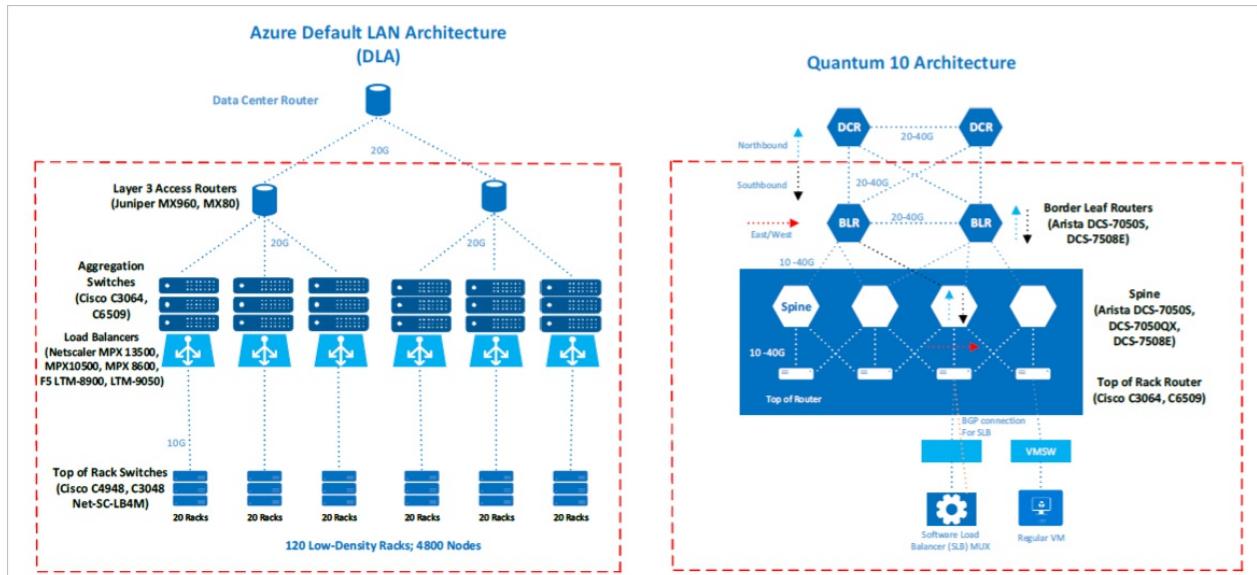
Network configuration

The network architecture of an Azure cluster within a datacenter consists of the following devices:

- Routers (datacenter, access router, and border leaf routers)
- Switches (aggregation and top-of-rack switches)
- Digi CMs
- Power distribution units

Azure has two separate architectures. Some existing Azure customers and shared services reside on the default LAN architecture (DLA), whereas new regions and virtual customers reside on Quantum 10 (Q10) architecture. The DLA architecture is a traditional tree design, with active/passive access routers and security access control lists (ACLs) applied to the access routers. The Quantum 10 architecture is a Close/mesh design of routers, where ACLs are not applied at the routers. Instead, ACLs are applied below the routing, through Software Load Balancing (SLB) or software defined VLANs.

The following diagram provides a high-level overview of the network architecture within an Azure cluster:



Quantum 10 devices

The Quantum 10 design conducts layer 3 switching spread over multiple devices in a Clos/mesh design. The advantages of the Q10 design include larger capability and greater ability to scale existing network infrastructure. The design employs border leaf routers, spine switches, and top-of-rack routers to pass traffic to clusters across multiple routes, allowing for fault tolerance. Software load balancing, instead of hardware devices, handles security services such as network address translation.

Access routers

The distribution/access L3 routers (ARs) perform the primary routing functionality for the distribution and access layers. These devices are deployed as a pair, and are the default gateway for subnets. Each AR pair can support multiple L2 aggregation switch pairs, depending on capacity. The maximum number depends on the capacity of the device, as well as failure domains. A typical number is three L2 aggregation switch pairs per AR pair.

L2 aggregation switches

These devices serve as an aggregation point for L2 traffic. They are the distribution layer for the L2 fabric, and can handle large amounts of traffic. Because these devices aggregate traffic, they require 802.1q functionality, and high bandwidth technologies such as port aggregation and 10GE.

L2 host switches

Hosts connect directly to these switches. They can be rack mounted switches, or chassis deployments. The 802.1q standard allows for the designation of one VLAN as a native VLAN, treating that VLAN as normal (untagged) Ethernet framing. Under normal circumstances, frames on the native VLAN are transmitted and received untagged on an 802.1q trunk port. This feature was designed for migration to 802.1q and compatibility with non-802.1q capable devices. In this architecture, only the network infrastructure uses the native VLAN.

This architecture specifies a standard for native VLAN selection. The standard ensures, where possible, that the AR devices have a unique, native VLAN for every trunk and the L2Aggregation to L2Aggregation trunks. The L2Aggregation to L2Host Switch trunks have a non-default native VLAN.

Link aggregation (802.3ad)

Link aggregation allows multiple individual links to be bundled together, and treated as a single logical link. To facilitate operational debugging, the number used to designate port-channel interfaces should be standardized. The rest of the network uses the same number at both ends of a port-channel.

The numbers specified for the L2Agg to L2Host switch are the port-channel numbers used on the L2Agg side. Because the range of numbers is more limited at the L2Host side, the standard is to use numbers 1 and 2 at the L2Host side. These refer to the port-channel going to the "a" side and the "b" side, respectively.

VLANS

The network architecture uses VLANs to group servers together into a single broadcast domain. VLAN numbers conform to 802.1q standards, which supports VLANs numbered 1–4094.

Customer VLANs

You have various VLAN implementation options you can deploy through the Azure portal to meet the separation and architecture needs of your solution. You deploy these solutions through virtual machines. For customer reference architecture examples, see [Azure reference architectures](#).

Edge architecture

Azure datacenters are built upon highly redundant and well-provisioned network infrastructures. Microsoft implements networks within the Azure datacenters with "need plus one" ($N+1$) redundancy architectures or better. Full failover features within and between datacenters help to ensure network and service availability. Externally, datacenters are served by dedicated, high-bandwidth network circuits. These circuits redundantly connect properties with over 1200 internet service providers globally at multiple peering points. This provides in

excess of 2,000 Gbps of potential edge capacity across the network.

Filtering routers at the edge and access layer of the Azure network provide well-established security at the packet level. This helps to prevent unauthorized attempts to connect to Azure. The routers help to ensure that the actual contents of the packets contain data in the expected format, and conform to the expected client/server communication scheme. Azure implements a tiered architecture, consisting of the following network segregation and access control components:

- **Edge routers.** These segregate the application environment from the internet. Edge routers are designed to provide anti-spoof protection and limit access by using ACLs.
- **Distribution (access) routers.** These allow only Microsoft approved IP addresses, provide anti-spoofing, and establish connections by using ACLs.

A10 DDOS mitigation architecture

Denial of service attacks continue to present a real threat to the reliability of online services. As attacks become more targeted and sophisticated, and as the services Microsoft provides become more geographically diverse, identifying and minimizing the impact of these attacks is a high priority. The following details explain how the A10 DDOS mitigation system is implemented from a network architecture perspective.

Azure uses A10 network devices at the datacenter router (DCR) that provide automated detection and mitigation. The A10 solution uses Azure Network Monitoring to sample flow packets and determine if there is an attack. If the attack is detected, A10 devices scrub to mitigate attacks. Only then is clean traffic is allowed into the Azure datacenter directly from the DCR. Microsoft uses the A10 solution to protect the Azure network infrastructure.

DDoS protections in the A10 solution include:

- UDP IPv4 and IPv6 flood protection
- ICMP IPv4 and IPv6 flood protection
- TCP IPv4 and IPv6 flood protection
- TCP SYN attack protection for IPv4 and IPv6
- Fragmentation attack

NOTE

Microsoft provides DDoS protection by default for all Azure customers.

Network connection rules

On its network, Azure deploys edge routers that provide security at the packet level to prevent unauthorized attempts to connect to Azure. Edge routers ensure that the actual contents of the packets contain data in the expected format, and conform to the expected client/server communication scheme.

Edge routers segregate the application environment from the internet. These routers are designed to provide anti-spoof protection, and limit access by using ACLs. Microsoft configures edge routers by using a tiered ACL approach, to limit network protocols that are allowed to transit the edge routers and access routers.

Microsoft positions network devices at access and edge locations, to act as boundary points where ingress or egress filters are applied.

Next steps

To learn more about what Microsoft does to help secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)

- Azure infrastructure availability
- Azure information system components and boundaries
- Azure production network
- Azure SQL Database security features
- Azure production operations and management
- Azure infrastructure monitoring
- Azure infrastructure integrity
- Azure customer data protection

The Azure production network

7/17/2018 • 7 minutes to read • [Edit Online](#)

The users of the Azure production network include both external customers who access their own Azure applications and internal Azure support personnel who manage the production network. This article discusses the security access methods and protection mechanisms for establishing connections to the Azure production network.

Internet routing and fault tolerance

A globally redundant internal and external Azure Domain Name Service (DNS) infrastructure, combined with multiple primary and secondary DNS server clusters, provides fault tolerance. At the same time, additional Azure network security controls, such as NetScaler, are used to prevent distributed denial of service (DDoS) attacks and protect the integrity of Azure DNS services.

The Azure DNS servers are located at multiple datacenter facilities. The Azure DNS implementation incorporates a hierarchy of secondary and primary DNS servers to publicly resolve Azure customer domain names. The domain names usually resolve to a CloudApp.net address, which wraps the virtual IP (VIP) address for the customer's service. Unique to Azure, the VIP that corresponds to internal dedicated IP (DIP) address of the tenant translation is done by the Microsoft load balancers responsible for that VIP.

Azure is hosted in geographically distributed Azure datacenters within the US, and it's built on state-of-the-art routing platforms that implement robust, scalable architectural standards. Among the notable features are:

- Multiprotocol Label Switching (MPLS)-based traffic engineering, which provides efficient link utilization and graceful degradation of service if there is an outage.
- Networks are implemented with "need plus one" ($N+1$) redundancy architectures or better.
- Externally, datacenters are served by dedicated, high-bandwidth network circuits that redundantly connect properties with over 1,200 internet service providers globally at multiple peering points. This connection provides in excess of 2,000 gigabytes per second (Gbps) of edge capacity.

Because Microsoft owns its own network circuits between datacenters, these attributes help the Azure offering achieve 99.9+ percent network availability without the need for traditional third-party internet service providers.

Connection to production network and associated firewalls

The Azure network internet traffic flow policy directs traffic to the Azure production network that's located in the nearest regional datacenter within the US. Because the Azure production datacenters maintain consistent network architecture and hardware, the traffic flow description that follows applies consistently to all datacenters.

After internet traffic for Azure is routed to the nearest datacenter, a connection is established to the access routers. These access routers serve to isolate traffic between Azure nodes and customer-instantiated VMs. Network infrastructure devices at the access and edge locations are the boundary points where ingress and egress filters are applied. These routers are configured through a tiered access-control list (ACL) to filter unwanted network traffic and apply traffic rate limits, if necessary. Traffic that is allowed by ACL is routed to the load balancers. Distribution routers are designed to allow only Microsoft-approved IP addresses, provide anti-spoofing, and establish TCP connections that use ACLs.

External load-balancing devices are located behind the access routers to perform network address translation (NAT) from internet-routable IPs to Azure internal IPs. The devices also route packets to valid production internal IPs and ports, and they act as a protection mechanism to limit exposing the internal production network

address space.

By default, Microsoft enforces Hypertext Transfer Protocol Secure (HTTPS) for all traffic that's transmitted to customers' web browsers, including sign-in and all traffic thereafter. The use of TLS v1.2 enables a secure tunnel for traffic to flow through. ACLs on access and core routers ensure that the source of the traffic is consistent with what is expected.

An important distinction in this architecture, when it's compared to traditional security architecture, is that there are no dedicated hardware firewalls, specialized intrusion detection or prevention devices, or other security appliances that are normally expected before connections are made to the Azure production environment. Customers usually expect these hardware firewall devices in the Azure network; however, none are employed within Azure. Almost exclusively, those security features are built into the software that runs the Azure environment to provide robust, multi-layered security mechanisms, including firewall capabilities. Additionally, the scope of the boundary and associated sprawl of critical security devices is easier to manage and inventory, as shown in the preceding illustration, because it is managed by the software that's running Azure.

Core security and firewall features

Azure implements robust software security and firewall features at various levels to enforce security features that are usually expected in a traditional environment to protect the core Security Authorization boundary.

Azure security features

Azure implements host-based software firewalls inside the production network. Several core security and firewall features reside within the core Azure environment. These security features reflect a defense-in-depth strategy within the Azure environment. Customer data in Azure is protected by the following firewalls:

Hypervisor firewall (packet filter): This firewall is implemented in the hypervisor and configured by the fabric controller (FC) agent. This firewall protects the tenant that runs inside the VM from unauthorized access. By default, when a VM is created, all traffic is blocked and then the FC agent adds rules and exceptions in the filter to allow authorized traffic.

Two categories of rules are programmed here:

- **Machine config or infrastructure rules:** By default, all communication is blocked. Exceptions exist that allow a VM to send and receive Dynamic Host Configuration Protocol (DHCP) communications and DNS information, and send traffic to the "public" internet outbound to other VMs within the FC cluster and OS Activation server. Because the VMs' allowed list of outgoing destinations does not include Azure router subnets and other Microsoft properties, the rules act as a layer of defense for them.
- **Role configuration file rules:** Defines the inbound ACLs based on the tenants' service model. For example, if a tenant has a web front end on port 80 on a certain VM, port 80 is opened to all IP addresses. If the VM has a worker role running, the worker role is opened only to the VM within the same tenant.

Native host firewall: Azure Service Fabric and Azure Storage run on a native OS, which has no hypervisor and, therefore, Windows Firewall is configured with the preceding two sets of rules.

Host firewall: The host firewall protects the host partition, which runs the hypervisor. The rules are programmed to allow only the FC and jump boxes to talk to the host partition on a specific port. The other exceptions are to allow DHCP response and DNS replies. Azure uses a machine configuration file, which contains a template of firewall rules for the host partition. A host firewall exception also exists that allows VMs to communicate to host components, wire server, and metadata server, through specific protocol/ports.

Guest firewall: The Windows Firewall piece of the guest OS, which is configurable by customers on customer VMs and storage.

Additional security features that are built into the Azure capabilities include:

- Infrastructure components that are assigned IP addresses that are from DIPs. An attacker on the internet

cannot address traffic to those addresses because it would not reach Microsoft. Internet gateway routers filter packets that are addressed solely to internal addresses, so they would not enter the production network. The only components that accept traffic that's directed to VIPs are load balancers.

- Firewalls that are implemented on all internal nodes have three primary security architecture considerations for any given scenario:
 - Firewalls are placed behind the load balancer and accept packets from anywhere. These packets are intended to be externally exposed and would correspond to the open ports in a traditional perimeter firewall.
 - Firewalls accept packets only from a limited set of addresses. This consideration is part of the defensive in-depth strategy against DDoS attacks. Such connections are cryptographically authenticated.
 - Firewalls can be accessed only from select internal nodes. They accept packets only from an enumerated list of source IP addresses, all of which are DIPs within the Azure network. For example, an attack on the corporate network could direct requests to these addresses, but the attacks would be blocked unless the source address of the packet was one in the enumerated list within the Azure network.
 - The access router at the perimeter blocks outbound packets that are addressed to an address that's inside the Azure network because of its configured static routes.

Next steps

To learn more about what Microsoft does to secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure information system components and boundaries](#)
- [Azure network architecture](#)
- [Azure SQL Database security features](#)
- [Azure production operations and management](#)
- [Azure infrastructure monitoring](#)
- [Azure infrastructure integrity](#)
- [Azure customer data protection](#)

Azure SQL Database security features

7/20/2018 • 6 minutes to read • [Edit Online](#)

Azure SQL Database provides a relational database service in Azure. To protect customer data and provide strong security features that customers expect from a relational database service, SQL Database has its own sets of security capabilities. These capabilities build upon the controls that are inherited from Azure.

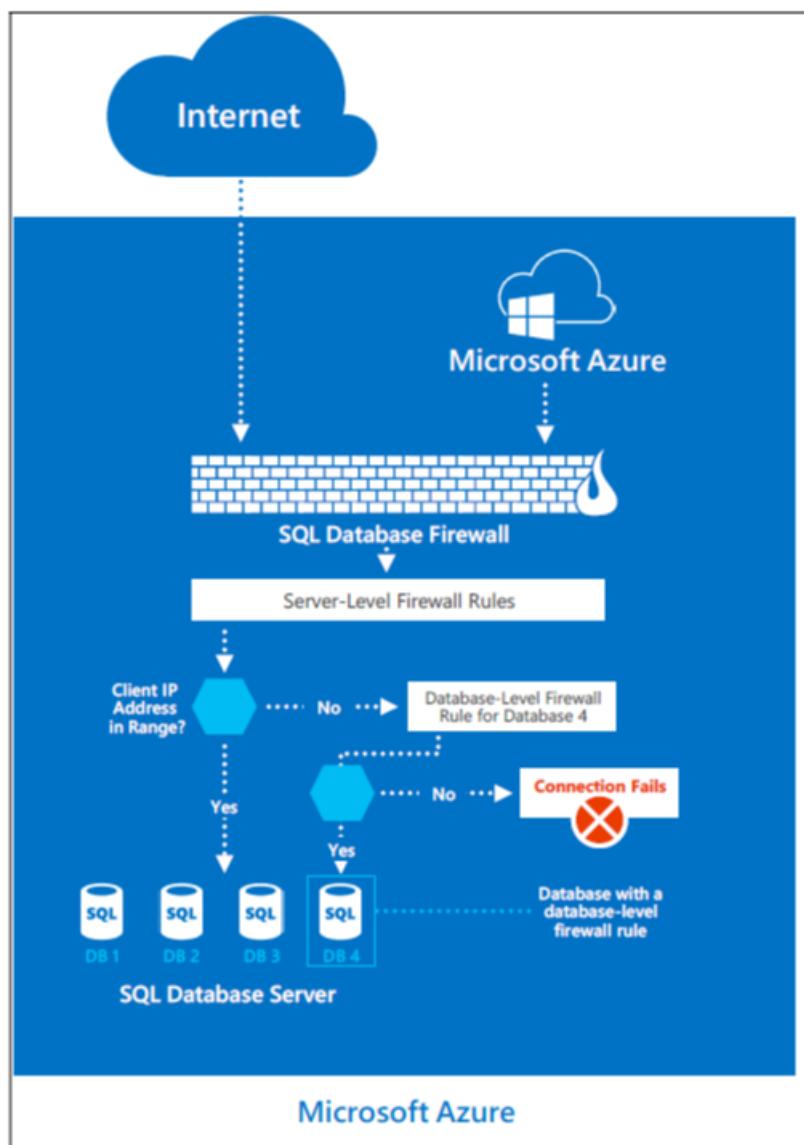
Security capabilities

Usage of the TDS protocol

Azure SQL Database supports only the tabular data stream (TDS) protocol, which requires the database to be accessible over only the default port of TCP/1433.

Azure SQL Database firewall

To help protect customer data, Azure SQL Database includes a firewall functionality, which by default prevents all access to the SQL Database server, as shown below.



The gateway firewall can limit addresses, which allows customers granular control to specify ranges of acceptable IP addresses. The firewall grants access based on the originating IP address of each request.

Customers can achieve firewall configuration by using a management portal or programmatically using the Azure SQL Database Management REST API. The Azure SQL Database gateway firewall by default prevents all customer TDS access to Azure SQL database instances. Customers must configure access by using access-control lists (ACLs) to permit Azure SQL Database connections by source and destination internet addresses, protocols, and port numbers.

DoSGuard

Denial of service (DoS) attacks are reduced by a SQL Database gateway service called DoSGuard. DoSGuard actively tracks failed logins from IP addresses. If there are multiple failed logins from a specific IP address within a period of time, the IP address is blocked from accessing any resources in the service for a pre-defined time period.

In addition, the Azure SQL Database gateway performs:

- Secure channel capability negotiations to implement TDS FIPS 140-2 validated encrypted connections when it connects to the database servers.
- Stateful TDS packet inspection while it accepts connections from clients. The gateway validates the connection information and passes on the TDS packets to the appropriate physical server based on the database name that's specified in the connection string.

The overarching principle for network security of the Azure SQL Database offering is to allow only the connection and communication that is necessary to allow the service to operate. All other ports, protocols, and connections are blocked by default. Virtual local area networks (VLANs) and ACLs are used to restrict network communications by source and destination networks, protocols, and port numbers.

Mechanisms that are approved to implement network-based ACLs include ACLs on routers and load balancers. These mechanisms are managed by Azure networking, guest VM firewall, and Azure SQL Database gateway firewall rules, which are configured by the customer.

Data segregation and customer isolation

The Azure production network is structured such that publicly accessible system components are segregated from internal resources. Physical and logical boundaries exist between web servers that provide access to the public-facing Azure portal and the underlying Azure virtual infrastructure, where customer application instances and customer data reside.

All publicly accessible information is managed within the Azure production network. The production network is subject to two-factor authentication and boundary protection mechanisms, uses the firewall and security feature set that is described in the previous section, and uses data isolation functions as noted in the next sections.

Unauthorized systems and isolation of the FC

Because the fabric controller (FC) is the central orchestrator of the Azure fabric, significant controls are in place to mitigate threats to it, especially from potentially compromised FAs within customer applications. The FC does not recognize any hardware whose device information (for example, MAC address) is not pre-loaded within the FC. The DHCP servers on the FC have configured lists of MAC addresses of the nodes they are willing to boot. Even if unauthorized systems are connected, they are not incorporated into fabric inventory, and therefore not connected or authorized to communicate with any system within the fabric inventory. This reduces the risk of unauthorized systems' communicating with the FC and gaining access to the VLAN and Azure.

VLAN isolation

The Azure production network is logically segregated into three primary VLANs:

- The main VLAN: Interconnects untrusted customer nodes.
- The FC VLAN: Contains trusted FCs and supporting systems.
- The device VLAN: Contains trusted network and other infrastructure devices.

Packet filtering

The IPFilter and the software firewalls that are implemented on the root OS and guest OS of the nodes enforce connectivity restrictions and prevent unauthorized traffic between VMs.

Hypervisor, root OS, and guest VMs

The isolation of the root OS from the guest VMs and the guest VMs from one another is managed by the hypervisor and the root OS.

Types of rules on firewalls

A rule is defined as:

{Security Response Center (Src) IP, Src Port, Destination IP, Destination Port, Destination Protocol, In/Out, Stateful/Stateless, Stateful Flow Timeout}.

Synchronous idle character (SYN) packets are allowed in or out only if any one of the rules permits it. For TCP, Azure uses stateless rules where the principle is that it allows only all non-SYN packets into or out of the VM. The security premise is that any host stack is resilient of ignoring a non-SYN if it has not seen a SYN packet previously. The TCP protocol itself is stateful, and in combination with the stateless SYN-based rule achieves an overall behavior of a stateful implementation.

For User Datagram Protocol (UDP), Azure uses a stateful rule. Every time a UDP packet matches a rule, a reverse flow is created in the other direction. This flow has a built-in timeout.

Customers are responsible for setting up their own firewalls on top of what Azure provides. Here customers are able to define the rules for inbound and outbound traffic.

Production configuration management

Standard secure configurations are maintained by respective operations teams in Azure and Azure SQL Database. All configuration changes to production systems are documented and tracked through a central tracking system. Software and hardware changes are tracked through the central tracking system. Networking changes that relate to ACL are tracked using an ACL management service.

All configuration changes to Azure are developed and tested in the staging environment, and they are thereafter deployed in production environment. Software builds are reviewed as part of testing. Security and privacy checks are reviewed as part of entry checklist criteria. Changes are deployed on scheduled intervals by the respective deployment team. Releases are reviewed and signed off by the respective deployment team personnel before they are deployed into production.

Changes are monitored for success. On a failure scenario, the change is rolled back to its previous state or a hotfix is deployed to address the failure with approval of the designated personnel. Source Depot, Git, TFS, Master Data Services (MDS), runners, Azure security monitoring, the FC, and the WinFabric platform are used to centrally manage, apply, and verify the configuration settings in the Azure virtual environment.

Similarly, hardware and network changes have established validation steps to evaluate their adherence to the build requirements. The releases are reviewed and authorized through a coordinated change advisory board (CAB) of respective groups across the stack.

Next steps

To learn more about what Microsoft does to secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure information system components and boundaries](#)
- [Azure network architecture](#)
- [Azure production network](#)

- Azure production operations and management
- Azure infrastructure monitoring
- Azure infrastructure integrity
- Azure customer data protection

Azure production operations and management

7/20/2018 • 3 minutes to read • [Edit Online](#)

The management and operation of the Azure production network is a coordinated effort between the operations teams of Azure and Azure SQL Database. The teams use several system and application performance-monitoring tools in the environment. And they use appropriate tools to monitor network devices, servers, services, and application processes.

To ensure the secure execution of services running in the Azure environment, the operations teams implement multiple levels of monitoring, logging, and reporting, including the following actions:

- Primarily, the Microsoft Monitoring Agent (MMA) gathers monitoring and diagnostic log information from many places, including the fabric controller (FC) and the root operating system (OS), and writes it to log files. The agent eventually pushes a digested subset of the information into a pre-configured Azure storage account. In addition, the freestanding monitoring and diagnostic service reads various monitoring and diagnostic log data and summarizes the information. The monitoring and diagnostic service writes the information to an integrated log. Azure uses the custom-built Azure security monitoring, which is an extension to the Azure monitoring system. It has components that observe, analyze, and report on security-pertinent events from various points in the platform.
- The Azure SQL Database Windows Fabric platform provides management, deployment, development, and operational oversight services for Azure SQL Database. The platform offers distributed, multi-step deployment services, health monitoring, automatic repairs, and service version compliance. It provides the following services:
 - Service modeling capabilities with high-fidelity development environment (datacenter clusters are expensive and scarce).
 - One-click deployment and upgrade workflows for service bootstrap and maintenance.
 - Health reporting with automated repair workflows to enable self-healing.
 - Real time monitoring, alerting, and debugging facilities across the nodes of a distributed system.
 - Centralized collection of operational data and metrics for distributed root cause analysis and service insight.
 - Operational tooling for deployment, change management, and monitoring.
 - The Azure SQL Database Windows Fabric platform and watchdog scripts run continuously and monitor in real time.

If any anomalies occur, the incident response process followed by the Azure incident triage team is activated. The appropriate Azure support personnel are notified to respond to the incident. Issue tracking and resolution are documented and managed in a centralized ticketing system. System uptime metrics are available under the non-disclosure agreement (NDA) and upon request.

Corporate network and multi-factor access to production

The corporate network user base includes Azure support personnel. The corporate network supports internal corporate functions and includes access to internal applications that are used for Azure customer support. The corporate network is both logically and physically separated from the Azure production network. Azure personnel access the corporate network by using Azure workstations and laptops. All users must have an Azure Active Directory (Azure AD) account, including a username and password, to access corporate network resources. Corporate network access uses Azure AD accounts, which are issued to all Microsoft personnel, contractors, and vendors and managed by Microsoft Information Technology. Unique user identifiers distinguish

personnel based on their employment status at Microsoft.

Access to internal Azure applications is controlled through authentication with Active Directory Federation Services (AD FS). AD FS is a service hosted by Microsoft Information Technology that provides authentication of corporate network users through applying a secure token and user claims. AD FS enables internal Azure applications to authenticate users against the Microsoft corporate active directory domain. To access the production network from the corporate network environment, users must authenticate by using multi-factor authentication.

Next steps

To learn more about what Microsoft does to secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure information system components and boundaries](#)
- [Azure network architecture](#)
- [Azure production network](#)
- [Azure SQL Database security features](#)
- [Azure infrastructure monitoring](#)
- [Azure infrastructure integrity](#)
- [Azure customer data protection](#)

Azure infrastructure monitoring

7/20/2018 • 2 minutes to read • [Edit Online](#)

Configuration and change management

Azure reviews and updates configuration settings and baseline configurations of hardware, software, and network devices annually. Changes are developed, tested, and approved prior to entering the production environment from a development and/or test environment.

The baseline configurations that are required for Azure-based services are reviewed by the Azure security and compliance team and by service teams. A service team review is part of the testing that occurs before the deployment of their production service.

Vulnerability management

Security update management helps protect systems from known vulnerabilities. Azure uses integrated deployment systems to manage the distribution and installation of security updates for Microsoft software. Azure is also able to draw on the resources of the Microsoft Security Response Center (MSRC). The MSRC identifies, monitors, responds to, and resolves security incidents and cloud vulnerabilities around the clock, every day of the year.

Vulnerability scanning

Vulnerability scanning is performed on server operating systems, databases, and network devices. The vulnerability scans are performed on a quarterly basis at minimum. Azure contracts with independent assessors to perform penetration testing of the Azure boundary. Red-team exercises are also routinely performed and the results are used to make security improvements.

Protective monitoring

Azure security has defined requirements for active monitoring. Service teams configure active monitoring tools in accordance with these requirements. Active monitoring tools include the Microsoft Monitoring Agent (MMA) and System Center Operations Manager. These tools are configured to provide time alerts to Azure security personnel in situations that require immediate action.

Incident management

Microsoft implements a security incident management process to facilitate a coordinated response to incidents, should one occur.

If Microsoft becomes aware of unauthorized access to customer data that's stored on its equipment or in its facilities, or it becomes aware of unauthorized access to such equipment or facilities resulting in loss, disclosure, or alteration of customer data, Microsoft takes the following actions:

- Promptly notifies the customer of the security incident.
- Promptly investigates the security incident and provides customers detailed information about the security incident.
- Takes reasonable and prompt steps to mitigate the effects and minimize any damage resulting from the security incident.

An incident management framework has been established that defines roles and allocates responsibilities. The

Azure security incident management team is responsible for managing security incidents, including escalation, and ensuring the involvement of specialist teams when necessary. Azure operations managers are responsible for overseeing the investigation and resolution of security and privacy incidents.

Next steps

To learn more about what Microsoft does to secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure information system components and boundaries](#)
- [Azure network architecture](#)
- [Azure production network](#)
- [Azure SQL Database security features](#)
- [Azure production operations and management](#)
- [Azure infrastructure integrity](#)
- [Azure customer data protection](#)

Azure infrastructure integrity

7/20/2018 • 3 minutes to read • [Edit Online](#)

Software installation

All components in the software stack that are installed in the Azure environment are custom built following the Microsoft Security Development Lifecycle (SDL) process. All software components, including operating system (OS) images and SQL Database, are deployed as part of the change management and release management process. The OS that runs on all nodes is a customized version of Windows Server 2008 or Windows Server 2012. The exact version is chosen by the fabric controller (FC) according to the role it intends for the OS to play. In addition, the host OS does not allow installation of any unauthorized software components.

Some Azure components are deployed as Azure customers on a guest VM running on a guest OS.

Virus scans on builds

Azure software component (including OS) builds have to undergo a virus scan that uses the Endpoint Protection anti-virus tool. Each virus scan creates a log within the associated build directory, detailing what was scanned and the results of the scan. The virus scan is part of the build source code for every component within Azure. Code is not moved to production without having a clean and successful virus scan. If any issues are noted, the build is frozen and then goes to the security teams within Microsoft Security to identify where the "rogue" code entered the build.

Closed and locked environment

By default, Azure infrastructure nodes and guest VMs do not have user accounts created on them. In addition, default Windows administrator accounts are also disabled. Administrators from Azure live support can, with proper authentication, log into these machines and administer the Azure production network for emergency repairs.

Azure SQL Database authentication

As with any implementation of SQL Server, user account management must be tightly controlled. Azure SQL Database supports only SQL Server authentication. To complement a customer's data security model, user accounts with strong passwords and configured with specific rights should be used as well.

ACLs and firewalls between the Microsoft corporate network and an Azure cluster

Access-control lists (ACLs) and firewalls between the service platform and the Microsoft corporate network protect SQL Database instances from unauthorized insider access. Further, only users from IP address ranges from the Microsoft corporate network can access the Windows Fabric platform-management endpoint.

ACLs and firewalls between nodes in a SQL Database cluster

As an additional protection, as part of the defense-in depth-strategy, ACLs and a firewall have been implemented between nodes in a SQL Database cluster. All communication inside the Windows Fabric platform cluster as well as all running code is trusted.

Custom monitoring agents

SQL Database employs custom monitoring agents (MAs), also called watchdogs, to monitor the health of the SQL Database cluster.

Web protocols

Role instance monitoring and restart

Azure ensures that all deployed, running roles (internet-facing web, or back-end processing worker roles) are subject to sustained health monitoring to ensure that they effectively and efficiently deliver the services for which they've been provisioned. If a role becomes unhealthy, by either a critical fault in the application that's being hosted or an underlying configuration problem within the role instance itself, the FC detects the problem within the role instance and initiates a corrective state.

Compute connectivity

Azure ensures that the deployed application or service is reachable via standard web-based protocols. Virtual instances of internet-facing web roles have external internet connectivity and are reachable directly by web users. To protect the sensitivity and integrity of the operations that worker roles perform on behalf of the publicly-accessible web role virtual instances, virtual instances of back-end processing worker roles have external internet connectivity but cannot be accessed directly by external web users.

Next steps

To learn more about what Microsoft does to secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure information system components and boundaries](#)
- [Azure network architecture](#)
- [Azure production network](#)
- [Azure SQL Database security features](#)
- [Azure production operations and management](#)
- [Azure infrastructure monitoring](#)
- [Azure customer data protection](#)

Azure customer data protection

7/20/2018 • 4 minutes to read • [Edit Online](#)

Access to customer data by Microsoft operations and support personnel is denied by default. When access to customer data is granted, leadership approval is required and then access is carefully managed and logged. The access-control requirements are established by the following Azure Security Policy:

- No access to customer data, by default.
- No user or administrator accounts on customer virtual machines (VMs).
- Grant the least privilege that's required to complete task; audit and log access requests.

Azure support personnel are assigned unique corporate Active Directory accounts by Microsoft. Azure relies on Microsoft corporate Active Directory, managed by Microsoft Information Technology (MSIT), to control access to key information systems. Multi-factor authentication is required, and access is granted only from secure consoles.

All access attempts are monitored and can be displayed via a basic set of reports.

Data protection

Azure provides customers with strong data security, both by default and as customer options.

Data segregation: Azure is a multi-tenant service, which means that multiple customer deployments and VMs are stored on the same physical hardware. Azure uses logical isolation to segregate each customer's data from the data of others. Segregation provides the scale and economic benefits of multi-tenant services while rigorously preventing customers from accessing one another's data.

At-rest data protection: Customers are responsible for ensuring that data stored in Azure is encrypted in accordance with their standards. Azure offers a wide range of encryption capabilities, giving customers the flexibility to choose the solution that best meets their needs. Azure Key Vault helps customers easily maintain control of keys that are used by cloud applications and services to encrypt data. Azure Disk Encryption enables customers to encrypt VMs. Azure Storage Service Encryption makes it possible to encrypt all data that's placed into a customer's storage account.

In-transit data protection: Customers can enable encryption for traffic between their own VMs and end users. Azure protects data in transit to or from outside components and data in transit internally, such as between two virtual networks. Azure uses the industry-standard Transport Layer Security (TLS) 1.2 or later protocol with 2,048-bit RSA/SHA256 encryption keys, as recommended by CESG/NCSC, to encrypt communications between:

- The customer and the cloud.
- Internally between Azure systems and datacenters.

Encryption: Encryption of data in storage and in transit can be deployed by customers as a best practice for ensuring confidentiality and integrity of data. It is straightforward for customers to configure their Azure cloud services to use SSL to protect communications from the internet and even between their Azure-hosted VMs.

Data redundancy: Microsoft helps ensure that data is protected if there is a cyberattack or physical damage to a datacenter. Customers may opt for:

- In-country storage for compliance or latency considerations.
- Out-of-country storage for security or disaster recovery purposes.

Data can be replicated within a selected geographic area for redundancy but cannot be transmitted outside it. Customers have multiple options for replicating data, including the number of copies and the number and location of replication datacenters.

When you create your storage account, select one of the following replication options:

- **Locally redundant storage (LRS):** Locally redundant storage maintains three copies of your data. LRS is replicated three times within a single facility in a single region. LRS protects your data from normal hardware failures, but not from a failure of a single facility.
- **Zone-redundant storage (ZRS):** Zone-redundant storage maintains three copies of your data. ZRS is replicated three times across two to three facilities to provide higher durability than LRS. Replication occurs within a single region or across two regions. ZRS helps ensure that your data is durable within a single region.
- **Geo-redundant storage (GRS):** Geo-redundant storage is enabled for your storage account by default when you create it. GRS maintains six copies of your data. With GRS, your data is replicated three times within the primary region. Your data is also replicated three times in a secondary region hundreds of miles away from the primary region, providing the highest level of durability. In the event of a failure at the primary region, Azure Storage fails over to the secondary region. GRS helps ensure that your data is durable in two separate regions.

Data destruction: When customers delete data or leave Azure, Microsoft follows strict standards for overwriting storage resources before their reuse, as well as the physical destruction of decommissioned hardware. Microsoft executes a complete deletion of data on customer request and on contract termination.

Customer data ownership

Microsoft does not inspect, approve, or monitor applications that customers deploy to Azure. Moreover, Microsoft does not know what kind of data customers choose to store in Azure. Microsoft does not claim data ownership over the customer information that's entered into Azure.

Records management

Azure has established internal records-retention requirements for back-end data. Customers are responsible for identifying their own record retention requirements. For records that are stored in Azure, customers are responsible for extracting their data and retaining their content outside of Azure for a customer-specified retention period.

Azure allows customers to export data and audit reports from the product. The exports are saved locally to retain the information for a customer-defined retention time period.

Electronic discovery (e-discovery)

Azure customers are responsible for complying with e-discovery requirements in their use of Azure services. If Azure customers must preserve their customer data, they may export and save the data locally. Additionally, customers can request exports of their data from the Azure Customer Support department. In addition to allowing customers to export their data, Azure conducts extensive logging and monitoring internally.

Next steps

To learn more about what Microsoft does to secure the Azure infrastructure, see:

- [Azure facilities, premises, and physical security](#)
- [Azure infrastructure availability](#)
- [Azure information system components and boundaries](#)

- Azure network architecture
- Azure production network
- Azure SQL Database security features
- Azure production operations and management
- Azure infrastructure monitoring
- Azure infrastructure integrity

Microsoft Antimalware for Azure Cloud Services and Virtual Machines

11/7/2018 • 10 minutes to read • [Edit Online](#)

Microsoft Antimalware for Azure is a free real-time protection that helps identify and remove viruses, spyware, and other malicious software. It generates alerts when known malicious or unwanted software tries to install itself or run on your Azure systems.

The solution is built on the same antimalware platform as Microsoft Security Essentials [MSE], Microsoft Forefront Endpoint Protection, Microsoft System Center Endpoint Protection, Windows Intune, and Windows Defender. Microsoft Antimalware for Azure is a single-agent solution for applications and tenant environments, designed to run in the background without human intervention. Protection may be deployed based on the needs of application workloads, with either basic secure-by-default or advanced custom configuration, including antimalware monitoring.

When you deploy and enable Microsoft Antimalware for Azure for your applications, the following core features are available:

- **Real-time protection** - monitors activity in Cloud Services and on Virtual Machines to detect and block malware execution.
- **Scheduled scanning** - Scans periodically to detect malware, including actively running programs.
- **Malware remediation** - automatically takes action on detected malware, such as deleting or quarantining malicious files and cleaning up malicious registry entries.
- **Signature updates** - automatically installs the latest protection signatures (virus definitions) to ensure protection is up-to-date on a pre-determined frequency.
- **Antimalware Engine updates** – automatically updates the Microsoft Antimalware engine.
- **Antimalware Platform updates** – automatically updates the Microsoft Antimalware platform.
- **Active protection** - reports telemetry metadata about detected threats and suspicious resources to Microsoft Azure to ensure rapid response to the evolving threat landscape, as well as enabling real-time synchronous signature delivery through the Microsoft Active Protection System (MAPS).
- **Samples reporting** - provides and reports samples to the Microsoft Antimalware service to help refine the service and enable troubleshooting.
- **Exclusions** – allows application and service administrators to configure exclusions for files, processes, and drives.
- **Antimalware event collection** - records the antimalware service health, suspicious activities, and remediation actions taken in the operating system event log and collects them into the customer's Azure Storage account.

NOTE

Microsoft Antimalware can also be deployed using Azure Security Center. Read [Install Endpoint Protection in Azure Security Center](#) for more information.

Architecture

Microsoft Antimalware for Azure includes the Microsoft Antimalware Client and Service, Antimalware classic deployment model, Antimalware PowerShell cmdlets, and Azure Diagnostics Extension. Microsoft Antimalware is supported on Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2 operating system families. It is not supported on the Windows Server 2008 operating system, and also is not supported in Linux.

The Microsoft Antimalware Client and Service is installed by default in a disabled state in all supported Azure guest operating system families in the Cloud Services platform. The Microsoft Antimalware Client and Service is not installed by default in the Virtual Machines platform and is available as an optional feature through the Azure portal and Visual Studio Virtual Machine configuration under Security Extensions.

When using Azure App Service, the underlying service that hosts the web app has Microsoft Antimalware enabled on it. This is used to protect Azure App Service infrastructure and does not run on customer content.

NOTE

Windows Defender is the built-in Antimalware enabled in Windows Server 2016. The Windows Defender Interface is also enabled by default on some Windows Server 2016 SKU's [see here for more information](#). The Azure VM Antimalware extension can still be added to a Windows Server 2016 Azure VM with Windows Defender, but in this scenario the extension will apply any optional [configuration policies](#) to be used by Windows Defender, the extension will not deploy any additional antimalware services. You can read more about this update [here](#).

Microsoft antimalware workflow

The Azure service administrator can enable Antimalware for Azure with a default or custom configuration for your Virtual Machines and Cloud Services using the following options:

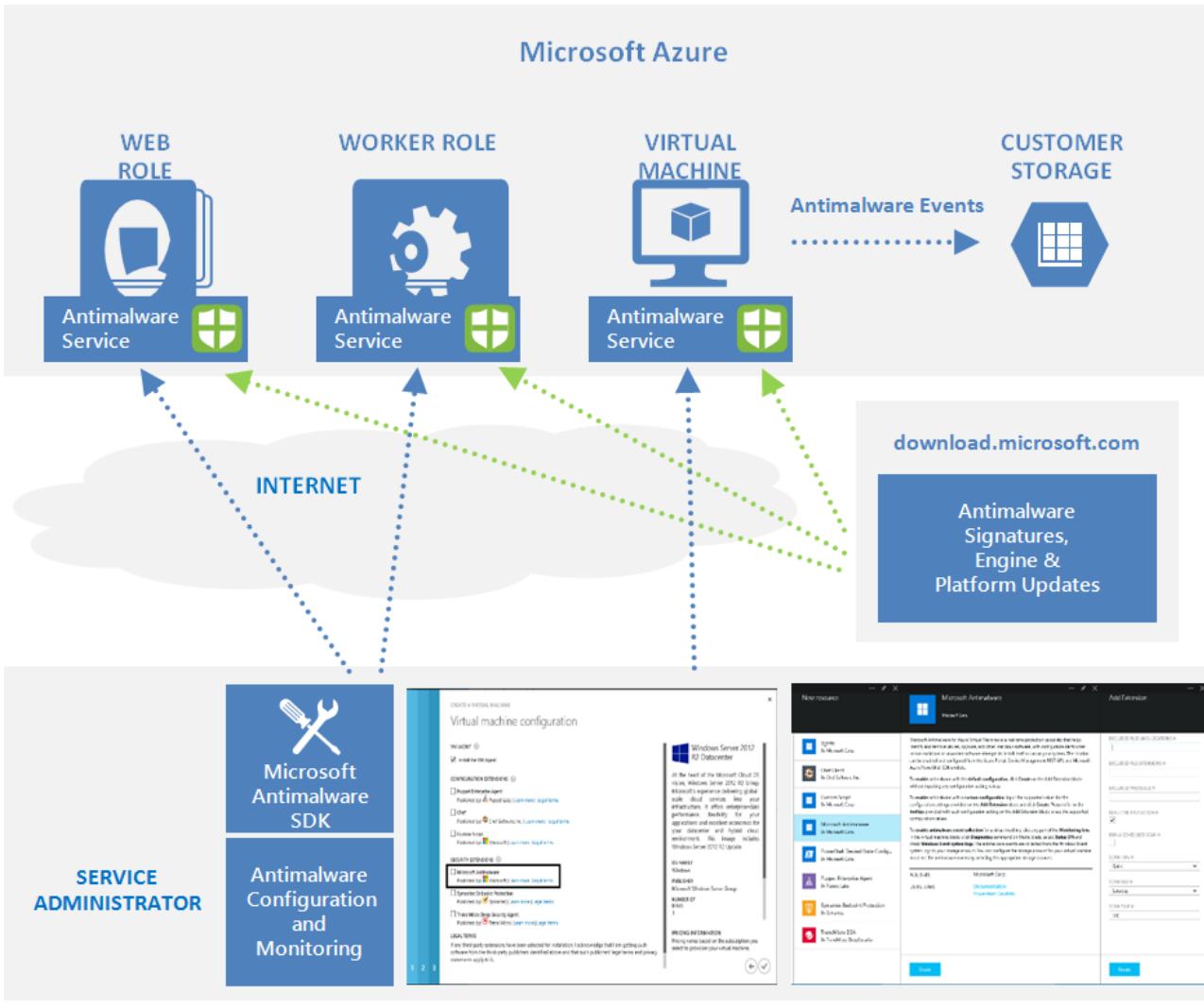
- Virtual Machines – In the Azure portal, under **Security Extensions**
- Virtual Machines – Using the Visual Studio virtual machines configuration in Server Explorer
- Virtual Machines and Cloud Services – Using the Antimalware [classic deployment model](#)
- Virtual Machines and Cloud Services – Using Antimalware PowerShell cmdlets

The Azure portal or PowerShell cmdlets push the Antimalware extension package file to the Azure system at a pre-determined fixed location. The Azure Guest Agent (or the Fabric Agent) launches the Antimalware Extension, applying the Antimalware configuration settings supplied as input. This step enables the Antimalware service with either default or custom configuration settings. If no custom configuration is provided, then the antimalware service is enabled with the default configuration settings. Refer to the *Antimalware configuration* section in the [Microsoft Antimalware for Azure – Code Samples](#) for more details.

Once running, the Microsoft Antimalware client downloads the latest protection engine and signature definitions from the Internet and loads them on the Azure system. The Microsoft Antimalware service writes service-related events to the system OS events log under the "Microsoft Antimalware" event source. Events include the Antimalware client health state, protection and remediation status, new and old configuration settings, engine updates and signature definitions, and others.

You can enable Antimalware monitoring for your Cloud Service or Virtual Machine to have the Antimalware event log events written as they are produced to your Azure storage account. The Antimalware Service uses the Azure Diagnostics extension to collect Antimalware events from the Azure system into tables in the customer's Azure Storage account.

The deployment workflow including configuration steps and options supported for the above scenarios are documented in [Antimalware deployment scenarios](#) section of this document.



NOTE

You can however use Powershell/APIs and Azure Resource Manager templates to deploy Virtual Machine Scale Sets with the Microsoft Anti-Malware extension. For installing an extension on an already running Virtual Machine, you can use the sample python script [vmssext.py](#). This script gets the existing extension config on the Scale Set and adds an extension to the list of existing extensions on the VM Scale Sets.

Default and Custom Antimalware Configuration

The default configuration settings are applied to enable Antimalware for Azure Cloud Services or Virtual Machines when you do not provide custom configuration settings. The default configuration settings have been pre-optimized for running in the Azure environment. Optionally, you can customize these default configuration settings as required for your Azure application or service deployment and apply them for other deployment scenarios.

NOTE

By default the Microsoft Antimalware User Interface on Azure Resource Manager is disabled, [cleanuppolicy.xml](#) file to bypass this error message is not supported. For information on how to create a custom policy, read [Enabling Microsoft Antimalware User Interface on Azure Resource Manager VMs Post Deployment](#).

The following table summarizes the configuration settings available for the Antimalware service. The default configuration settings are marked under the column labeled "Default" below.

Setting	Options	Default	Description
Enable Antimalware	true	None	true - Enables the Antimalware service

	(lower case sensitive)		false – not supported Note – This is a required configuration setting to enable the Antimalware service
Exclusions Extensions	extension1, extension2,	None	List of file extensions to exclude from scanning. Example: gif, log, txt excludes files with the .gif, .log, or .txt extension from being scanned. Each excluded file extension should be added as a separate row element value in your antimalware XML configuration or semicolon separated in antimalware JSON configuration
Exclusions Paths	path1, path2	None	List of paths to files or folders to exclude from scanning. Example: e:\approot\worker.dll, e:\approot\temp excludes the file worker.dll in the e:\approot folder and anything under the folder e:\approot\temp from being scanned. Note: For antimalware JSON configuration for virtual machines, use two backslashes (\\\) instead of one to escape properly. For example: e:\\approot\\\\worker.dll Each excluded path should be added as a separate row element value in your antimalware XML configuration or semicolon separated in antimalware JSON configuration
Exclusions Processes	process1, process2,	None	List of process exclusions. Any file opened by an excluded process will not be scanned (the process itself will still be scanned – to exclude the process itself, use the ExcludedPaths option). Example: C:\Program Files\MyApp.exe excludes any files opened by MyApp.exe from being scanned. Each excluded process should be added as a separate row element value in your antimalware XML configuration or semicolon separated in antimalware JSON configuration
RealtimeProtectionEnabled	true false (lower case sensitive)	true	true – Enables real-time protection false – Disables real-time protection Default = true when AntimalwareEnabled = true
ScheduledScanSettings isEnabled	true false (lower case sensitive)	false	Enables or disables a periodic scan for active malware on the system Default = false
ScheduledScanSettings Day	0 – 8	7	0 – scan daily, 1 – Sunday, 2 – Monday, 3 – Tuesday..., 7 – Saturday, 8 – disabled Default = 7 if only ScheduledScanSettings isEnabled = true
ScheduledScanSettings Time	0 – 1440	120	Hour at which to begin the scheduled scan. Measured in 60 minute increments from midnight 60 mins = 1:00 AM 120 mins = 2:00 AM

			<p>...</p> <p>1380 mins = 11:00 PM</p> <p>Default = 120 mins if ScheduledScanSettings isEnabled = true</p>
ScheduledScanSettings Scan Type	Quick/Full	Quick	Default = Quick if ScheduledScanSettings isEnabled = true
Monitoring	ON OFF	OFF	<p>ON - Enable Antimalware event collection to user subscription storage using Azure Diagnostics extension</p> <p>OFF – Disable Antimalware event collection to user subscription storage by removing antimalware monitoring configuration in Azure Diagnostics extension if it was previously turned ON</p>
StorageAccountName	Storage Account Name	None	Storage account name for your Azure store table to collect antimalware events in storage Note - Storage account name is required if monitoring is specified as ON

Antimalware Deployment Scenarios

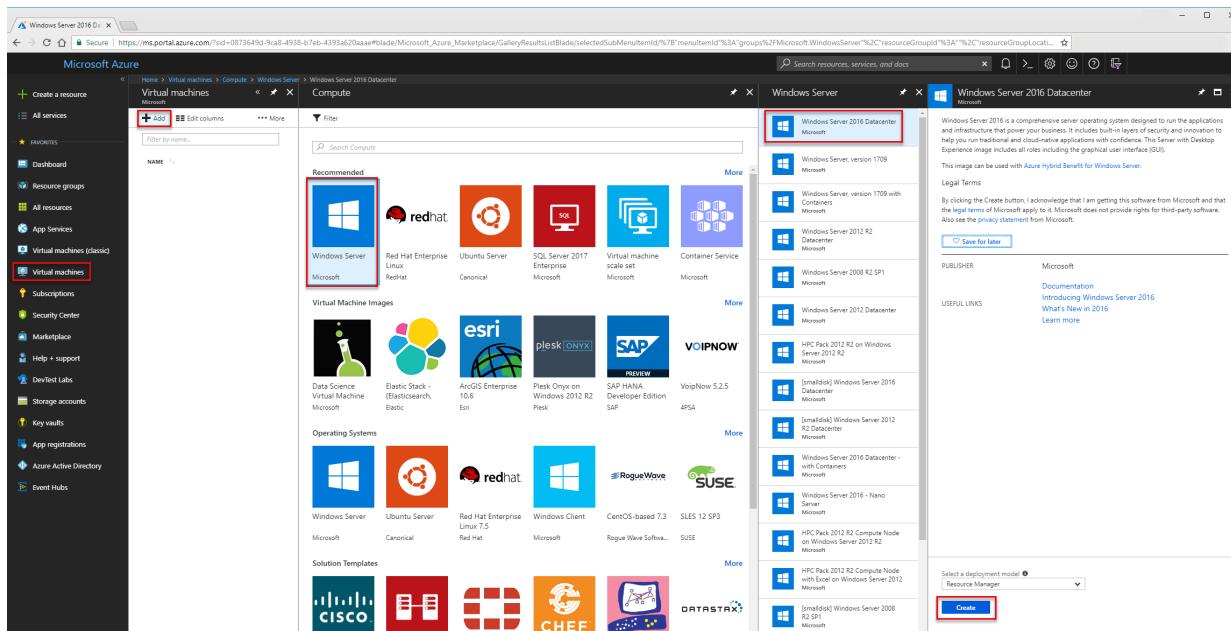
The scenarios to enable and configure antimalware, including monitoring for Azure Cloud Services and Virtual Machines, are discussed in this section.

Virtual machines - enable and configure antimalware

Deployment While creating a VM using the Azure portal

To enable and configure Microsoft Antimalware for Azure Virtual Machines using the Azure portal while provisioning a Virtual Machine, follow the steps below:

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. To create a new virtual machine, navigate to **Virtual machines**, select **Add**, and choose **Windows Server**.
3. Select the version of Windows server that you would like to use.
4. Select **Create**.



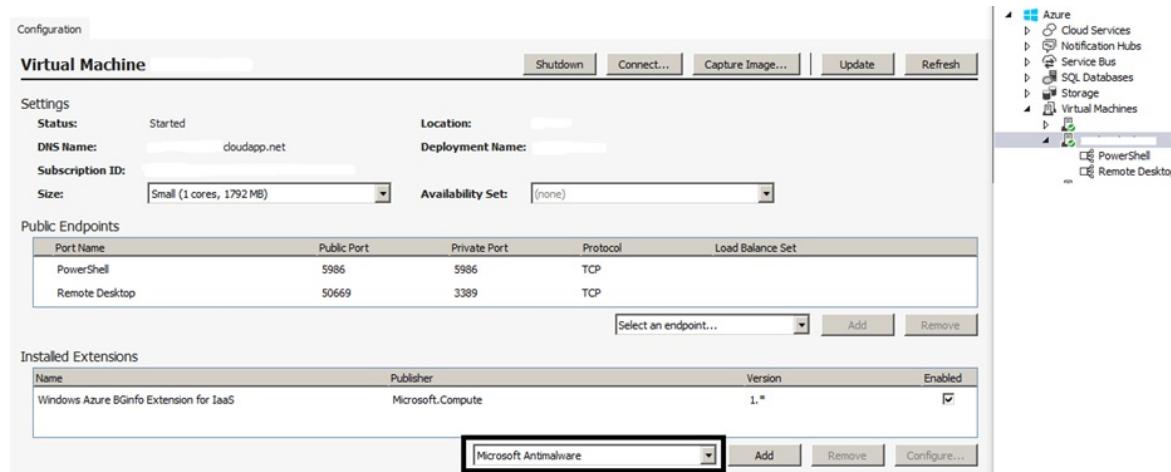
5. Provide a **Name**, **Username**, **Password**, and create a new resource group or choose an existing resource group.

6. Select **Ok**.
7. Choose a vm size.
8. In the next section, make the appropriate choices for your needs select the **Extensions** section.
9. Select **Add extension**
10. Under **New resource**, choose **Microsoft Antimalware**.
11. Select **Create**
12. In the **Install extension** section file, locations, and process exclusions can be configured as well as other scan options. Choose **Ok**.
13. Choose **Ok**.
14. Back in the **Settings** section, choose **Ok**.
15. In the **Create** screen, choose **Ok**.

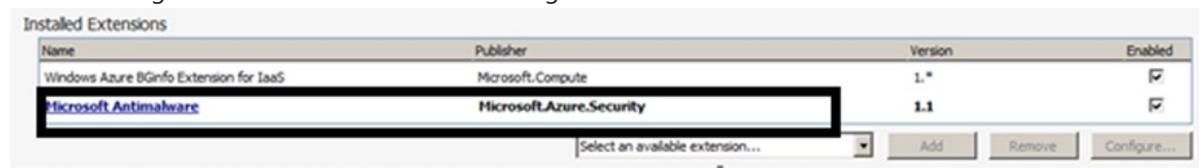
Deployment using the Visual Studio virtual machine configuration

To enable and configure the Microsoft Antimalware service using Visual Studio:

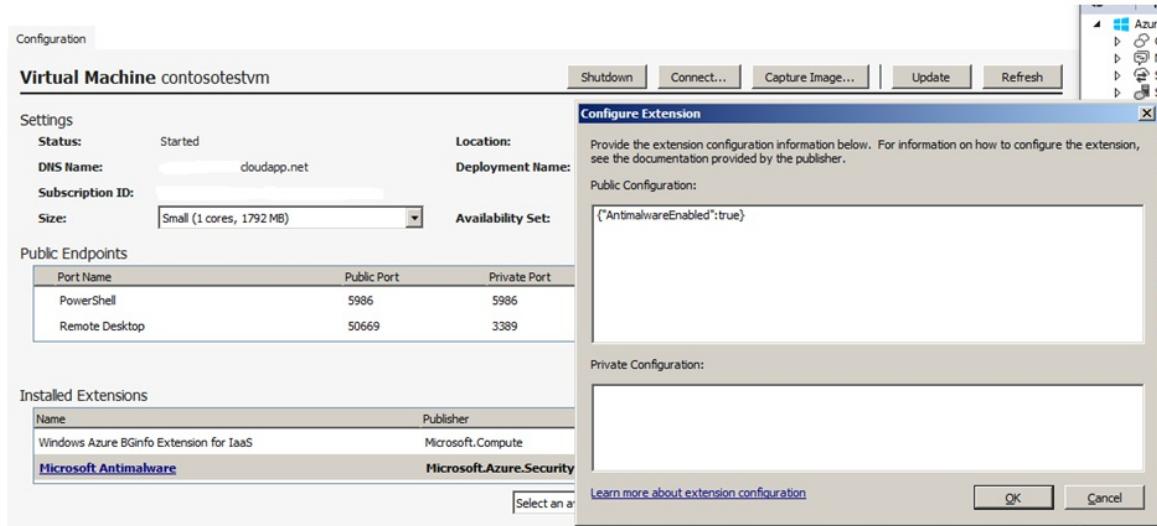
1. Connect to Microsoft Azure in Visual Studio.
2. Choose your Virtual Machine in the **Virtual Machines** node in **Server Explorer**



3. Right-click **configure** to view the Virtual Machine configuration page
4. Select **Microsoft Antimalware** extension from the dropdown list under **Installed Extensions** and click **Add** to configure with default antimalware configuration.



5. To customize the default Antimalware configuration, select (highlight) the Antimalware extension in the installed extensions list and click **Configure**.
6. Replace the default Antimalware configuration with your custom configuration in supported JSON format in the **public configuration** textbox and click OK.
7. Click the **Update** button to push the configuration updates to your Virtual Machine.



NOTE

The Visual Studio Virtual Machines configuration for Antimalware supports only JSON format configuration. The Antimalware JSON configuration settings template is included in the [Microsoft Antimalware For Azure - Code Samples](#), showing the supported Antimalware configuration settings.

Deployment Using PowerShell cmdlets

An Azure application or service can enable and configure Microsoft Antimalware for Azure Virtual Machines using PowerShell cmdlets.

To enable and configure Microsoft antimalware using antimalware PowerShell cmdlets:

1. Set up your PowerShell environment - Refer to the documentation at <https://github.com/Azure/azure-powershell>
2. Use the [Set-AzureVMMicrosoftAntimalwareExtension](#) Antimalware cmdlet to enable and configure Microsoft Antimalware for your Virtual Machine.

NOTE

The Azure Virtual Machines configuration for Antimalware supports only JSON format configuration. The Antimalware JSON configuration settings template is included in the [Microsoft Antimalware For Azure - Code Samples](#), showing the supported Antimalware configuration settings.

Enable and Configure Antimalware Using PowerShell cmdlets

An Azure application or service can enable and configure Microsoft Antimalware for Azure Cloud Services using PowerShell cmdlets. Note that Microsoft Antimalware is installed in a disabled state in the Cloud Services platform and requires an action by an Azure application to enable it.

To enable and configure Microsoft Antimalware using PowerShell cmdlets:

1. Set up your PowerShell environment - Refer to the documentation at <https://github.com/Azure/azure-powershell>
2. Use the [Set-AzureServiceAntimalwareExtension](#) Antimalware cmdlet to enable and configure Microsoft Antimalware for your Cloud Service.

The Antimalware XML configuration settings template is included in the [Microsoft Antimalware For Azure - Code Samples](#), showing the supported Antimalware configuration settings.

Cloud Services and Virtual Machines - Configuration Using PowerShell cmdlets

An Azure application or service can retrieve the Microsoft Antimalware configuration for Cloud Services and Virtual Machines using PowerShell cmdlets.

To retrieve the Microsoft Antimalware configuration using PowerShell cmdlets:

1. Set up your PowerShell environment - Refer to the documentation at <https://github.com/Azure/azure-powershell>
2. **For Virtual Machines:** Use the [Get-AzureVMMicrosoftAntimalwareExtension Antimalware cmdlet](#) to get the antimalware configuration.
3. **For Cloud Services:** Use the [Get-AzureServiceAntimalwareConfig Antimalware cmdlet](#) to get the Antimalware configuration.

Remove Antimalware Configuration Using PowerShell cmdlets

An Azure application or service can remove the Antimalware configuration and any associated Antimalware monitoring configuration from the relevant Azure Antimalware and diagnostics service extensions associated with the Cloud Service or Virtual Machine.

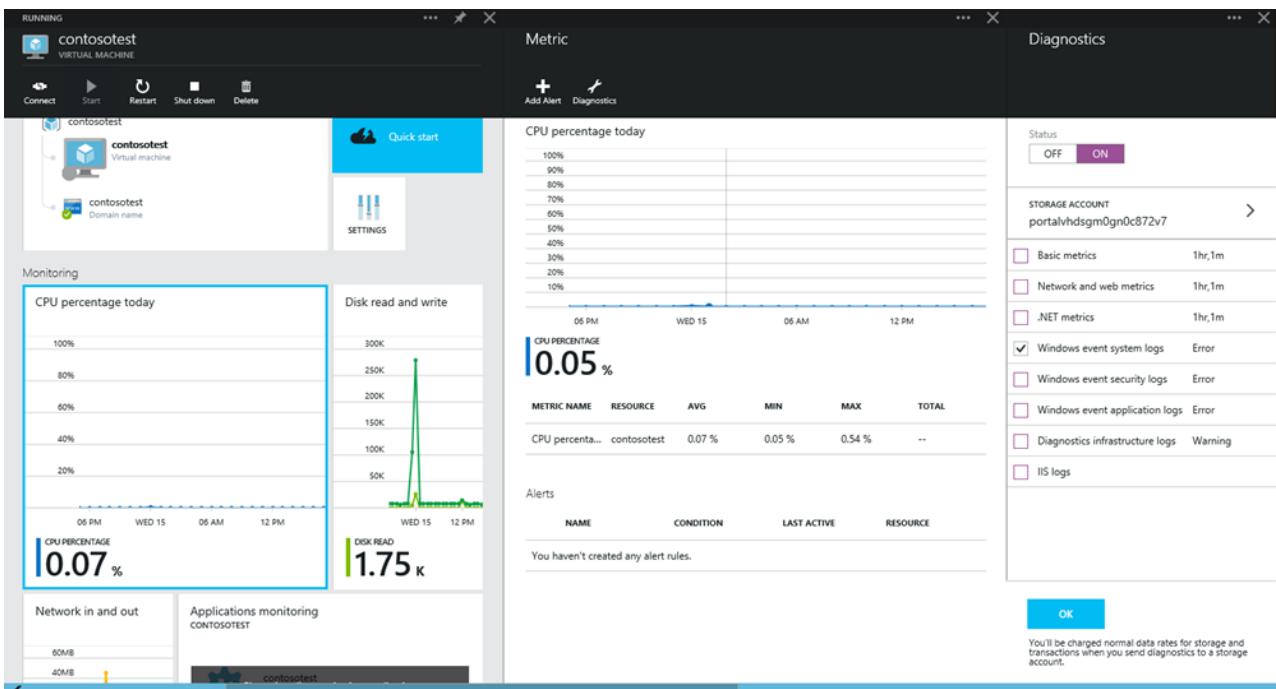
To remove Microsoft Antimalware using PowerShell cmdlets:

1. Set up your PowerShell environment - Refer to the documentation at <https://github.com/Azure/azure-powershell>
2. **For Virtual Machines:** Use the [Remove-AzureVMMicrosoftAntimalwareExtension Antimalware cmdlet](#).
3. **For Cloud Services:** Use the [Remove-AzureServiceAntimalwareExtension Antimalware cmdlet](#).

To **enable** antimalware event collection for a virtual machine using the Azure Preview Portal:

1. Click any part of the Monitoring lens in the Virtual Machine blade
2. Click the Diagnostics command on Metric blade
3. Select **Status ON** and check the option for Windows event system
4. You can choose to uncheck all other options in the list, or leave them enabled per your application service needs.
5. The Antimalware event categories "Error", "Warning", "Informational", etc., are captured in your Azure Storage account.

Antimalware events are collected from the Windows event system logs to your Azure Storage account. You can configure the Storage Account for your Virtual Machine to collect Antimalware events by selecting the appropriate storage account.



NOTE

For more information on how to Diagnostics Logging for Azure Antimalware, read [Enabling Diagnostics Logging for Azure Antimalware](#).

Enable and configure antimalware monitoring using PowerShell cmdlets

You can enable collection of Microsoft Antimalware events for your Cloud Service or Virtual Machine using Azure Diagnostics through Antimalware PowerShell cmdlets. The Azure Diagnostics extension can be configured to capture events from the System event log source "Microsoft Antimalware" to your Azure Storage account. The Antimalware event categories "Error", "Warning", "Informational", etc., are captured in your Azure Storage account.

To enable Antimalware event collection to your Azure Storage account using PowerShell cmdlets:

1. Set up your PowerShell environment - Refer to <https://github.com/Azure/azure-powershell>
2. **For Virtual Machines** - Use the [Set-AzureVMMicrosoftAntimalwareExtension](#) Antimalware cmdlet with the Monitoring ON option.
3. **For Cloud Services** - Use the [Set-AzureServiceAntimalwareExtension](#) Antimalware cmdlet with the Monitoring ON option.

You can view the Antimalware raw events by looking at the *WADWindowsEventLogsTable* table in your Azure Storage account that you configured to enable Antimalware monitoring. This can be useful to validate that Antimalware event collection is working, including getting insight into the Antimalware service's health. For more information, including sample code on how to extract Antimalware events from your storage account, see [Microsoft Antimalware For Azure - Code Samples](#).

Azure Virtual Machines security overview

12/5/2018 • 8 minutes to read • [Edit Online](#)

You can use Azure Virtual Machines to deploy a wide range of computing solutions in an agile way. The service supports Microsoft Windows, Linux, Microsoft SQL Server, Oracle, IBM, SAP, and Azure BizTalk Services. So you can deploy any workload and any language on nearly any operating system.

An Azure virtual machine gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs the virtual machine. You can build and deploy your applications with the assurance that your data is protected and safe in highly secure datacenters.

With Azure, you can build security-enhanced, compliant solutions that:

- Protect your virtual machines from viruses and malware.
- Encrypt your sensitive data.
- Secure network traffic.
- Identify and detect threats.
- Meet compliance requirements.

The goal of this article is to provide an overview of the core Azure security features that can be used with virtual machines. Links to articles give details of each feature so you can learn more.

Antimalware

With Azure, you can use antimalware software from security vendors such as Microsoft, Symantec, Trend Micro, and Kaspersky. This software helps protect your virtual machines from malicious files, adware, and other threats.

Microsoft Antimalware for Azure Cloud Services and Virtual Machines is a real-time protection capability that helps identify and remove viruses, spyware, and other malicious software. Microsoft Antimalware for Azure provides configurable alerts when known malicious or unwanted software attempts to install itself or run on your Azure systems.

Microsoft Antimalware for Azure is a single-agent solution for applications and tenant environments. It's designed to run in the background without human intervention. You can deploy protection based on the needs of your application workloads, with either basic secure-by-default or advanced custom configuration, including antimalware monitoring.

When you deploy and enable Microsoft Antimalware for Azure, the following core features are available:

- **Real-time protection:** Monitors activity in Cloud Services and on Virtual Machines to detect and block malware execution.
- **Scheduled scanning:** Periodically performs targeted scanning to detect malware, including actively running programs.
- **Malware remediation:** Automatically takes action on detected malware, such as deleting or quarantining malicious files and cleaning up malicious registry entries.
- **Signature updates:** Automatically installs the latest protection signatures (virus definitions) to ensure that protection is up-to-date on a pre-determined frequency.
- **Antimalware engine updates:** Automatically updates the Microsoft Antimalware for Azure engine.
- **Antimalware platform updates:** Automatically updates the Microsoft Antimalware for Azure platform.
- **Active protection:** Reports telemetry metadata to Azure about detected threats and suspicious resources to ensure rapid response. Enables real-time synchronous signature delivery through the Microsoft Active

Protection System (MAPS).

- **Samples reporting:** Provides and reports samples to the Microsoft Antimalware for Azure service to help refine the service and enable troubleshooting.
- **Exclusions:** Allows application and service administrators to configure certain files, processes, and drives to exclude them from protection and scanning for performance and other reasons.
- **Antimalware event collection:** Records antimalware service health, suspicious activities, and remediation actions taken in the operating system event log and collects them in your Azure storage account.

Learn more about antimalware software to help protect your virtual machines:

- [Microsoft Antimalware for Azure Cloud Services and Virtual Machines](#)
- [Deploying Antimalware Solutions on Azure Virtual Machines](#)
- [How to install and configure Trend Micro Deep Security as a service on a Windows VM](#)
- [How to install and configure Symantec Endpoint Protection on a Windows VM](#)
- [Security solutions in the Azure Marketplace](#)

For even more powerful protection, consider using [Windows Defender Advanced Threat Protection](#). With Windows Defender ATP, you get:

- [Attack surface reduction](#)
- [Next generation protection](#)
- [Endpoint protection and response](#)
- [Automated investigation and remediation](#)
- [Secure score](#)
- [Advanced hunting](#)
- [Management and APIs](#)
- [Microsoft Threat Protection](#)

Learn more:

- [Get Started with WDATP](#)
- [Overview of WDATP capabilities](#)

Hardware security module

Improving key security can enhance encryption and authentication protections. You can simplify the management and security of your critical secrets and keys by storing them in Azure Key Vault.

Key Vault provides the option to store your keys in hardware security modules (HSMs) certified to FIPS 140-2 Level 2 standards. Your SQL Server encryption keys for backup or [transparent data encryption](#) can all be stored in Key Vault with any keys or secrets from your applications. Permissions and access to these protected items are managed through [Azure Active Directory](#).

Learn more:

- [What is Azure Key Vault?](#)
- [Get started with Azure Key Vault](#)
- [Azure Key Vault blog](#)

Virtual machine disk encryption

Azure Disk Encryption is a new capability for encrypting your Windows and Linux virtual machine disks. Azure Disk Encryption uses the industry-standard [BitLocker](#) feature of Windows and the [dm-crypt](#) feature of Linux to provide volume encryption for the OS and the data disks.

The solution is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets in your key vault subscription. It ensures that all data in the virtual machine disks are encrypted at rest in Azure Storage.

Learn more:

- [Azure Disk Encryption for IaaS VMs](#)
- [Quickstart: Encrypt a Windows IaaS VM with Azure PowerShell](#)

Virtual machine backup

Azure Backup is a scalable solution that helps protect your application data with zero capital investment and minimal operating costs. Application errors can corrupt your data, and human errors can introduce bugs into your applications. With Azure Backup, your virtual machines running Windows and Linux are protected.

Learn more:

- [What is Azure Backup?](#)
- [Azure Backup service FAQ](#)

Azure Site Recovery

An important part of your organization's BCDR strategy is figuring out how to keep corporate workloads and apps running when planned and unplanned outages occur. Azure Site Recovery helps orchestrate replication, failover, and recovery of workloads and apps so that they're available from a secondary location if your primary location goes down.

Site Recovery:

- **Simplifies your BCDR strategy:** Site Recovery makes it easy to handle replication, failover, and recovery of multiple business workloads and apps from a single location. Site Recovery orchestrates replication and failover but doesn't intercept your application data or have any information about it.
- **Provides flexible replication:** By using Site Recovery, you can replicate workloads running on Hyper-V virtual machines, VMware virtual machines, and Windows/Linux physical servers.
- **Supports failover and recovery:** Site Recovery provides test failovers to support disaster recovery drills without affecting production environments. You can also run planned failovers with a zero-data loss for expected outages, or unplanned failovers with minimal data loss (depending on replication frequency) for unexpected disasters. After failover, you can fail back to your primary sites. Site Recovery provides recovery plans that can include scripts and Azure automation workbooks so that you can customize failover and recovery of multi-tier applications.
- **Eliminates secondary datacenters:** You can replicate to a secondary on-premises site, or to Azure. Using Azure as a destination for disaster recovery eliminates the cost and complexity of maintaining a secondary site. Replicated data is stored in Azure Storage.
- **Integrates with existing BCDR technologies:** Site Recovery partners with other applications' BCDR features. For example, you can use Site Recovery to help protect the SQL Server back end of corporate workloads. This includes native support for SQL Server Always On to manage the failover of availability groups.

Learn more:

- [What is Azure Site Recovery?](#)
- [How does Azure Site Recovery work?](#)
- [What workloads are protected by Azure Site Recovery?](#)

Virtual networking

Virtual machines need network connectivity. To support that requirement, Azure requires virtual machines to be connected to an Azure virtual network.

An Azure virtual network is a logical construct built on top of the physical Azure network fabric. Each logical Azure virtual network is isolated from all other Azure virtual networks. This isolation helps insure that network traffic in your deployments is not accessible to other Microsoft Azure customers.

Learn more:

- [Azure network security overview](#)
- [Virtual Network overview](#)
- [Networking features and partnerships for enterprise scenarios](#)

Security policy management and reporting

Azure Security Center helps you prevent, detect, and respond to threats. Security Center gives you increased visibility into, and control over, the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions. It helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Security Center helps you optimize and monitor the security of your virtual machines by:

- Providing [security recommendations](#) for the virtual machines. Example recommendations include: apply system updates, configure ACLs endpoints, enable antimalware, enable network security groups, and apply disk encryption.
- Monitoring the state of your virtual machines.

Learn more:

- [Introduction to Azure Security Center](#)
- [Azure Security Center frequently asked questions](#)
- [Azure Security Center planning and operations](#)

Compliance

Azure Virtual Machines is certified for FISMA, FedRAMP, HIPAA, PCI DSS Level 1, and other key compliance programs. This certification makes it easier for your own Azure applications to meet compliance requirements and for your business to address a wide range of domestic and international regulatory requirements.

Learn more:

- [Microsoft Trust Center: Compliance](#)
- [Trusted Cloud: Microsoft Azure Security, Privacy, and Compliance](#)

Confidential Computing

While confidential computing is not technically part of virtual machine security, the topic of virtual machine security belongs to the higher-level subject of "compute" security. Confidential computing belongs within the category of "compute" security.

Confidential computing ensures that when data is "in the clear," which is required for efficient processing, the data is protected inside a Trusted Execution Environment https://en.wikipedia.org/wiki/Trusted_execution_environment (TEE - also known as an enclave), an example of which is shown in the figure below.

TEEs ensure there is no way to view data or the operations inside from the outside, even with a debugger. They

even ensure that only authorized code is permitted to access data. If the code is altered or tampered, the operations are denied and the environment disabled. The TEE enforces these protections throughout the execution of code within it.

Learn more:

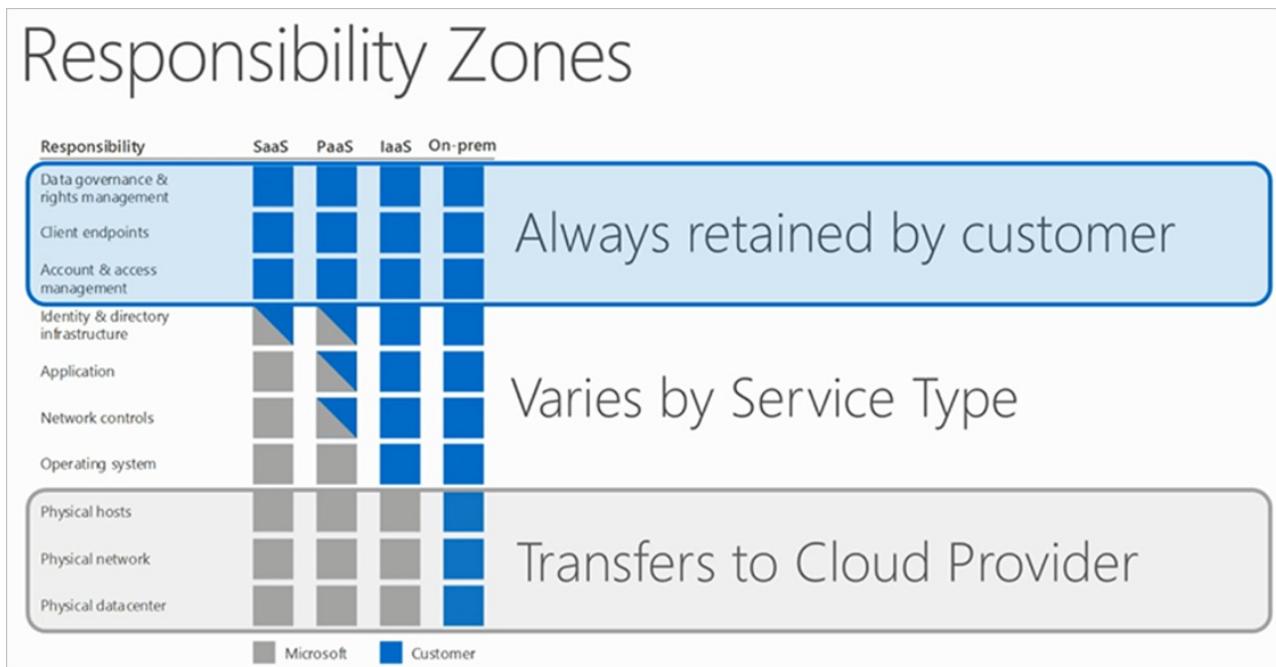
- [Introducing Azure confidential computing](#)
- [Azure confidential computing](#)

Security best practices for IaaS workloads in Azure

12/12/2018 • 11 minutes to read • [Edit Online](#)

In most infrastructure as a service (IaaS) scenarios, [Azure virtual machines \(VMs\)](#) are the main workload for organizations that use cloud computing. This fact is evident in [hybrid](#) scenarios where organizations want to slowly migrate workloads to the cloud. In such scenarios, follow the [general security considerations for IaaS](#), and apply security best practices to all your VMs.

Your responsibility for security is based on the type of cloud service. The following chart summarizes the balance of responsibility for both Microsoft and you:



Security requirements vary depending on a number of factors including different types of workloads. Not one of these best practices can by itself secure your systems. Like anything else in security, you have to choose the appropriate options and see how the solutions can complement each other by filling gaps.

This article describes security best practices for VMs and operating systems.

The best practices are based on a consensus of opinion, and they work with current Azure platform capabilities and feature sets. Because opinions and technologies can change over time, this article will be updated to reflect those changes.

Protect VMs by using authentication and access control

The first step in protecting your VMs is to ensure that only authorized users can set up new VMs and access VMs.

Best practice: Control VM access.

Detail: Use [Azure policies](#) to establish conventions for resources in your organization and create customized policies. Apply these policies to resources, such as [resource groups](#). VMs that belong to a resource group inherit its policies.

If your organization has many subscriptions, you might need a way to efficiently manage access, policies, and compliance for those subscriptions. [Azure management groups](#) provide a level of scope above subscriptions. You organize subscriptions into management groups (containers) and apply your governance conditions to those groups. All subscriptions within a management group automatically inherit the conditions applied to the group.

Management groups give you enterprise-grade management at a large scale no matter what type of subscriptions you might have.

Best practice: Reduce variability in your setup and deployment of VMs.

Detail: Use [Azure Resource Manager](#) templates to strengthen your deployment choices and make it easier to understand and inventory the VMs in your environment.

Best practice: Secure privileged access.

Detail: Use a [least privilege approach](#) and built-in Azure roles to enable users to access and set up VMs:

- [Virtual Machine Contributor](#): Can manage VMs, but not the virtual network or storage account to which they are connected.
- [Classic Virtual Machine Contributor](#): Can manage VMs created by using the classic deployment model, but not the virtual network or storage account to which the VMs are connected.
- [Security Admin](#): In Security Center only: Can view security policies, view security states, edit security policies, view alerts and recommendations, dismiss alerts and recommendations.
- [Dev/Test Labs User](#): Can view everything and connect, start, restart, and shut down VMs.

Your subscription admins and coadmins can change this setting, making them administrators of all the VMs in a subscription. Be sure that you trust all of your subscription admins and coadmins to log in to any of your machines.

NOTE

We recommend that you consolidate VMs with the same lifecycle into the same resource group. By using resource groups, you can deploy, monitor, and roll up billing costs for your resources.

Organizations that control VM access and setup improve their overall VM security.

Use multiple VMs for better availability

If your VM runs critical applications that need to have high availability, we strongly recommend that you use multiple VMs. For better availability, use an [availability set](#).

An availability set is a logical grouping that you can use in Azure to ensure that the VM resources you place within it are isolated from each other when they're deployed in an Azure datacenter. Azure ensures that the VMs you place in an availability set run across multiple physical servers, compute racks, storage units, and network switches. If a hardware or Azure software failure occurs, only a subset of your VMs are affected, and your overall application continues to be available to your customers. Availability sets are an essential capability when you want to build reliable cloud solutions.

Protect against malware

You should install antimalware protection to help identify and remove viruses, spyware, and other malicious software. You can install [Microsoft Antimalware](#) or a Microsoft partner's endpoint protection solution ([Trend Micro](#), [Symantec](#), [McAfee](#), [Windows Defender](#), and [System Center Endpoint Protection](#)).

Microsoft Antimalware includes features like real-time protection, scheduled scanning, malware remediation, signature updates, engine updates, samples reporting, and exclusion event collection. For environments that are hosted separately from your production environment, you can use an antimalware extension to help protect your VMs and cloud services.

You can integrate Microsoft Antimalware and partner solutions with [Azure Security Center](#) for ease of deployment and built-in detections (alerts and incidents).

Best practice: Install an antimalware solution to protect against malware.

Detail: Install a Microsoft partner solution or Microsoft Antimalware

Best practice: Integrate your antimalware solution with Security Center to monitor the status of your protection.

Detail: Manage endpoint protection issues with Security Center

Manage your VM updates

Azure VMs, like all on-premises VMs, are meant to be user managed. Azure doesn't push Windows updates to them. You need to manage your VM updates.

Best practice: Keep your VMs current.

Detail: Use the [Update Management](#) solution in Azure Automation to manage operating system updates for your Windows and Linux computers that are deployed in Azure, in on-premises environments, or in other cloud providers. You can quickly assess the status of available updates on all agent computers and manage the process of installing required updates for servers.

Computers that are managed by Update Management use the following configurations to perform assessment and update deployments:

- Microsoft Monitoring Agent (MMA) for Windows or Linux
- PowerShell Desired State Configuration (DSC) for Linux
- Automation Hybrid Runbook Worker
- Microsoft Update or Windows Server Update Services (WSUS) for Windows computers

If you use Windows Update, leave the automatic Windows Update setting enabled.

Best practice: Ensure at deployment that images you built include the most recent round of Windows updates.

Detail: Check for and install all Windows updates as a first step of every deployment. This measure is especially important to apply when you deploy images that come from either you or your own library. Although images from the Azure Marketplace are updated automatically by default, there can be a lag time (up to a few weeks) after a public release.

Best practice: Periodically redeploy your VMs to force a fresh version of the OS.

Detail: Define your VM with an [Azure Resource Manager template](#) so you can easily redeploy it. Using a template gives you a patched and secure VM when you need it.

Best practice: Install the latest security updates.

Detail: Some of the first workloads that customers move to Azure are labs and external-facing systems. If your Azure VMs host applications or services that need to be accessible to the internet, be vigilant about patching. Patch beyond the operating system. Unpatched vulnerabilities on partner applications can also lead to problems that can be avoided if good patch management is in place.

Best practice: Deploy and test a backup solution.

Detail: A backup needs to be handled the same way that you handle any other operation. This is true of systems that are part of your production environment extending to the cloud.

Test and dev systems must follow backup strategies that provide restore capabilities that are similar to what users have grown accustomed to, based on their experience with on-premises environments. Production workloads moved to Azure should integrate with existing backup solutions when possible. Or, you can use [Azure Backup](#) to help address your backup requirements.

Organizations that don't enforce software-update policies are more exposed to threats that exploit known, previously fixed vulnerabilities. To comply with industry regulations, companies must prove that they are diligent and using correct security controls to help ensure the security of their workloads located in the cloud.

Software-update best practices for a traditional datacenter and Azure IaaS have many similarities. We recommend that you evaluate your current software update policies to include VMs located in Azure.

Manage your VM security posture

Cyberthreats are evolving. Safeguarding your VMs requires a monitoring capability that can quickly detect threats, prevent unauthorized access to your resources, trigger alerts, and reduce false positives.

To monitor the security posture of your [Windows](#) and [Linux VMs](#), use [Azure Security Center](#). In Security Center, safeguard your VMs by taking advantage of the following capabilities:

- Apply OS security settings with recommended configuration rules.
- Identify and download system security and critical updates that might be missing.
- Deploy recommendations for endpoint antimalware protection.
- Validate disk encryption.
- Assess and remediate vulnerabilities.
- Detect threats.

Security Center can actively monitor for threats, and potential threats are exposed in security alerts. Correlated threats are aggregated in a single view called a security incident.

Security Center stores data in [Azure Log Analytics](#). Log Analytics provides a query language and analytics engine that gives you insights into the operation of your applications and resources. Data is also collected from [Azure Monitor](#), management solutions, and agents installed on virtual machines in the cloud or on-premises. This shared functionality helps you form a complete picture of your environment.

Organizations that don't enforce strong security for their VMs remain unaware of potential attempts by unauthorized users to circumvent security controls.

Monitor VM performance

Resource abuse can be a problem when VM processes consume more resources than they should. Performance issues with a VM can lead to service disruption, which violates the security principle of availability. This is particularly important for VMs that are hosting IIS or other web servers, because high CPU or memory usage might indicate a denial of service (DoS) attack. It's imperative to monitor VM access not only reactively while an issue is occurring, but also proactively against baseline performance as measured during normal operation.

We recommend that you use [Azure Monitor](#) to gain visibility into your resource's health. Azure Monitor features:

- [Resource diagnostic log files](#): Monitors your VM resources and identifies potential issues that might compromise performance and availability.
- [Azure Diagnostics extension](#): Provides monitoring and diagnostics capabilities on Windows VMs. You can enable these capabilities by including the extension as part of the [Azure Resource Manager template](#).

Organizations that don't monitor VM performance can't determine whether certain changes in performance patterns are normal or abnormal. A VM that's consuming more resources than normal might indicate an attack from an external resource or a compromised process running in the VM.

Encrypt your virtual hard disk files

We recommend that you encrypt your virtual hard disks (VHDs) to help protect your boot volume and data volumes at rest in storage, along with your encryption keys and secrets.

[Azure Disk Encryption](#) helps you encrypt your Windows and Linux IaaS virtual machine disks. Azure Disk Encryption uses the industry-standard [BitLocker](#) feature of Windows and the [DM-Crypt](#) feature of Linux to provide volume encryption for the OS and the data disks. The solution is integrated with [Azure Key Vault](#) to help you control and manage the disk-encryption keys and secrets in your key vault subscription. The solution also ensures that all data on the virtual machine disks are encrypted at rest in Azure Storage.

Following are best practices for using Azure Disk Encryption:

Best practice: Enable encryption on VMs.

Detail: Azure Disk Encryption generates and writes the encryption keys to your key vault. Managing encryption keys in your key vault requires Azure AD authentication. Create an Azure AD application for this purpose. For authentication purposes, you can use either client secret-based authentication or [client certificate-based Azure AD authentication](#).

Best practice: Use a key encryption key (KEK) for an additional layer of security for encryption keys. Add a KEK to your key vault.

Detail: Use the [Add-AzureKeyVaultKey](#) cmdlet to create a key encryption key in the key vault. You can also import a KEK from your on-premises hardware security module (HSM) for key management. For more information, see the [Key Vault documentation](#). When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault. Keeping an escrow copy of this key in an on-premises key management HSM offers additional protection against accidental deletion of keys.

Best practice: Take a [snapshot](#) and/or backup before disks are encrypted. Backups provide a recovery option if an unexpected failure happens during encryption.

Detail: VMs with managed disks require a backup before encryption occurs. After a backup is made, you can use the **Set-AzureRmVMDiskEncryptionExtension** cmdlet to encrypt managed disks by specifying the `-skipVmBackup` parameter. For more information about how to back up and restore encrypted VMs, see the [Azure Backup](#) article.

Best practice: To make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the key vault and the VMs to be located in the same region.

Detail: Create and use a key vault that is in the same region as the VM to be encrypted.

When you apply Azure Disk Encryption, you can satisfy the following business needs:

- IaaS VMs are secured at rest through industry-standard encryption technology to address organizational security and compliance requirements.
- IaaS VMs start under customer-controlled keys and policies, and you can audit their usage in your key vault.

Next steps

See [Azure security best practices and patterns](#) for more security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.

The following resources are available to provide more general information about Azure security and related Microsoft services:

- [Azure Security Team Blog](#) - for up to date information on the latest in Azure Security
- [Microsoft Security Response Center](#) - where Microsoft security vulnerabilities, including issues with Azure, can be reported or via email to secure@microsoft.com

Security Recommendations for Azure Marketplace Images

12/12/2018 • 3 minutes to read • [Edit Online](#)

We recommend that each solution comply with the following security configuration recommendations. This will help maintain a high level of security for partner solution images in the Azure Marketplace.

These recommendations can also be helpful for organizations that do not have images in the Azure marketplace. You may want to check your company's Windows and Linux image configurations against the guidelines found in the tables below.

Open Source-based Images

Category	Check
Security	All the latest security patches for the Linux distribution are installed.
Security	Industry guidelines to secure the VM image for the specific Linux distribution have been followed.
Security	Limit the attack surface by keeping minimal footprint with only necessary Windows Server roles, features, services and networking ports.
Security	Scan source code and resulting VM image for malware.
Security	The VHD image only includes necessary locked accounts, that do not have default passwords that would allow interactive login; no back doors.
Security	Firewall rules are disabled, unless application functionally relies on them, such as a firewall appliance.
Security	All sensitive information has been removed from the VHD image, such as test SSH keys, known hosts file, log files, and unnecessary certificates.
Security	It is recommended that LVM should not be used.
Security	Latest versions of required libraries should be included: - OpenSSL v1.0 or greater - Python 2.5 or above (Python 2.6+ is highly recommended) - Python pyasn1 package if not already installed - d.OpenSSL v 1.0 or greater
Security	Bash/Shell history entries must be cleared

Networking	SSH server should be included by default. Set SSH keep alive to sshd config with the following option: ClientAliveInterval 180
Networking	Image should not contain any custom network configuration. Delete the resolv.conf: <code>rm /etc/resolv.conf</code>
Deployment	<p>Latest Azure Linux Agent should be installed</p> <ul style="list-style-type: none"> - The agent should be installed using the RPM or Deb package. - You may also use the manual install process, but the installer packages are recommended and preferred. - If installing the agent manually from the GitHub repository, first copy the <code>waagent</code> file to <code>/usr/sbin</code> and run (as root): <pre># chmod 755 /usr/sbin/waagent # /usr/sbin/waagent -install</pre> The agent configuration file will be placed at <code>/etc/waagent.conf</code>.
Deployment	Ensures that Azure Support can provide our partners with serial console output when needed and provide adequate timeout for OS disk mounting from cloud storage. Image must have added the following parameters to the Kernel Boot Line: <pre>console=ttyS0 earlyprintk=ttyS0 rootdelay=300</pre>
Deployment	No swap partition on the OS disk. Swap can be requested for creation on the local resource disk by the Linux Agent.
Deployment	It is recommended that a single root partition is created for the OS disk.
Deployment	64-bit operating system only.

Windows Server-based Images

Category	Check
Security	Use a secure OS base image. The VHD used for the source of any image based on Windows Server must be from the Windows Server OS images provided through Microsoft Azure.
Security	Install all latest security updates.
Security	Applications should not have a dependency on restricted user names such as Administrator, root and admin.
Security	BitLocker Drive Encryption is not supported on the operating system hard disk. BitLocker may be used on data disks.
Security	Limit the attack surface by keeping minimal footprint with only necessary Windows Server roles, features, services and networking ports enabled.

Security	Scan source code and resulting VM image for malware.
Security	Set Windows Server images security update to auto-update.
Security	The VHD image only includes necessary locked accounts, that do not have default passwords that would allow interactive login; no back doors.
Security	Firewall rules are disabled, unless application functionally relies on them, such as a firewall appliance.
Deployment	64-bit operating system only.

Azure identity management security overview

11/7/2018 • 8 minutes to read • [Edit Online](#)

Identity management is the process of authenticating and authorizing [security principals](#). It also involves controlling information about those principals (identities). Security principals (identities) may include services, applications, users, groups, etc. Microsoft identity and access management solutions help IT protect access to applications and resources across the corporate datacenter and into the cloud. Such protection enables additional levels of validation, such as Multi-Factor Authentication and conditional access policies. Monitoring suspicious activity through advanced security reporting, auditing, and alerting helps mitigate potential security issues. [Azure Active Directory Premium](#) provides single sign-on (SSO) to thousands of cloud software as a service (SaaS) apps and access to web apps that you run on-premises.

By taking advantage of the security benefits of Azure Active Directory (Azure AD), you can:

- Create and manage a single identity for each user across your hybrid enterprise, keeping users, groups, and devices in sync.
- Provide SSO access to your applications, including thousands of pre-integrated SaaS apps.
- Enable application access security by enforcing rules-based Multi-Factor Authentication for both on-premises and cloud applications.
- Provision secure remote access to on-premises web applications through Azure AD Application Proxy.

The goal of this article is to provide an overview of the core Azure security features that help with identity management. We also provide links to articles that give details of each feature so you can learn more.

The article focuses on the following core Azure Identity management capabilities:

- Single sign-on
- Reverse proxy
- Multi-Factor Authentication
- Role based access control (RBAC)
- Security monitoring, alerts, and machine learning-based reports
- Consumer identity and access management
- Device registration
- Privileged identity management
- Identity protection
- Hybrid identity management/Azure AD connect
- Azure AD access reviews

Single sign-on

SSO means being able to access all the applications and resources that you need to do business, by signing in only once using a single user account. Once signed in, you can access all of the applications you need without being required to authenticate (for example, type a password) a second time.

Many organizations rely upon SaaS applications such as Office 365, Box, and Salesforce for user productivity. Historically, IT staff needed to individually create and update user accounts in each SaaS application, and users had to remember a password for each SaaS application.

Azure AD extends on-premises Active Directory environments into the cloud, enabling users to use their primary organizational account to sign in not only to their domain-joined devices and company resources, but also to all the

web and SaaS applications they need for their jobs.

Not only do users not have to manage multiple sets of usernames and passwords, you can provision or de-provision application access automatically, based on their organizational groups and their employee status. Azure AD introduces security and access governance controls with which you can centrally manage users' access across SaaS applications.

Learn more:

- [Overview of single sign-on](#)
- [What is application access and single sign-on with Azure Active Directory?](#)
- [Integrate Azure Active Directory single sign-on with SaaS apps](#)

Reverse proxy

Azure AD Application Proxy lets you publish on-premises applications, such as [SharePoint](#) sites, [Outlook Web App](#), and [IIS](#)-based apps inside your private network and provides secure access to users outside your network.

Application Proxy provides remote access and SSO for many types of on-premises web applications with the thousands of SaaS applications that Azure AD supports. Employees can sign in to your apps from home on their own devices and authenticate through this cloud-based proxy.

Learn more:

- [Enabling Azure AD Application Proxy](#)
- [Publish applications using Azure AD Application Proxy](#)
- [Single sign-on with Application Proxy](#)
- [Working with conditional access](#)

Multi-Factor Authentication

Azure Multi-Factor Authentication is a method of authentication that requires the use of more than one verification method and adds a critical second layer of security to user sign-ins and transactions. Multi-Factor Authentication helps safeguard access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication via a range of verification options: phone calls, text messages, or mobile app notifications or verification codes and third-party OAuth tokens.

Learn more:

- [Multi-Factor Authentication](#)
- [What is Azure Multi-Factor Authentication?](#)
- [How Azure Multi-Factor Authentication works](#)

RBAC

RBAC is an authorization system built on Azure Resource Manager that provides fine-grained access management of resources in Azure. RBAC allows you to granularly control the level of access that users have. For example, you can limit a user to only manage virtual networks and another user to manage all resources in a resource group. Azure includes several built-in roles that you can use. The following lists four fundamental built-in roles. The first three apply to all resource types.

Learn more:

- [What is role-based access control \(RBAC\)?](#)
- [Built-in roles for Azure resources](#)

Security monitoring, alerts, and machine learning-based reports

Security monitoring, alerts, and machine learning-based reports that identify inconsistent access patterns can help you protect your business. You can use Azure AD access and usage reports to gain visibility into the integrity and security of your organization's directory. With this information, a directory administrator can better determine where possible security risks might lie so that they can adequately plan to mitigate those risks.

In the Azure portal, reports fall into the following categories:

- **Anomaly reports:** Contain sign-in events that we found to be anomalous. Our goal is to make you aware of such activity and enable you to determine whether an event is suspicious.
- **Integrated Application reports:** Provide insights into how cloud applications are being used in your organization. Azure AD offers integration with thousands of cloud applications.
- **Error reports:** Indicate errors that might occur when you provision accounts to external applications.
- **User-specific reports:** Display device sign-in activity data for a specific user.
- **Activity logs:** Contain a record of all audited events within the last 24 hours, last 7 days, or last 30 days, and group activity changes and password reset and registration activity.

Learn more:

- [View your access and usage reports](#)
- [Get started with Azure Active Directory reporting](#)
- [Azure Active Directory reporting guide](#)

Consumer identity and access management

Azure AD B2C is a highly available, global, identity management service for consumer-facing applications that scales to hundreds of millions of identities. It can be integrated across mobile and web platforms. Your consumers can sign in to all your applications through customizable experiences by using their existing social accounts or by creating new credentials.

In the past, application developers who wanted to sign up customers and sign them in to their applications would have written their own code. And they would have used on-premises databases or systems to store usernames and passwords. Azure AD B2C offers your organization a better way to integrate consumer identity management into applications with the help of a secure, standards-based platform and a large set of extensible policies.

When you use Azure AD B2C, your consumers can sign up for your applications by using their existing social accounts (Facebook, Google, Amazon, LinkedIn) or by creating new credentials (email address and password, or username and password).

Learn more:

- [What is Azure Active Directory B2C?](#)
- [Azure Active Directory B2C preview: Sign up and sign in consumers in your applications](#)
- [Azure Active Directory B2C Preview: Types of applications](#)

Device registration

Azure AD device registration is the foundation for device-based [conditional access](#) scenarios. When a device is registered, Azure AD device registration provides the device with an identity that it uses to authenticate the device when a user signs in. The authenticated device and the attributes of the device can then be used to enforce conditional access policies for applications that are hosted in the cloud and on-premises.

When combined with a mobile device management solution such as Intune, the device attributes in Azure AD are updated with additional information about the device. You can then create conditional access rules that enforce

access from devices to meet your standards for security and compliance.

Learn more:

- [Get started with Azure AD device registration](#)
- [Automatic device registration with Azure AD for Windows domain-joined devices](#)
- [Set up automatic registration of Windows domain-joined devices with Azure AD](#)

Privileged identity management

With Azure AD Privileged Identity Management, you can manage, control, and monitor your privileged identities and access to resources in Azure AD as well as other Microsoft online services, such as Office 365 and Microsoft Intune.

Users sometimes need to carry out privileged operations in Azure or Office 365 resources, or in other SaaS apps. This need often means that organizations have to give users permanent privileged access in Azure AD. Such access is a growing security risk for cloud-hosted resources, because organizations can't sufficiently monitor what the users are doing with their administrator privileges. Additionally, if a user account with privileged access is compromised, that one breach could affect the organization's overall cloud security. Azure AD Privileged Identity Management helps to mitigate this risk.

With Azure AD Privileged Identity Management, you can:

- See which users are Azure AD administrators.
- Enable on-demand, just-in-time (JIT) administrative access to Microsoft services such as Office 365 and Intune.
- Get reports about administrator access history and changes in administrator assignments.
- Get alerts about access to a privileged role.

Learn more:

- [Azure AD Privileged Identity Management](#)
- [Roles in Azure AD Privileged Identity Management](#)
- [Azure AD Privileged Identity Management: How to add or remove a user role](#)

Identity protection

Azure AD Identity Protection is a security service that provides a consolidated view into risk events and potential vulnerabilities that affect your organization's identities. Identity Protection takes advantage of existing Azure AD anomaly-detection capabilities, which are available through Azure AD Anomalous Activity reports. Identity Protection also introduces new risk event types that can detect anomalies in real time.

Learn more:

- [Azure AD Identity Protection](#)
- [Channel 9: Azure AD and Identity Show: Identity Protection Preview](#)

Hybrid identity management/Azure AD connect

Microsoft's identity solutions span on-premises and cloud-based capabilities, creating a single user identity for authentication and authorization to all resources, regardless of location. We call this hybrid identity. Azure AD Connect is the Microsoft tool designed to meet and accomplish your hybrid identity goals. This allows you to provide a common identity for your users for Office 365, Azure, and SaaS applications integrated with Azure AD. It provides the following features:

- Synchronization
- AD FS and federation integration

- Pass through authentication
- Health Monitoring

Learn more:

- [Hybrid identity white paper](#)
- [Azure Active Directory](#)
- [Azure AD team blog](#)

Azure AD access reviews

Azure Active Directory (Azure AD) access reviews enable organizations to efficiently manage group memberships, access to enterprise applications, and privileged role assignments.

Learn more:

- [Azure AD access reviews](#)
- [Manage user access with Azure AD access reviews](#)

Choose the right authentication method for your Azure Active Directory hybrid identity solution

11/7/2018 • 15 minutes to read • [Edit Online](#)

This article begins a series of articles that help organizations implement a complete Azure Active Directory (Azure AD) hybrid identity solution. This solution was outlined as the [Hybrid Identity Digital Transformation Framework](#). It covers the business outcomes and goals organizations can focus on to implement a robust and secure hybrid identity solution.

The first business outcome of the framework spells out the requirements for organizations to secure the authentication process when users access cloud apps. The first business goal in the authentication secured business outcome is users' ability to sign in to cloud apps by using their on-premises usernames and passwords. This sign-in process to and how users authenticate make everything in the cloud possible.

Choosing the correct authentication method is the first concern for organizations wanting to move their apps to the cloud. Don't take this decision lightly, for the following reasons:

1. It's the first decision for an organization that wants to move to the cloud.
2. The authentication method is a critical component of an organization's presence in the cloud. It controls access to all cloud data and resources.
3. It's the foundation of all the other advanced security and user experience features in Azure AD.
4. The authentication method is difficult to change after it's implemented.

Identity is the new control plane of IT security. So authentication is an organization's access guard to the new cloud world. Organizations need an identity control plane that strengthens their security and keeps their cloud apps safe from intruders.

Out of scope

Organizations that don't have an existing on-premises directory footprint aren't the focus of this article. Typically, those businesses create identities only in the cloud, which doesn't require a hybrid identity solution. Cloud-only identities exist solely in the cloud and aren't associated with corresponding on-premises identities.

Authentication methods

When the Azure AD hybrid identity solution is your new control plane, authentication is the foundation of cloud access. Choosing the correct authentication method is a crucial first decision in setting up an Azure AD hybrid identity solution. Implement the authentication method that is configured by using Azure AD Connect, which also provisions users in the cloud.

To choose an authentication method, you need to consider the time, existing infrastructure, complexity, and cost of implementing your choice. These factors are different for every organization and might change over time.

Azure AD supports the following authentication methods for hybrid identity solutions.

Cloud authentication

When you choose this authentication method, Azure AD handles users' sign-in process. Coupled with seamless single sign-on (SSO), users can sign in to cloud apps without having to reenter their credentials. With cloud authentication, you can choose from two options:

Azure AD password hash synchronization. The simplest way to enable authentication for on-premises directory objects in Azure AD. Users can use the same username and password that they use on-premises without having to deploy any additional infrastructure. Some premium features of Azure AD, like Identity Protection, require password hash synchronization for no matter which authentication method you choose.

NOTE

Passwords are never stored in clear text or encrypted with a reversible algorithm in Azure AD. For more information on the actual process of password hash synchronization, see [Implement password hash synchronization with Azure AD Connect sync](#).

Azure AD Pass-through Authentication. Provides a simple password validation for Azure AD authentication services by using a software agent that runs on one or more on-premises servers. The servers validate the users directly with your on-premises Active Directory, which ensures that the password validation doesn't happen in the cloud.

Companies with a security requirement to immediately enforce on-premises user account states, password policies, and sign-in hours might use this authentication method. For more information on the actual pass-through authentication process, see [User sign-in with Azure AD pass-through authentication](#).

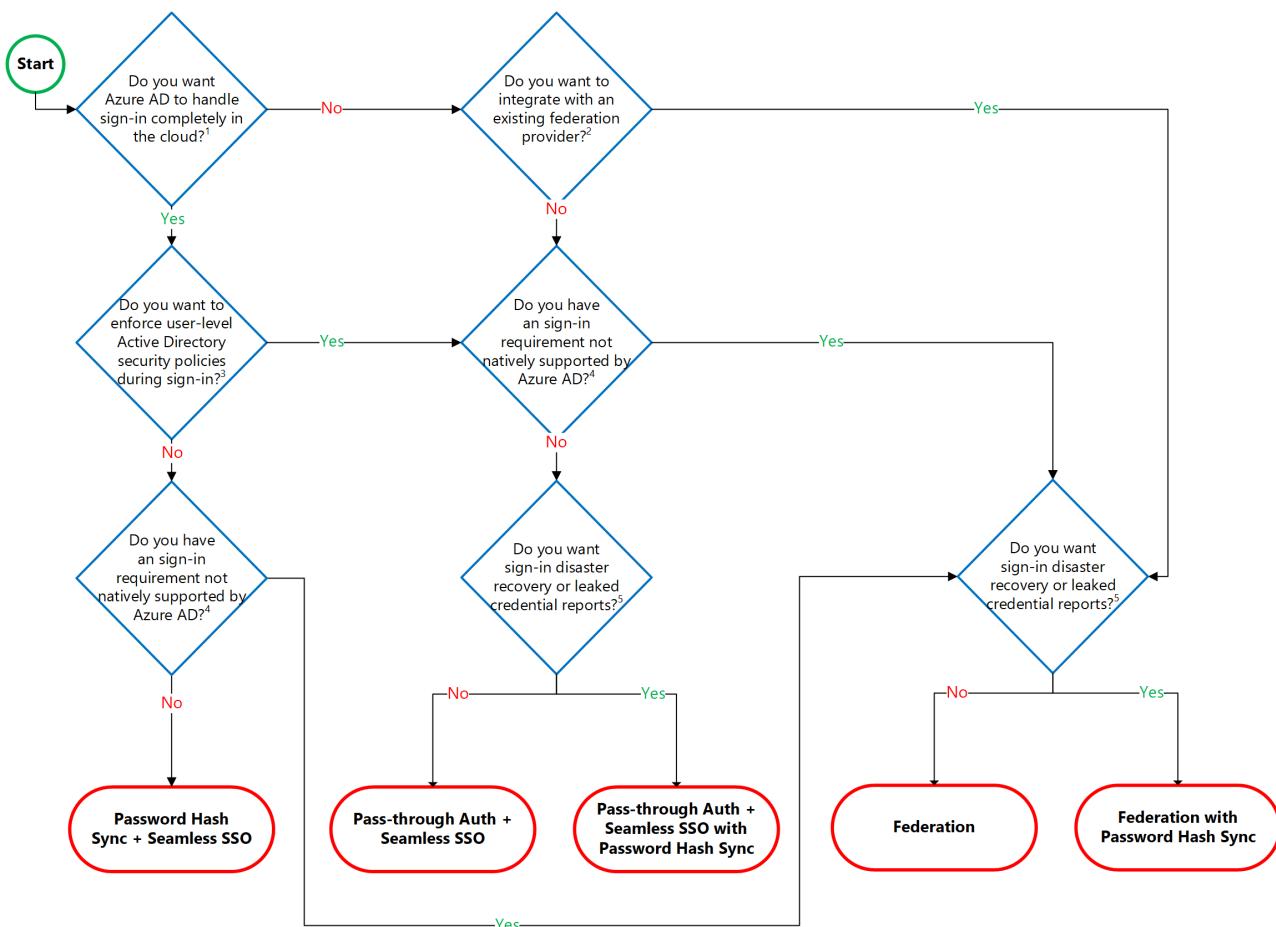
Federated authentication

When you choose this authentication method, Azure AD hands off the authentication process to a separate trusted authentication system, such as on-premises Active Directory Federation Services (AD FS), to validate the user's password.

The authentication system can provide additional advanced authentication requirements. Examples are smartcard-based authentication or third-party multifactor authentication. For more information, see [Deploying Active Directory Federation Services](#).

The following section helps you decide which authentication method is right for you by using a decision tree. It helps you determine whether to deploy cloud or federated authentication for your Azure AD hybrid identity solution.

Decision tree



Details on decision questions:

1. Azure AD can handle sign-in for users without relying on on-premises components to verify passwords.
2. Azure AD can hand off user sign-in to a trusted authentication provider such as Microsoft's AD FS.
3. If you need to apply user-level Active Directory security policies such as account expired, disabled account, password expired, account locked out, and sign-in hours on each user sign-in, Azure AD requires some on-premises components.
4. Sign-in features not natively supported by Azure AD:
 - Sign-in using smartcards or certificates.
 - Sign-in using on-premises MFA Server.
 - Sign-in using 3rd party authentication solution.
 - Multi-site on-premises authentication solution.
5. Azure AD Identity Protection requires Password Hash Sync regardless of which sign-in method you choose, to provide the "Users with leaked credentials" report. Organizations can failover to Password Hash Sync if their primary sign-in method fails and it was configured before the failure event.

Detailed considerations

Cloud authentication: Password hash synchronization

- **Effort.** Password hash synchronization requires the least effort regarding deployment, maintenance, and infrastructure. This level of effort typically applies to organizations that only need their users to sign in to Office 365, SaaS apps, and other Azure AD-based resources. When turned on, password hash synchronization is part of the Azure AD Connect sync process and runs every two minutes.
- **User experience.** To improve users' sign-in experience, deploy seamless SSO with password hash synchronization. Seamless SSO eliminates unnecessary prompts when users are signed in.
- **Advanced scenarios.** If organizations choose to, it's possible to use insights from identities with Azure AD

Identity Protection reports. An example is as the leaked credentials report. Windows Hello for Business is another option that has [specific requirements when you use password hash synchronization](#).

Organizations that require multifactor authentication with password hash synchronization must use Azure AD multifactor authentication. Those organizations can't use third-party or on-premises multifactor authentication methods.

- **Business continuity.** Using password hash synchronization with cloud authentication is highly available as a cloud service that scales to all Microsoft datacenters. To make sure password hash synchronization does not go down for extended periods, deploy a second Azure AD Connect server in staging mode in a standby configuration.
- **Considerations.** Currently, password hash synchronization doesn't immediately enforce changes in on-premises account states. In this situation, a user has access to cloud apps until the user account state is synchronized to Azure AD. Organizations might want to overcome this limitation by running a new synchronization cycle after administrators do bulk updates to on-premises user account states. An example is disabling accounts.

NOTE

The password expired and account locked-out states aren't currently synced to Azure AD with Azure AD Connect.

Refer to [implementing password hash synchronization](#) for deployment steps.

Cloud authentication: Pass-through Authentication

- **Effort.** For pass-through authentication, you need one or more (we recommend three) lightweight agents installed on existing servers. These agents must have access to your on-premises Active Directory Domain Services, including your on-premises AD domain controllers. They need outbound access to the Internet and access to your domain controllers. For this reason, it's not supported to deploy the agents in a perimeter network.

Pass-through Authentication requires unconstrained network access to domain controllers. All network traffic is encrypted and limited to authentication requests. For more information on this process, see the [security deep dive](#) on pass-through authentication.

- **User experience.** To improve users' sign-in experience, deploy seamless SSO with Pass-through Authentication. Seamless SSO eliminates unnecessary prompts after users sign in.
- **Advanced scenarios.** Pass-through Authentication enforces the on-premises account policy at the time of sign in. For example, access is denied when an on-premises user's account state is disabled, locked out, or [password expired](#) or falls outside the hours when the user is allowed to sign in.

Organizations that require multifactor authentication with pass-through authentication must use Azure Multi-Factor Authentication (MFA). Those organizations can't use a third-party or on-premises multifactor authentication method. Advanced features require that password hash synchronization is deployed whether or not you choose pass-through authentication. An example is the leaked credentials report of Identity Protection.

- **Business continuity.** We recommend that you deploy two extra pass-through authentication agents. These extras are in addition to the first agent on the Azure AD Connect server. This additional deployment ensures high availability of authentication requests. When you have three agents deployed, one agent can still fail when another agent is down for maintenance.

There's another benefit to deploying password hash synchronization in addition to pass-through authentication. It acts as a backup authentication method when the primary authentication method is no longer available.

- **Considerations.** You can use password hash synchronization as a backup authentication method for pass-through authentication, when the agents can't validate a user's credentials due to a significant on-premises failure. Failover to password hash synchronization doesn't happen automatically and you must use Azure AD Connect to switch the sign-on method manually.

For other considerations on Pass-through Authentication, including Alternate ID support, see [frequently asked questions](#).

Refer to [implementing pass-through authentication](#) for deployment steps.

Federated authentication

- **Effort.** A federated authentication system relies on an external trusted system to authenticate users. Some companies want to reuse their existing federated system investment with their Azure AD hybrid identity solution. The maintenance and management of the federated system falls outside the control of Azure AD. It's up to the organization by using the federated system to make sure it's deployed securely and can handle the authentication load.
- **User experience.** The user experience of federated authentication depends on the implementation of the features, topology, and configuration of the federation farm. Some organizations need this flexibility to adapt and configure the access to the federation farm to suit their security requirements. For example, it's possible to configure internally connected users and devices to sign in users automatically, without prompting them for credentials. This configuration works because they already signed in to their devices. If necessary, some advanced security features make users' sign-in process more difficult.
- **Advanced scenarios.** A federated authentication solution is usually required when customers have an authentication requirement that Azure AD doesn't support natively. See detailed information to help you [choose the right sign-in option](#). Consider the following common requirements:
 - Authentication that requires smartcards or certificates.
 - On-premises MFA servers or third-party multifactor providers.
 - Authentication by using third-party authentication solutions. See the [Azure AD federation compatibility list](#).
 - Sign in that requires an sAMAccountName, for example, DOMAIN\username, instead of a User Principal Name (UPN), for example, user@domain.com.
- **Business continuity.** Federated systems typically require a load-balanced array of servers, known as a farm. This farm is configured in an internal network and perimeter network topology to ensure high availability for authentication requests.

Deploy password hash synchronization along with federated authentication as a backup authentication method when the primary authentication method is no longer available. An example is when the on-premises servers aren't available. Some large enterprise organizations require a federation solution to support multiple Internet ingress points configured with geo-DNS for low-latency authentication requests.

- **Considerations.** Federated systems typically require a more significant investment in on-premises infrastructure. Most organizations choose this option if they already have an on-premises federation investment. And if it's a strong business requirement to use a single-identity provider. Federation is more complex to operate and troubleshoot compared to cloud authentication solutions.

For a nonroutable domain that can't be verified in Azure AD, you need extra configuration to implement user ID sign in. This requirement is known as Alternate login ID support. See [Configuring Alternate Login ID](#) for limitations and requirements. If you choose to use a third-party multi-factor authentication provider with federation, ensure the provider supports WS-Trust to allow devices to join Azure AD.

Refer to [Deploying Federation Servers](#) for deployment steps.

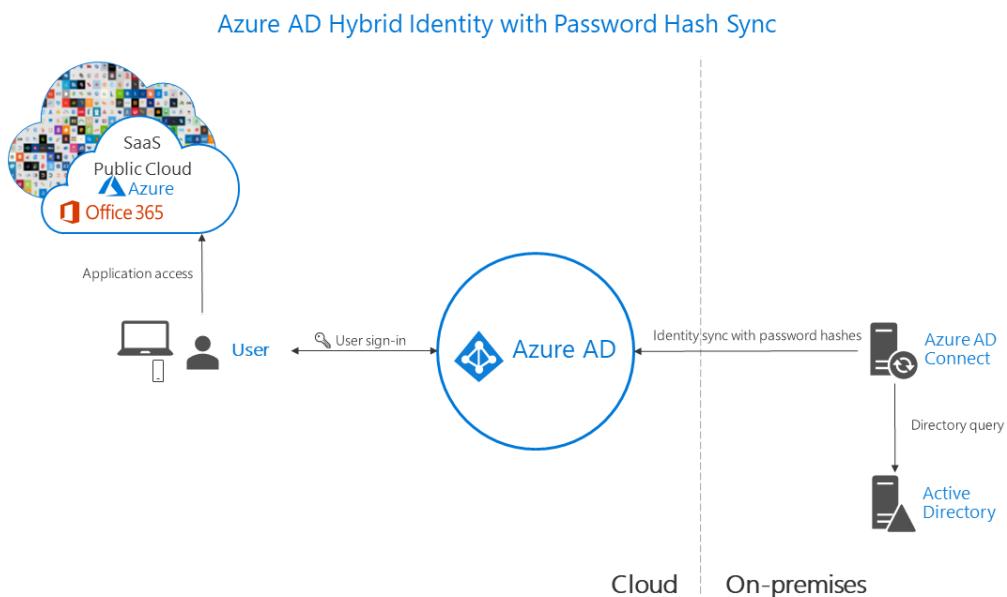
NOTE

When you deploy your Azure AD hybrid identity solution, you must implement one of the supported topologies of Azure AD Connect. Learn more about supported and unsupported configurations at [Topologies for Azure AD Connect](#).

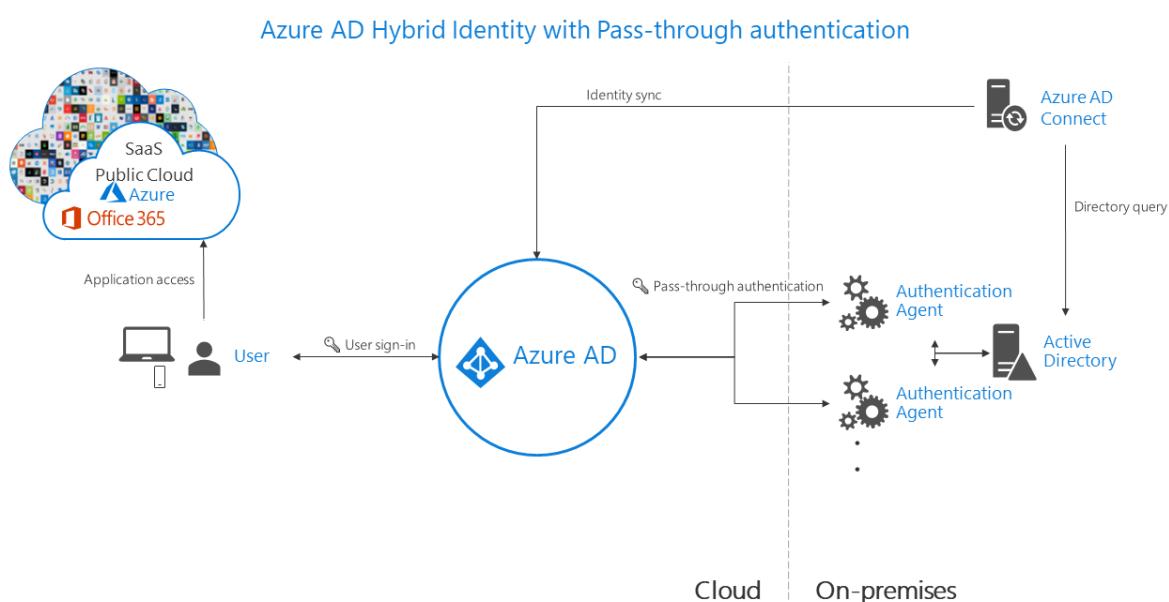
Architecture diagrams

The following diagrams outline the high-level architecture components required for each authentication method you can use with your Azure AD hybrid identity solution. They provide an overview to help you compare the differences between the solutions.

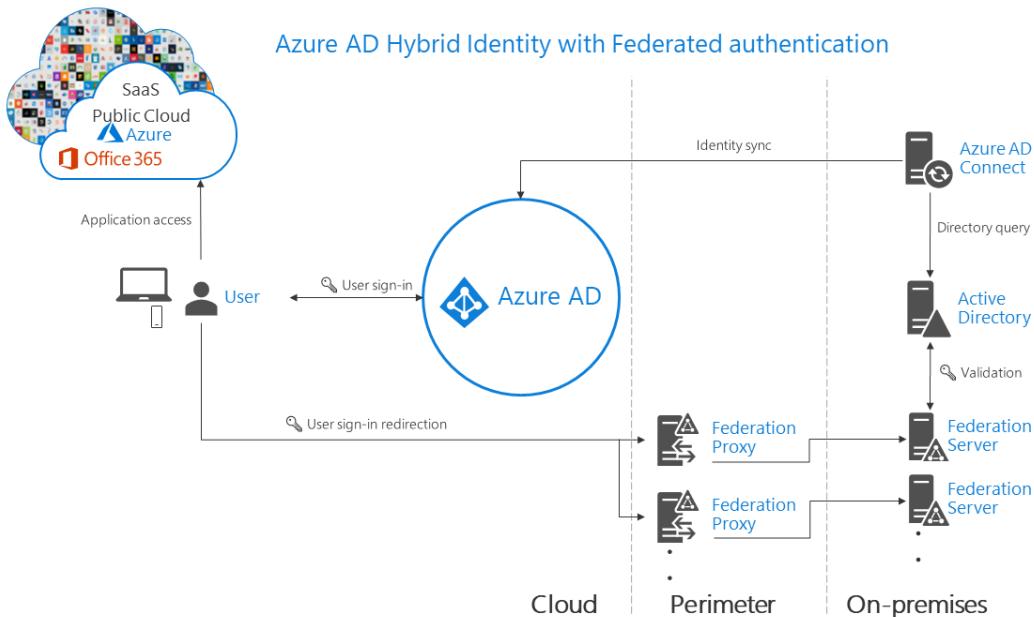
- Simplicity of a password hash synchronization solution:



- Agent requirements of pass-through authentication:



- Components required for federation in your perimeter and internal network of your organization:



Comparing methods

CONSIDERATION	PASSWORD HASH SYNCHRONIZATION + SEAMLESS SSO	PASS-THROUGH AUTHENTICATION + SEAMLESS SSO	FEDERATION WITH AD FS
Where does authentication happen?	In the cloud	In the cloud after a secure password verification exchange with the on-premises authentication agent	On-premises
What are the on-premises server requirements beyond the provisioning system: Azure AD Connect?	None	One server for each additional authentication agent	Two or more AD FS servers Two or more WAP servers in the perimeter/DMZ network
What are the requirements for on-premises Internet and networking beyond the provisioning system?	None	Outbound Internet access from the servers running authentication agents	Inbound Internet access to WAP servers in the perimeter Inbound network access to AD FS servers from WAP servers in the perimeter Network load balancing
Is there an SSL certificate requirement?	No	No	Yes
Is there a health monitoring solution?	Not required	Agent status provided by Azure Active Directory admin center	Azure AD Connect Health
Do users get single sign-on to cloud resources from domain-joined devices within the company network?	Yes with Seamless SSO	Yes with Seamless SSO	Yes

CONSIDERATION	PASSWORD HASH SYNCHRONIZATION + SEAMLESS SSO	PASS-THROUGH AUTHENTICATION + SEAMLESS SSO	FEDERATION WITH AD FS
What sign-in types are supported?	<p>UserPrincipalName + password</p> <p>Windows Integrated Authentication by using Seamless SSO</p> <p>Alternate login ID</p>	<p>UserPrincipalName + password</p> <p>Windows Integrated Authentication by using Seamless SSO</p> <p>Alternate login ID</p>	<p>UserPrincipalName + password</p> <p>sAMAccountName + password</p> <p>Windows Integrated Authentication</p> <p>Certificate and smart card authentication</p> <p>Alternate login ID</p>
Is Windows Hello for Business supported?	<p>Key trust model</p> <p>Certificate trust model with Intune</p>	<p>Key trust model</p> <p>Certificate trust model with Intune</p>	<p>Key trust model</p> <p>Certificate trust model</p>
What are the multifactor authentication options?	<p>Azure MFA</p> <p>Custom Controls with conditional access*</p>	<p>Azure MFA</p> <p>Custom Controls with conditional access*</p>	<p>Azure MFA</p> <p>Azure MFA server</p> <p>Third-party MFA</p> <p>Custom Controls with conditional access*</p>
What user account states are supported?	<p>Disabled accounts (up to 30-minute delay)</p>	<p>Disabled accounts</p> <p>Account locked out</p> <p>Account expired</p> <p>Password expired</p> <p>Sign-in hours</p>	<p>Disabled accounts</p> <p>Account locked out</p> <p>Account expired</p> <p>Password expired</p> <p>Sign-in hours</p>
What are the conditional access options?	Azure AD conditional access	<p>Azure AD conditional access</p>	<p>Azure AD conditional access</p> <p>AD FS claim rules</p>
Is blocking legacy protocols supported?	Yes	Yes	Yes
Can you customize the logo, image, and description on the sign-in pages?	Yes, with Azure AD Premium	Yes, with Azure AD Premium	Yes
What advanced scenarios are supported?	<p>Smart password lockout</p> <p>Leaked credentials reports</p>	<p>Smart password lockout</p>	<p>Multisite low-latency authentication system</p> <p>AD FS extranet lockout</p> <p>Integration with third-party identity systems</p>

NOTE

Custom controls in Azure AD conditional access does not currently support device registration.

Recommendations

Your identity system ensures your users' access to cloud apps and the line-of-business apps that you migrate and make available in the cloud. To keep authorized users productive and bad actors out of your organization's sensitive data, authentication controls access to apps.

Use or enable password hash synchronization for whichever authentication method you choose, for the following reasons:

1. **High availability and disaster recovery.** Pass-through Authentication and federation rely on on-premises infrastructure. For pass-through authentication, the on-premises footprint includes the server hardware and networking the Pass-through Authentication agents require. For federation, the on-premises footprint is even larger. It requires servers in your perimeter network to proxy authentication requests and the internal federation servers.

To avoid single points of failures, deploy redundant servers. Then authentication requests will always be serviced if any component fails. Both pass-through authentication and federation also rely on domain controllers to respond to authentication requests, which can also fail. Many of these components need maintenance to stay healthy. Outages are more likely when maintenance isn't planned and implemented correctly. Avoid outages by using password hash synchronization because the Microsoft Azure AD cloud authentication service scales globally and is always available.

2. **On-premises outage survival.** The consequences of an on-premises outage due to a cyber-attack or disaster can be substantial, ranging from reputational brand damage to a paralyzed organization unable to deal with the attack. Recently, many organizations were victims of malware attacks, including targeted ransomware, that caused their on-premises servers to go down. When Microsoft helps customers deal with these kinds of attacks, it sees two categories of organizations:

- Organizations that previously turned on password hash synchronization changed their authentication method to use password hash synchronization. They were back online in a matter of hours. By using access to email via Office 365, they worked to resolve issues and access other cloud-based workloads.
- Organizations that didn't previously enable password hash synchronization had to resort to untrusted external consumer email systems for communications and resolving issues. In those cases, it took them weeks or more to be up and running again.

3. **Identity protection.** One of the best ways to protect users in the cloud is Azure AD Identity Protection. Microsoft continually scans the Internet for user and password lists that bad actors sell and make available on the dark web. Azure AD can use this information to verify if any of the usernames and passwords in your organization are compromised. So it's critical to enable password hash synchronization no matter what authentication method you use, whether that's federated or pass-through authentication. Leaked credentials are presented as a report. Use this information to block or force users to change their passwords when they try to sign in with leaked passwords.

Lastly, according to [Gartner](#), Microsoft has the most full-featured set of identity and access management functions. Microsoft handles [450 billion authentication requests](#) every month to provide access to thousands of SaaS applications like Office 365 from virtually any device.

Conclusion

This article outlines various authentication options that organizations can configure and deploy to support access

to cloud apps. To meet various business, security, and technical requirements, organizations can choose between password hash synchronization, Pass-through Authentication, and federation.

Consider each authentication method. Does the effort to deploy the solution, and the user's experience of the sign-in process, address your business requirements? Evaluate whether your organization needs the advanced scenarios and business continuity features of each authentication method. Finally, evaluate the considerations of each authentication method. Do any of them prevent you from implementing your choice?

Next steps

In today's world, threats are present 24 hours a day and come from everywhere. Implement the correct authentication method, and it will mitigate your security risks and protect your identities.

[Get started](#) with Azure AD and deploy the right authentication solution for your organization.

If you're thinking about migrating from federated to cloud authentication, learn more about [changing the sign-in method](#). To help you plan and implement the migration, use [these project deployment plans](#).

Five steps to securing your identity infrastructure

11/19/2018 • 12 minutes to read • [Edit Online](#)

If you're reading this document, you're aware of the significance of security. You likely already carry the responsibility for securing your organization. If you need to convince others of the importance of security, send them to read the latest [Microsoft Security Intelligence report](#).

This document will help you get a more secure posture using the capabilities of Azure Active Directory by using a five-step checklist to inoculate your organization against cyber-attacks.

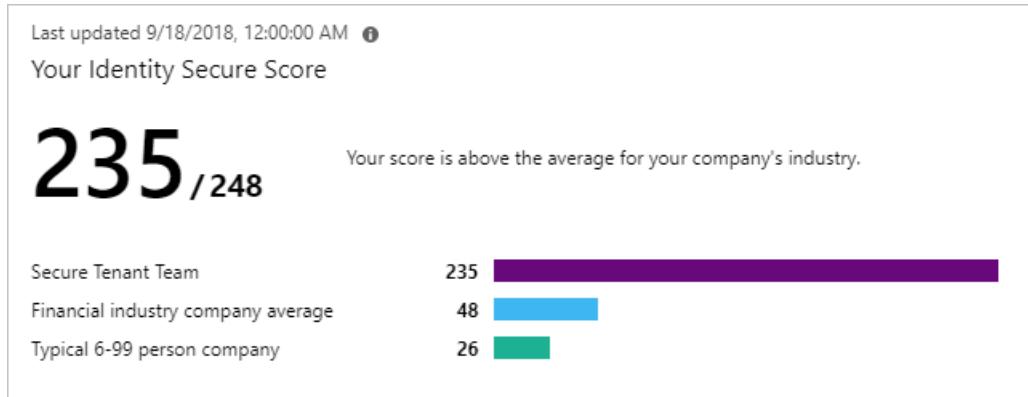
This checklist will help you quickly deploy critical recommended actions to protect your organization immediately by explaining how to:

- Strengthen your credentials.
- Reduce your attack surface area.
- Automate threat response.
- Increase your awareness of auditing and monitoring.
- Enable more predictable and complete end-user security with self-help.

NOTE

Many of the recommendations in this document apply only to applications that are configured to use Azure Active Directory as their identity provider. Configuring apps for Single Sign-On assures the benefits of credential policies, threat detection, auditing, logging, and other features add to those applications. [Single sign-on through Azure Active Directory](#) is the foundation - on which all these recommendations are based.

The recommendations in this document are aligned with the [Identity Secure Score](#), an automated assessment of your Azure AD tenant's identity security configuration. Organizations can use the Identity Secure Score page in the Azure AD portal to find gaps in their current security configuration to ensure they follow current Microsoft best practices for security. Implementing each recommendation in the Secure Score page will increase your score and allow you to track your progress, plus help you compare your implementation against other similar size organizations or your industry.



Before you begin: Protect privileged accounts with MFA

Before you begin this checklist, make sure you don't get compromised while you're reading this checklist. You first need to protect your privileged accounts.

Attackers who get control of privileged accounts can do tremendous damage, so it's critical to protect these

accounts first. Enable and require [Azure Multi-Factor Authentication \(MFA\)](#) for all administrators in your organization using [Baseline Protection](#). If you haven't implemented MFA, do it now! It's that important.

All set? Let's get started on the checklist.

Step 1 - Strengthen your credentials

Most enterprise security breaches originate with an account compromised with one of a handful of methods such as password spray, breach replay, or phishing. Learn more about these attacks in this video (45 min):

Make sure your organization use strong authentication

Given the frequency of passwords being guessed, phished, stolen with malware, or reused, it's critical to back the password with some form of strong credential – learn more about [Azure Multi-Factor Authentication](#).

Start banning commonly attacked passwords and turn off traditional complexity, and expiration rules.

Many organizations use the traditional complexity (requiring special characters, numbers, uppercase, and lowercase) and password expiration rules. [Microsoft's research](#) has shown these policies cause users to choose passwords that are easier to guess.

Azure AD's [dynamic banned password](#) feature uses current attacker behavior to prevent users from setting passwords that can easily be guessed. This capability is always on when users are created in the cloud, but is now also available for hybrid organizations when they deploy [Azure AD password protection for Windows Server Active Directory](#). Azure AD password protection blocks users from choosing these common passwords and can be extended to block password containing custom keywords you specify. For example, you can prevent your users from choosing passwords containing your company's product names or a local sport team.

Microsoft recommends adopting the following modern password policy based on [NIST guidance](#):

1. Require passwords have at least 8 characters. Longer isn't necessarily better, as they cause users to choose predictable passwords, save passwords in files, or write them down.
2. Disable expiration rules, which drive users to easily guessed passwords such as **Summer2018!**
3. Disable character-composition requirements and prevent users from choosing commonly attacked passwords, as they cause users to choose predictable character substitutions in passwords.

You can use [PowerShell to prevent passwords from expiring](#) for users if you create identities in Azure AD directly. Hybrid organizations should implement these policies using [domain group policy settings](#) or [Windows PowerShell](#).

Protect against leaked credentials and add resilience against outages

If your organization uses a hybrid identity solution with pass-through authentication or federation, then you should enable password hash sync for the following two reasons:

- The [Users with leaked credentials](#) report in the Azure AD management warns you of username and password pairs, which have been exposed on the "dark web." An incredible volume of passwords is leaked via phishing, malware, and password reuse on third-party sites that are later breached. Microsoft finds many of these leaked credentials and will tell you, in this report, if they match credentials in your organization – but only if you [enable password hash sync](#)!
- In the event of an on-premises outage (for example, in a ransomware attack) you'll be able to switch over to using [cloud authentication using password hash sync](#). This backup authentication method will allow you to continue accessing apps configured for authentication with Azure Active Directory, including Office 365. In this case IT staff will not need to resort to personal email accounts to share data until the on-premises outage is resolved.

Learn more about how [password hash sync](#) works.

NOTE

If you enable password hash sync and are using Azure AD Domain services, Kerberos (AES 256) hashes and optionally NTLM (RC4, no salt) hashes will also be encrypted and synchronized to Azure AD.

Implement AD FS extranet smart lockout

Organizations, which configure applications to authenticate directly to Azure AD benefit from [Azure AD smart lockout](#). If you use AD FS in Windows Server 2012R2, implement AD FS [extranet lockout protection](#). If you use AD FS on Windows Server 2016, implement [extranet smart lockout](#). AD FS Smart Extranet lockout protects against brute force attacks, which target AD FS while preventing users from being locked out in Active Directory.

Take advantage of intrinsically secure, easier to use credentials

Using [Windows Hello](#), you can replace passwords with strong two-factor authentication on PCs and mobile devices. This authentication consists of a new type of user credential that is tied securely to a device and uses a biometric or PIN.

Step 2 - Reduce your attack surface

Given the pervasiveness of password compromise, minimizing the attack surface in your organization is critical. Eliminating use of older, less secure protocols, limiting access entry points, and exercising more significant control of administrative access to resources can help reduce the attack surface area.

Block legacy authentication

Apps using their own legacy methods to authenticate with Azure AD and access company data, pose another risk for organizations. Examples of apps using legacy authentication are POP3, IMAP4, or SMTP clients. Legacy authentication apps authenticate on behalf of the user and prevent Azure AD from doing advanced security evaluations. The alternative, modern authentication, will reduce your security risk, because it supports multi-factor authentication and conditional access. We recommend the following three actions:

1. Block [legacy authentication if you use AD FS](#).
2. Setup [SharePoint Online and Exchange Online to use modern authentication](#).
3. Use [Conditional access policies to block legacy authentication](#).

Block invalid authentication entry points

Using the assume breach mentality, you should reduce the impact of compromised user credentials when they happen. For each app in your environment consider the valid use cases: which groups, which networks, which devices and other elements are authorized – then block the rest. With [Azure AD conditional access](#), you can control how authorized users access their apps and resources based on specific conditions you define.

Block end-user consent

By default, all users in Azure AD are allowed to grant applications that leverage OAuth 2.0 and the Microsoft identity [consent framework](#) permissions to access company data. While consenting does allow users to easily acquire useful applications that integrate with Microsoft 365 and Azure, it can represent a risk if not used and monitored carefully. [Disabling all future user consent operations](#) can help reduce your surface area and mitigate this risk. If end-user consent is disabled previous consent grants will still be honored, but all future consent operations must be performed by an administrator. Before disabling this functionality it is recommended to ensure that users will understand how to request admin approval for new applications; doing this should help reduce user friction, minimize support volume, and make sure users do not sign up for applications using non-Azure AD credentials.

Implement Azure AD Privileged Identity Management

Another impact of "assume breach" is the need to minimize the likelihood a compromised account can operate with a privileged role.

Azure AD Privileged Identity Management (PIM) helps you minimize account privileges by helping you:

- Identify and manage users assigned to administrative roles.
- Understand unused or excessive privilege roles you should remove.
- Establish rules to make sure privileged roles are protected by multi-factor authentication.
- Establish rules to make sure privileged roles are granted only long enough to accomplish the privileged task.

Enable Azure AD PIM, then view the users who are assigned administrative roles and remove unnecessary accounts in those roles. For remaining privileged users, move them from permanent to eligible. Finally, establish appropriate policies to make sure when they need to gain access to those privileged roles, they can do so securely, with the necessary change control.

As part of deploying your privileged account process, follow the [best practice to create at least two emergency accounts](#) to make sure you have access to Azure AD if you lock yourself out.

Step 3 - Automate threat response

Azure Active Directory has many capabilities that automatically intercept attacks, to remove the latency between detection and response. You can reduce the costs and risks, when you reduce the time criminals use to embed themselves into your environment. Here are the concrete steps you can take.

Implement user risk security policy using Azure AD Identity Protection

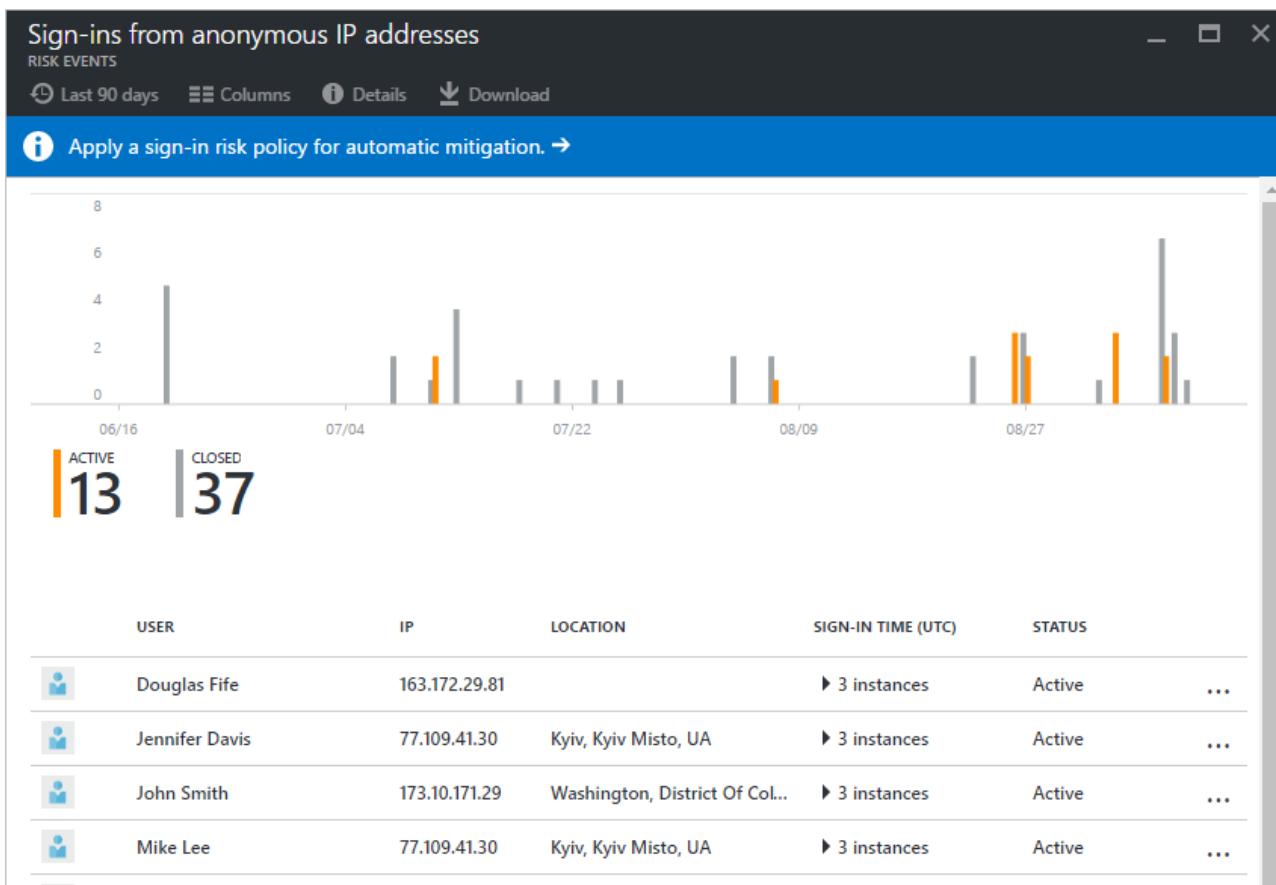
User risk indicates the likelihood a user's identity has been compromised and is calculated based on the [user risk events](#) that are associated with a user's identity. A user risk policy is a conditional access policy that evaluates the risk level to a specific user or group. Based on Low, Medium, High risk-level, a policy can be configured to block access or require a secure password change using multi-factor authentication. Microsoft's recommendation is to require a secure password change for users on high risk.

The screenshot shows the Azure AD Identity Protection interface. The left sidebar has sections for GENERAL (Overview, Getting started), INVESTIGATE (Users flagged for risk, Risk events, Vulnerabilities), and CONFIGURE (Multi-factor authentication regi..., User risk policy, Sign-in risk policy). The main area title is "Azure AD Identity Protection - Users flagged for risk". It includes a search bar, download and refresh buttons, and a message: "Apply a user risk policy for automatic mitigation. →". Below is a table with columns: USER, MFA, RISK LEVEL, RISK EVENTS, STATUS, and LAST UPDATED (UTC). The table lists several users with their respective details.

USER	MFA	RISK LEVEL	RISK EVENTS	STATUS	LAST UPDATED (UTC)
John Nash		High	215 risk events	At risk	12/7/2016 10:51 AM
Jon Doe	✓	Medium	1 risk event	At risk	11/15/2016 7:18 PM
Junpu Chen	✓	Medium	0 risk events	At risk	9/12/2016 10:57 AM
Security Admin	✓	Medium	0 risk events	At risk	8/23/2016 2:40 PM
Security Reader	✓	Medium	0 risk events	At risk	8/23/2016 2:40 PM
Ben Hecht		Secured	0 risk events	Remediated	1/31/2016 3:21 PM
On-Premises Directory Synchroniz...		Secured	0 risk events	Remediated	12/14/2015 7:21 PM
secReader2		Secured	0 risk events	Remediated	9/7/2016 5:18 AM

Implement sign-in risk policy using Azure AD Identity Protection

Sign-in risk is the likelihood someone other than the account owner is attempting to sign on using the identity. A [sign-in risk policy](#) is a conditional access policy that evaluates the risk level to a specific user or group. Based on the risk level (high/medium/low), a policy can be configured to block access or force multi-factor authentication. Make sure you force multi-factor authentication on Medium or above risk sign-ins.



Step 4 - Increase your awareness

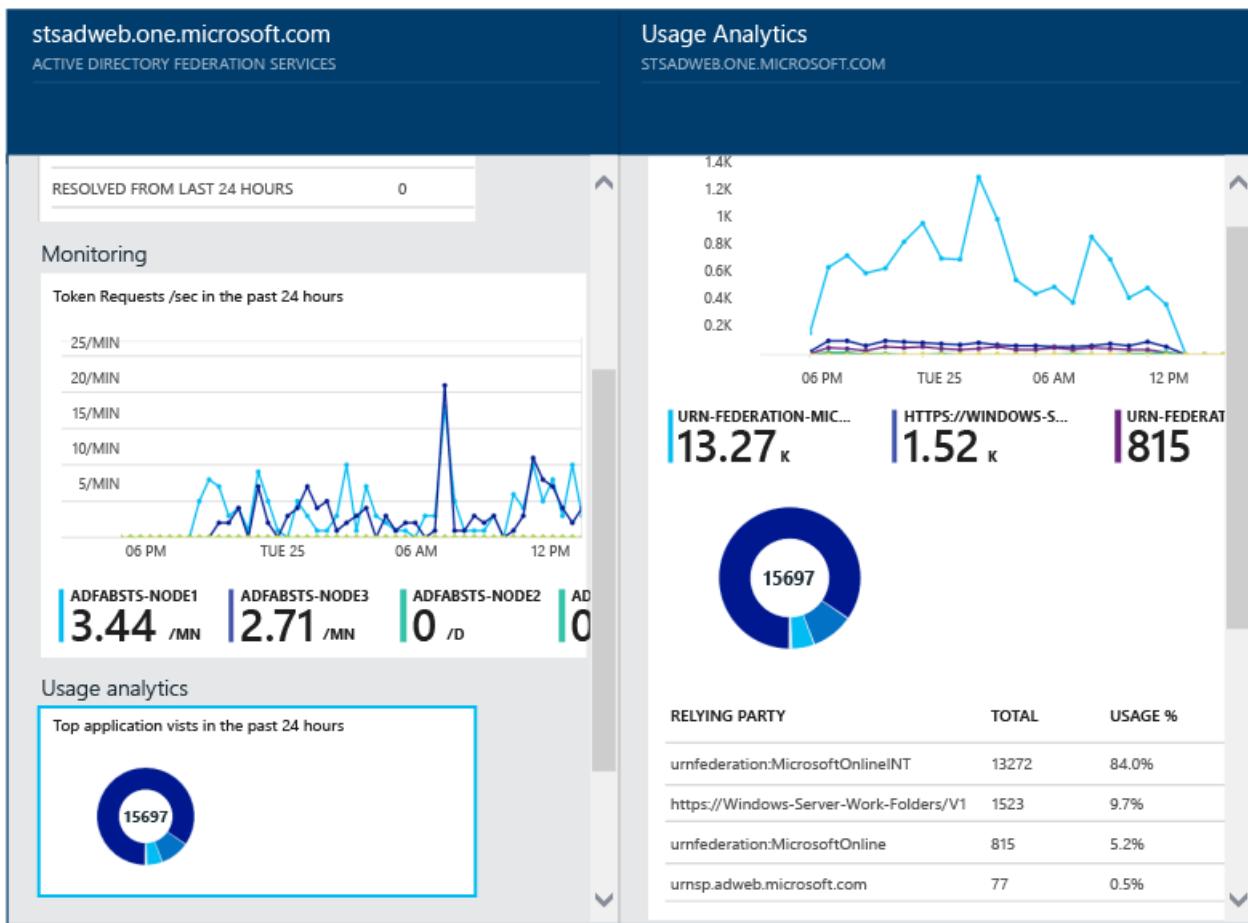
Auditing and logging of security-related events and related alerts are essential components of an efficient protection strategy. Security logs and reports provide you with an electronic record of suspicious activities and help you detect patterns that may indicate attempted or successful external penetration of the network, and internal attacks. You can use auditing to monitor user activity, document regulatory compliance, perform forensic analysis, and more. Alerts provide notifications of security events.

Monitor Azure AD

Microsoft Azure services and features provide you with configurable security auditing and logging options to help you identify gaps in your security policies and mechanisms and address those gaps to help prevent breaches. You can use [Azure Logging and Auditing](#) and use [Audit activity reports in the Azure Active Directory portal](#).

Monitor Azure AD Connect Health in hybrid environments

[Monitoring AD FS with Azure AD Connect Health](#) provides you with greater insight into potential issues and visibility of attacks on your AD FS infrastructure. Azure AD Connect Health delivers alerts with details, resolution steps, and links to related documentation; usage analytics for several metrics related to authentication traffic; performance monitoring and reports.

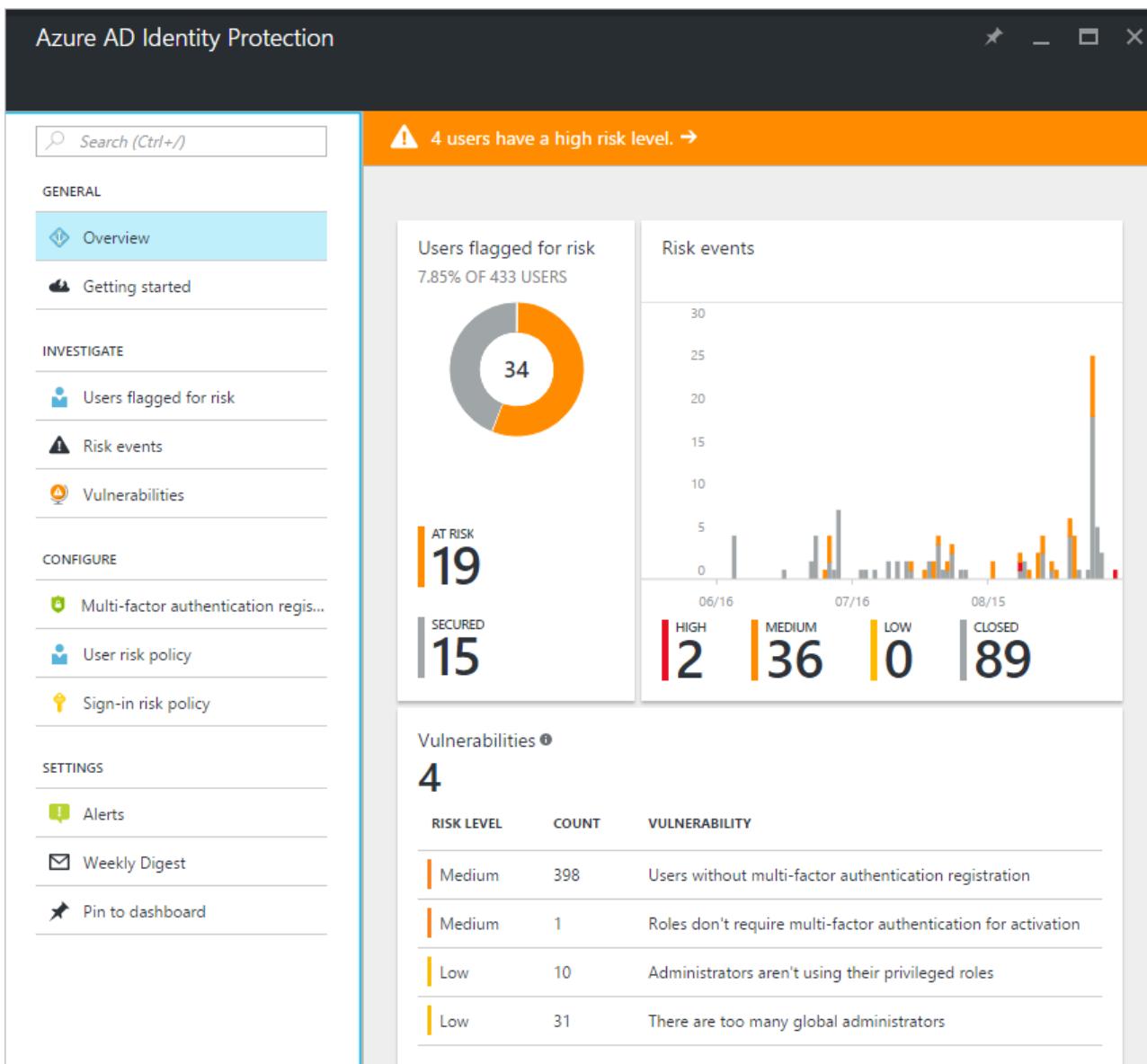


Monitor Azure AD Identity Protection events

Azure AD Identity Protection is a notification, monitoring and reporting tool you can use to detect potential vulnerabilities affecting your organization's identities. It detects risk events, such as leaked credentials, impossible travel, and sign-ins from infected devices, anonymous IP addresses, IP addresses associated with the suspicious activity, and unknown locations. Enable notification alerts to receive email of users at risk and/or a weekly digest email.

Azure AD Identity Protection provides two important reports you should monitor daily:

1. Risky sign-in reports will surface user sign-in activities you should investigate, the legitimate owner may not have performed the sign-in.
2. Risky user reports will surface user accounts that may have been compromised, such as leaked credential that was detected or the user signed in from different locations causing an impossible travel event.



Audit apps and consented permissions

Users can be tricked into navigating to a compromised web site or apps which will gain access to their profile information and user data, such as their email. A malicious actor can use the consented permissions it received to encrypt their mailbox content and demand a ransom to regain your mailbox data. [Administrators should review and audit](#) the permissions given by users.

Step 5 - Enable end-user self-help

As much as possible you'll want to balance security with productivity. Along the same lines of approaching your journey with the mindset that you're setting a foundation for security in the long run, you can remove friction from your organization by empowering your users while remaining vigilant.

Implement self-service password reset

Azure's [self-service password reset \(SSPR\)](#) offers a simple means for IT administrators to allow users to reset or unlock their passwords or accounts without administrator intervention. The system includes detailed reporting that tracks when users access the system, along with notifications to alert you to misuse or abuse.

Implement self-service group management

Azure AD provides the ability to manage access to resources using security groups and Office 365 groups. These groups can be managed by group owners instead of IT administrators. Known as [self-service group management](#), this feature allows group owners who are not assigned an administrative role to create and manage groups without relying on administrators to handle their requests.

Implement Azure AD access reviews

With [Azure AD access reviews](#), you can manage group memberships, access to enterprise applications, and privileged role assignments to make sure you maintain a security standard that does not give users access for extended periods of time when they don't need it.

Summary

There are many aspects to a secure Identity infrastructure, but this five-step checklist will help you quickly accomplish a safer and secure identity infrastructure:

- Strengthen your credentials.
- Reduce your attack surface area.
- Automate threat response.
- Increase your awareness of auditing and monitoring.
- Enable more predictable and complete end-user security with self-help.

We appreciate how seriously you take Identity Security and hope this document is a useful roadmap to a more secure posture for your organization.

Next steps

If you need assistance to plan and deploy the recommendations, refer to the [Azure AD project deployment plans](#) for help.

Azure Identity Management and access control security best practices

11/7/2018 • 14 minutes to read • [Edit Online](#)

Many consider identity to be the new boundary layer for security, taking over that role from the traditional network-centric perspective. This evolution of the primary pivot for security attention and investments come from the fact that network perimeters have become increasingly porous and that perimeter defense cannot be as effective as they once were prior to the explosion of **BYOD** devices and cloud applications.

In this article, we discuss a collection of Azure identity management and access control security best practices. These best practices are derived from our experience with [Azure AD](#) and the experiences of customers like yourself.

For each best practice, we explain:

- What the best practice is
- Why you want to enable that best practice
- What might be the result if you fail to enable the best practice
- Possible alternatives to the best practice
- How you can learn to enable the best practice

This Azure identity management and access control security best practices article is based on a consensus opinion and Azure platform capabilities and feature sets, as they exist at the time this article was written. Opinions and technologies change over time and this article will be updated on a regular basis to reflect those changes.

Azure identity management and access control security best practices discussed in this article include:

- Treat identity as the primary security perimeter
- Centralize identity management
- Enable single sign-on
- Turn on conditional access
- Enable password management
- Enforce multi-factor verification for users
- Use role-based access control
- Lower exposure of privileged accounts
- Control locations where resources are located

Treat identity as the primary security perimeter

Many consider identity to be the primary perimeter for security. This is a shift from the traditional focus on network security. Network perimeters keep getting more porous, and that perimeter defense can't be as effective as it was before the explosion of **BYOD** devices and cloud applications. [Azure Active Directory \(Azure AD\)](#) is the Azure solution for identity and access management. Azure AD is a multitenant, cloud-based directory and identity management service from Microsoft. It combines core directory services, application access management, and identity protection into a single solution.

The following sections list best practices for identity and access security using Azure AD.

Centralize identity management

In a [hybrid identity](#) scenario we recommend that you integrate your on-premises and cloud directories. Integration

enables your IT team to manage accounts from one single location, regardless of where an account is created. Integration also helps your users to be more productive by providing a common identity for accessing both cloud and on-premises resources.

Best practice: Integrate your on-premises directories with Azure AD.

Detail: Use [Azure AD Connect](#) to synchronize your on-premises directory with your cloud directory.

Best practice: Turn on password hash synchronization.

Detail: Password hash synchronization is a feature used to synchronize hashes of user password hashes from an on-premises Active Directory instance to a cloud-based Azure AD instance.

Even if you decide to use federation with Active Directory Federation Services (AD FS) or other identity providers, you can optionally set up password hash synchronization as a backup in case your on-premises servers fail or become temporarily unavailable. This enables users to sign in to the service by using the same password that they use to sign in to their on-premises Active Directory instance. It also allows Identity Protection to detect compromised credentials by comparing those password hashes with passwords known to be compromised, if a user has used their same email address and password on other services not connected to Azure AD.

For more information, see [Implement password hash synchronization with Azure AD Connect sync](#).

Organizations that don't integrate their on-premises identity with their cloud identity can have more overhead in managing accounts. This overhead increases the likelihood of mistakes and security breaches.

Enable single sign-on

In a mobile-first, cloud-first world, you want to enable single sign-on (SSO) to devices, apps, and services from anywhere so your users can be productive wherever and whenever. When you have multiple identity solutions to manage, this becomes an administrative problem not only for IT but also for users who have to remember multiple passwords.

By using the same identity solution for all your apps and resources, you can achieve SSO. And your users can use the same set of credentials to sign in and access the resources that they need, whether the resources are located on-premises or in the cloud.

Best practice: Enable SSO.

Detail: Azure AD [extends on-premises Active Directory](#) to the cloud. Users can use their primary work or school account for their domain-joined devices, company resources, and all of the web and SaaS applications that they need to get their jobs done. Users don't have to remember multiple sets of usernames and passwords, and their application access can be automatically provisioned (or deprovisioned) based on their organization group memberships and their status as an employee. And you can control that access for gallery apps or for your own on-premises apps that you've developed and published through the [Azure AD Application Proxy](#).

Use SSO to enable users to access their [SaaS applications](#) based on their work or school account in Azure AD. This is applicable not only for Microsoft SaaS apps, but also other apps, such as [Google Apps](#) and [Salesforce](#). You can configure your application to use Azure AD as a [SAML-based identity](#) provider. As a security control, Azure AD does not issue a token that allows users to sign in to the application unless they have been granted access through Azure AD. You can grant access directly, or through a group that users are a member of.

Organizations that don't create a common identity to establish SSO for their users and applications are more exposed to scenarios where users have multiple passwords. These scenarios increase the likelihood of users reusing passwords or using weak passwords.

Turn on conditional access

Users can access your organization's resources by using a variety of devices and apps from anywhere. As an IT administrator, you want to make sure that these devices meet your standards for security and compliance. Just

focusing on who can access a resource is not sufficient anymore.

To balance security and productivity, you need to think about how a resource is accessed before you can make an access control decision. With Azure AD conditional access, you can address this requirement. With conditional access, you can make automated access control decisions for accessing your cloud apps that are based on conditions.

Best practice: Manage and control access to corporate resources.

Detail: Configure Azure AD [conditional access](#) based on a group, location, and application sensitivity for SaaS apps and Azure AD-connected apps.

Enable password management

If you have multiple tenants or you want to enable users to [reset their own passwords](#), it's important that you use appropriate security policies to prevent abuse.

Best practice: Set up self-service password reset (SSPR) for your users.

Detail: Use the Azure AD [self-service password reset](#) feature.

Best practice: Monitor how or if SSPR is really being used.

Detail: Monitor the users who are registering by using the Azure AD [Password Reset Registration Activity report](#).

The reporting feature that Azure AD provides helps you answer questions by using prebuilt reports. If you're appropriately licensed, you can also create custom queries.

Enforce multi-factor verification for users

We recommend that you require two-step verification for all of your users. This includes administrators and others in your organization who can have a significant impact if their account is compromised (for example, financial officers).

There are multiple options for requiring two-step verification. The best option for you depends on your goals, the Azure AD edition you're running, and your licensing program. See [How to require two-step verification for a user](#) to determine the best option for you. See the [Azure AD](#) and [Azure Multi-Factor Authentication](#) pricing pages for more information about licenses and pricing.

Following are options and benefits for enabling two-step verification:

Option 1: Enable Multi-Factor Authentication by changing user state.

Benefit: This is the traditional method for requiring two-step verification. It works with both [Azure Multi-Factor Authentication in the cloud](#) and [Azure Multi-Factor Authentication Server](#). Using this method requires users to perform two-step verification every time they sign in and overrides conditional access policies.

Option 2: Enable Multi-Factor Authentication with conditional access policy.

Benefit: This option allows you to prompt for two-step verification under specific conditions by using [conditional access](#). Specific conditions can be user sign-in from different locations, untrusted devices, or applications that you consider risky. Defining specific conditions where you require two-step verification enables you to avoid constant prompting for your users, which can be an unpleasant user experience.

This is the most flexible way to enable two-step verification for your users. Enabling a conditional access policy works only for Azure Multi-Factor Authentication in the cloud and is a premium feature of Azure AD. You can find more information on this method in [Deploy cloud-based Azure Multi-Factor Authentication](#).

Option 3: Enable Multi-Factor Authentication with conditional access policies by evaluating user and sign-in risk of Azure AD Identity Protection.

Benefit: This option enables you to:

- Detect potential vulnerabilities that affect your organization's identities.

- Configure automated responses to detected suspicious actions that are related to your organization's identities.
- Investigate suspicious incidents and take appropriate action to resolve them.

This method uses the Azure AD Identity Protection risk evaluation to determine if two-step verification is required based on user and sign-in risk for all cloud applications. This method requires Azure Active Directory P2 licensing. You can find more information on this method in [Azure Active Directory Identity Protection](#).

NOTE

Option 1, enabling Multi-Factor Authentication by changing the user state, overrides conditional access policies. Because options 2 and 3 use conditional access policies, you cannot use option 1 with them.

Organizations that don't add extra layers of identity protection, such as two-step verification, are more susceptible for credential theft attack. A credential theft attack can lead to data compromise.

Use role-based access control (RBAC)

Restricting access based on the [need to know](#) and [least privilege](#) security principles is imperative for organizations that want to enforce security policies for data access. You can use [role-based access control \(RBAC\)](#) to assign permissions to users, groups, and applications at a certain scope. The scope of a role assignment can be a subscription, a resource group, or a single resource.

You can use [built-in RBAC](#) roles in Azure to assign privileges to users. Organizations that do not enforce data access control by using capabilities such as RBAC might be giving more privileges than necessary to their users. This can lead to data compromise by allowing user access to certain types of data (for example, high business impact) that they shouldn't have.

Lower exposure of privileged accounts

Securing privileged access is a critical first step to protecting business assets. Minimizing the number of people who have access to secure information or resources reduces the chance of a malicious user getting access, or an authorized user inadvertently affecting a sensitive resource.

Privileged accounts are accounts that administer and manage IT systems. Cyber attackers target these accounts to gain access to an organization's data and systems. To secure privileged access, you should isolate the accounts and systems from the risk of being exposed to a malicious user.

We recommend that you develop and follow a roadmap to secure privileged access against cyber attackers. For information about creating a detailed roadmap to secure identities and access that are managed or reported in Azure AD, Microsoft Azure, Office 365, and other cloud services, review [Securing privileged access for hybrid and cloud deployments in Azure AD](#).

The following summarizes the best practices found in [Securing privileged access for hybrid and cloud deployments in Azure AD](#):

Best practice: Manage, control, and monitor access to privileged accounts.

Detail: Turn on [Azure AD Privileged Identity Management](#). After you turn on Privileged Identity Management, you'll receive notification email messages for privileged access role changes. These notifications provide early warning when additional users are added to highly privileged roles in your directory.

Best practice: Identify and categorize accounts that are in highly privileged roles.

Detail: After turning on Azure AD Privileged Identity Management, view the users who are in the global administrator, privileged role administrator, and other highly privileged roles. Remove any accounts that are no longer needed in those roles, and categorize the remaining accounts that are assigned to admin roles:

- Individually assigned to administrative users, and can be used for non-administrative purposes (for example,

personal email)

- Individually assigned to administrative users and designated for administrative purposes only
- Shared across multiple users
- For emergency access scenarios
- For automated scripts
- For external users

Best practice: Implement “just in time” (JIT) access to further lower the exposure time of privileges and increase your visibility into the use of privileged accounts.

Detail: Azure AD Privileged Identity Management lets you:

- Limit users to only taking on their privileges JIT.
- Assign roles for a shortened duration with confidence that the privileges are revoked automatically.

Best practice: Define at least two emergency access accounts.

Detail: Emergency access accounts help organizations restrict privileged access in an existing Azure Active Directory environment. These accounts are highly privileged and are not assigned to specific individuals. Emergency access accounts are limited to scenarios where normal administrative accounts can't be used. Organizations must limit the emergency account's usage to only the necessary amount of time.

Evaluate the accounts that are assigned or eligible for the global admin role. If you don't see any cloud-only accounts by using the `*.onmicrosoft.com` domain (intended for emergency access), create them. For more information, see [Managing emergency access administrative accounts in Azure AD](#).

Best practice: Turn on Multi-Factor Authentication, and register all other highly privileged single-user non-federated admin accounts.

Detail: Require Azure Multi-Factor Authentication at sign-in for all individual users who are permanently assigned to one or more of the Azure AD admin roles: global administrator, privileged role administrator, Exchange Online administrator, and SharePoint Online administrator. Use the guide to enable [Multi-Factor Authentication for your admin accounts](#) and ensure that all those users have [registered](#).

Best practice: Take steps to mitigate the most frequently used attacked techniques.

Detail: [Identify Microsoft accounts in administrative roles that need to be switched to work or school accounts](#)

[Ensure separate user accounts and mail forwarding for global administrator accounts](#)

[Ensure that the passwords of administrative accounts have recently changed](#)

[Turn on password hash synchronization](#)

[Require Multi-Factor Authentication for users in all privileged roles as well as exposed users](#)

[Obtain your Office 365 Secure Score \(if using Office 365\)](#)

[Review the Office 365 security and compliance guidance \(if using Office 365\)](#)

[Configure Office 365 Activity Monitoring \(if using Office 365\)](#)

[Establish incident/emergency response plan owners](#)

[Secure on-premises privileged administrative accounts](#)

If you don't secure privileged access, you might find that you have too many users in highly privileged roles and are more vulnerable to attacks. Malicious actors, including cyber attackers, often target admin accounts and other elements of privileged access to gain access to sensitive data and systems by using credential theft.

Control locations where resources are created

Enabling cloud operators to perform tasks while preventing them from breaking conventions that are needed to manage your organization's resources is very important. Organizations that want to control the locations where resources are created should hard code these locations.

You can use [Azure Resource Manager](#) to create security policies whose definitions describe the actions or resources that are specifically denied. You assign those policy definitions at the desired scope, such as the subscription, the resource group, or an individual resource.

NOTE

Security policies are not the same as RBAC. They actually use RBAC to authorize users to create those resources.

Organizations that are not controlling how resources are created are more susceptible to users who might abuse the service by creating more resources than they need. Hardening the resource creation process is an important step to securing a multitenant scenario.

Actively monitor for suspicious activities

An active identity monitoring system can quickly detect suspicious behavior and trigger an alert for further investigation. The following table lists two Azure AD capabilities that can help organizations monitor their identities:

Best practice: Have a method to identify:

- Attempts to sign in [without being traced](#).
- [Brute force](#) attacks against a particular account.
- Attempts to sign in from multiple locations.
- Sign-ins from [infected devices](#).
- Suspicious IP addresses.

Detail: Use Azure AD Premium [anomaly reports](#). Have processes and procedures in place for IT admins to run these reports on a daily basis or on demand (usually in an incident response scenario).

Best practice: Have an active monitoring system that notifies you of risks and can adjust risk level (high, medium, or low) to your business requirements.

Detail: Use [Azure AD Identity Protection](#), which flags the current risks on its own dashboard and sends daily summary notifications via email. To help protect your organization's identities, you can configure risk-based policies that automatically respond to detected issues when a specified risk level is reached.

Organizations that don't actively monitor their identity systems are at risk of having user credentials compromised. Without knowledge that suspicious activities are taking place through these credentials, organizations can't mitigate this type of threat.

Next step

See [Azure security best practices and patterns](#) for more security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.

Enforce multi-factor authentication (MFA) for subscription administrators

4/25/2018 • 2 minutes to read • [Edit Online](#)

When you create your administrators, including your global administrator account, it is essential that you use very strong authentication methods.

You can perform day-to-day administration by assigning specific administrator roles—such as Exchange administrator or Password administrator—to user accounts of IT staff as needed. Additionally, enabling [Azure Multi-factor Authentication \(MFA\)](#) for your administrators adds a second layer of security to user sign-ins and transactions. Azure MFA also helps IT reduce the likelihood that a compromised credential will have access to organization's data.

For example: you enforce Azure MFA for your users and configure it to use a phone call or text message as verification. If the user's credentials are compromised, the attacker won't be able to access any resource since he will not have access to user's phone. Organizations that do not add extra layers of identity protection are more susceptible for credential theft attack, which may lead to data compromise.

One alternative for organizations that want to keep the entire authentication control on-premises is to use [Azure Multi-Factor Authentication Server](#), also called "MFA on-premises". By using this method, you will still be able to enforce multi-factor authentication, while keeping the MFA server on-premises.

To check who in your organization has administrative privileges you can verify by using the following Microsoft Azure AD V2 PowerShell command:

```
Get-AzureADDirectoryRole | Where { $_.DisplayName -eq "Company Administrator" } | Get-AzureADDirectoryRoleMember | Ft DisplayName
```

Enabling MFA

Review how [MFA](#) operates before you proceed.

As long as your users have licenses that include Azure Multi-Factor Authentication, there's nothing that you need to do to turn on Azure MFA. You can start requiring two-step verification on an individual user basis. The licenses that enable Azure MFA are:

- Azure Multi-Factor Authentication
- Azure Active Directory Premium
- Enterprise Mobility + Security

Turn on two-step verification for users

Use one of the procedures listed in [How to require two-step verification](#) for a user or group to start using Azure MFA. You can choose to enforce two-step verification for all sign-ins, or you can create conditional access policies to require two-step verification only when it matters to you.

Azure network security overview

11/7/2018 • 22 minutes to read • [Edit Online](#)

Network security could be defined as the process of protecting resources from unauthorized access or attack by applying controls to network traffic. The goal is to ensure that only legitimate traffic is allowed. Azure includes a robust networking infrastructure to support your application and service connectivity requirements. Network connectivity is possible between resources located in Azure, between on-premises and Azure hosted resources, and to and from the internet and Azure.

This article covers some of the options that Azure offers in the area of network security. You can learn about:

- Azure networking
- Network access control
- Azure Firewall
- Secure remote access and cross-premises connectivity
- Availability
- Name resolution
- Perimeter network (DMZ) architecture
- Azure DDoS protection
- Azure Front Door
- Traffic manager
- Monitoring and threat detection

Azure networking

Azure requires virtual machines to be connected to an Azure Virtual Network. A virtual network is a logical construct built on top of the physical Azure network fabric. Each virtual network is isolated from all other virtual networks. This helps ensure that network traffic in your deployments is not accessible to other Azure customers.

Learn more:

- [Virtual network overview](#)

Network access control

Network access control is the act of limiting connectivity to and from specific devices or subnets within a virtual network. The goal of network access control is to limit access to your virtual machines and services to approved users and devices. Access controls are based on decisions to allow or deny connections to and from your virtual machine or service.

Azure supports several types of network access control, such as:

- Network layer control
- Route control and forced tunneling
- Virtual network security appliances

Network layer control

Any secure deployment requires some measure of network access control. The goal of network access control is to restrict virtual machine communication to the necessary systems. Other communication attempts are blocked.

NOTE

Storage Firewalls are covered in the [Azure storage security overview](#) article

Network security rules (NSGs)

If you need basic network level access control (based on IP address and the TCP or UDP protocols), you can use Network Security Groups (NSGs). An NSG is a basic, stateful, packet filtering firewall, and it enables you to control access based on a [5-tuple](#). NSGs include functionality to simplify management and reduce the chances of configuration mistakes:

- **Augmented security rules** simplify NSG rule definition and allow you to create complex rules rather than having to create multiple simple rules to achieve the same result.
- **Service tags** are Microsoft created labels that represent a group of IP addresses. They update dynamically to include IP ranges that meet the conditions that define inclusion in the label. For example, if you want to create a rule that applies to all Azure storage on the east region you can use Storage.EastUS
- **Application security groups** allow you to deploy resources to application groups and control the access to those resources by creating rules that use those application groups. For example, if you have webservers deployed to the 'Webservers' application group you can create a rule that applies a NSG allowing 443 traffic from the Internet to all systems in the 'Webservers' application group.

NSGs do not provide application layer inspection or authenticated access controls.

Learn more:

- [Network Security Groups](#)

ASC just in time VM access

[Azure security center](#) can manage the NSGs on VMs and lock access to the VM until a user with the appropriate role-based access control [RBAC](#) permissions requests access. When the user is successfully authorized ASC makes modifications to the NSGs to allow access to selected ports for the time specified. When the time expires the NSGs are restored to their previous secured state.

Learn more:

- [Azure Security Center Just in T](#)

Service endpoints

Service endpoints are another way to apply control over your traffic. You can limit communication with supported services to just your VNets over a direct connection. Traffic from your VNet to the specified Azure service remains on the Microsoft Azure backbone network.

Learn more:

- [Service endpoints](#)

Route control and forced tunneling

The ability to control routing behavior on your virtual networks is critical. If routing is configured incorrectly, applications and services hosted on your virtual machine might connect to unauthorized devices, including systems owned and operated by potential attackers.

Azure networking supports the ability to customize the routing behavior for network traffic on your virtual networks. This enables you to alter the default routing table entries in your virtual network. Control of routing behavior helps you make sure that all traffic from a certain device or group of devices enters or leaves your virtual network through a specific location.

For example, you might have a virtual network security appliance on your virtual network. You want to make sure that all traffic to and from your virtual network goes through that virtual security appliance. You can do this by

configuring [User Defined Routes](#) (UDRs) in Azure.

Forced tunneling is a mechanism you can use to ensure that your services are not allowed to initiate a connection to devices on the internet. Note that this is different from accepting incoming connections and then responding to them. Front-end web servers need to respond to requests from internet hosts, and so internet-sourced traffic is allowed inbound to these web servers and the web servers are allowed to respond.

What you don't want to allow is a front-end web server to initiate an outbound request. Such requests might represent a security risk because these connections can be used to download malware. Even if you do want these front-end servers to initiate outbound requests to the internet, you might want to force them to go through your on-premises web proxies. This enables you to take advantage of URL filtering and logging.

Instead, you would want to use forced tunneling to prevent this. When you enable forced tunneling, all connections to the internet are forced through your on-premises gateway. You can configure forced tunneling by taking advantage of UDRs.

Learn more:

- [What are User Defined Routes and IP Forwarding](#)

Virtual network security appliances

While NSGs, UDRs, and forced tunneling provide you a level of security at the network and transport layers of the [OSI model](#), you might also want to enable security at levels higher than the network.

For example, your security requirements might include:

- Authentication and authorization before allowing access to your application
- Intrusion detection and intrusion response
- Application layer inspection for high-level protocols
- URL filtering
- Network level antivirus and Antimalware
- Anti-bot protection
- Application access control
- Additional DDoS protection (above the DDoS protection provided by the Azure fabric itself)

You can access these enhanced network security features by using an Azure partner solution. You can find the most current Azure partner network security solutions by visiting the [Azure Marketplace](#), and searching for "security" and "network security".

Azure Firewall

Azure Firewall is a managed, cloud-based network security service that protects your Azure Virtual Network resources. It is a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability. Some features include:

- High availability
- Cloud scalability
- Application FQDN filtering rules
- Network traffic filtering rules

Learn more:

- [Azure Firewall overview](#)

Secure remote access and cross-premises connectivity

Setup, configuration, and management of your Azure resources needs to be done remotely. In addition, you might want to deploy [hybrid IT](#) solutions that have components on-premises and in the Azure public cloud. These scenarios require secure remote access.

Azure networking supports the following secure remote access scenarios:

- Connect individual workstations to a virtual network
- Connect your on-premises network to a virtual network with a VPN
- Connect your on-premises network to a virtual network with a dedicated WAN link
- Connect virtual networks to each other

Connect individual workstations to a virtual network

You might want to enable individual developers or operations personnel to manage virtual machines and services in Azure. For example, let's say you need access to a virtual machine on a virtual network. But your security policy does not allow RDP or SSH remote access to individual virtual machines. In this case, you can use a [point-to-site VPN](#) connection.

The point-to-site VPN connection enables you to set up a private and secure connection between the user and the virtual network. When the VPN connection is established, the user can RDP or SSH over the VPN link into any virtual machine on the virtual network. (This assumes that the user can authenticate and is authorized.) Point-to-site VPN supports:

- Secure Socket Tunneling Protocol (SSTP), a proprietary SSL-based VPN protocol. An SSL VPN solution can penetrate firewalls, since most firewalls open TCP port 443, which SSL uses. SSTP is only supported on Windows devices. Azure supports all versions of Windows that have SSTP (Windows 7 and later).
- IKEv2 VPN, a standards-based IPsec VPN solution. IKEv2 VPN can be used to connect from Mac devices (OSX versions 10.11 and above).
- [OpenVPN](#)

Learn more:

- [Configure a point-to-site connection to a virtual network using PowerShell](#)

Connect your on-premises network to a virtual network with a VPN

You might want to connect your entire corporate network, or portions of it, to a virtual network. This is common in hybrid IT scenarios, where organizations [extend their on-premises datacenter into Azure](#). In many cases, organizations host parts of a service in Azure, and parts on-premises. For example, they might do so when a solution includes front-end web servers in Azure and back-end databases on-premises. These types of "cross-premises" connections also make management of Azure located resources more secure, and enable scenarios such as extending Active Directory domain controllers into Azure.

One way to accomplish this is to use a [site-to-site VPN](#). The difference between a site-to-site VPN and a point-to-site VPN is that the latter connects a single device to a virtual network. A site-to-site VPN connects an entire network (such as your on-premises network) to a virtual network. Site-to-site VPNs to a virtual network use the highly secure IPsec tunnel mode VPN protocol.

Learn more:

- [Create a Resource Manager VNet with a site-to-site VPN connection using the Azure portal](#)
- [Planning and design for VPN gateway](#)

Connect your on-premises network to a virtual network with a dedicated WAN link

Point-to-site and site-to-site VPN connections are effective for enabling cross-premises connectivity. However, some organizations consider them to have the following drawbacks:

- VPN connections move data over the internet. This exposes these connections to potential security issues involved with moving data over a public network. In addition, reliability and availability for internet connections cannot be guaranteed.
- VPN connections to virtual networks might not have the bandwidth for some applications and purposes, as they max out at around 200 Mbps.

Organizations that need the highest level of security and availability for their cross-premises connections typically use dedicated WAN links to connect to remote sites. Azure provides you the ability to use a dedicated WAN link that you can use to connect your on-premises network to a virtual network. Azure ExpressRoute, Express route direct, and Express route global reach enable this.

Learn more:

- [ExpressRoute technical overview](#)
- [ExpressRoute direct](#)
- [Express route global reach](#)

Connect virtual networks to each other

It is possible to use many virtual networks for your deployments. There are various reasons why you might do this. You might want to simplify management, or you might want increased security. Regardless of the motivation for putting resources on different virtual networks, there might be times when you want resources on each of the networks to connect with one another.

One option is for services on one virtual network to connect to services on another virtual network, by “looping back” through the internet. The connection starts on one virtual network, goes through the internet, and then comes back to the destination virtual network. This option exposes the connection to the security issues inherent in any internet-based communication.

A better option might be to create a site-to-site VPN that connects between two virtual networks. This method uses the same [IPSec tunnel mode](#) protocol as the cross-premises site-to-site VPN connection mentioned above.

The advantage of this approach is that the VPN connection is established over the Azure network fabric, instead of connecting over the internet. This provides you an extra layer of security, compared to site-to-site VPNs that connect over the internet.

Learn more:

- [Configure a VNet-to-VNet Connection by using Azure Resource Manager and PowerShell](#)

Another way to connect your virtual networks is [VNET peering](#). This feature allows you to connect two Azure networks so that communication between them happens over the Microsoft backbone infrastructure without it ever going over the Internet. VNET peering can connect two VNETs within the same region or two VNETs across Azure regions. NSGs can be used to limit connectivity between different subnets or systems.

Availability

Availability is a key component of any security program. If your users and systems can't access what they need to access over the network, the service can be considered compromised. Azure has networking technologies that support the following high-availability mechanisms:

- HTTP-based load balancing
- Network level load balancing
- Global load balancing

Load balancing is a mechanism designed to equally distribute connections among multiple devices. The goals of load balancing are:

- To increase availability. When you load balance connections across multiple devices, one or more of the devices can become unavailable without compromising the service. The services running on the remaining online devices can continue to serve the content from the service.
- To increase performance. When you load balance connections across multiple devices, a single device doesn't have to handle all processing. Instead, the processing and memory demands for serving the content is spread across multiple devices.

HTTP-based load balancing

Organizations that run web-based services often desire to have an HTTP-based load balancer in front of those web services. This helps ensure adequate levels of performance and high availability. Traditional, network-based load balancers rely on network and transport layer protocols. HTTP-based load balancers, on the other hand, make decisions based on characteristics of the HTTP protocol.

Azure Application Gateway provides HTTP-based load balancing for your web-based services. Application Gateway supports:

- Cookie-based session affinity. This capability makes sure that connections established to one of the servers behind that load balancer stays intact between the client and server. This ensures stability of transactions.
- SSL offload. When a client connects with the load balancer, that session is encrypted by using the HTTPS (SSL) protocol. However, in order to increase performance, you can use the HTTP (unencrypted) protocol to connect between the load balancer and the web server behind the load balancer. This is referred to as "SSL offload," because the web servers behind the load balancer don't experience the processor overhead involved with encryption. The web servers can therefore service requests more quickly.
- URL-based content routing. This feature makes it possible for the load balancer to make decisions about where to forward connections based on the target URL. This provides a lot more flexibility than solutions that make load balancing decisions based on IP addresses.

Learn more:

- [Application Gateway overview](#)

Network level load balancing

In contrast to HTTP-based load balancing, network level load balancing makes decisions based on IP address and port (TCP or UDP) numbers. You can gain the benefits of network level load balancing in Azure by using Azure Load Balancer. Some key characteristics of Load Balancer include:

- Network level load balancing based on IP address and port numbers.
- Support for any application layer protocol.
- Load balances to Azure virtual machines and cloud services role instances.
- Can be used for both internet-facing (external load balancing) and non-internet facing (internal load balancing) applications and virtual machines.
- Endpoint monitoring, which is used to determine if any of the services behind the load balancer have become unavailable.

Learn more:

- [Internet-facing load balancer between multiple virtual machines or services](#)
- [Internal load balancer overview](#)

Global load balancing

Some organizations want the highest level of availability possible. One way to reach this goal is to host applications in globally distributed datacenters. When an application is hosted in datacenters located throughout the world, it's possible for an entire geopolitical region to become unavailable, and still have the application up and running.

This load-balancing strategy can also yield performance benefits. You can direct requests for the service to the datacenter that is nearest to the device that is making the request.

In Azure, you can gain the benefits of global load balancing by using Azure Traffic Manager.

Learn more:

- [What is Traffic Manager?](#)

Name resolution

Name resolution is a critical function for all services you host in Azure. From a security perspective, compromise of the name resolution function can lead to an attacker redirecting requests from your sites to an attacker's site. Secure name resolution is a requirement for all your cloud hosted services.

There are two types of name resolution you need to address:

- Internal name resolution. This is used by services on your virtual networks, your on-premises networks, or both. Names used for internal name resolution are not accessible over the internet. For optimal security, it's important that your internal name resolution scheme is not accessible to external users.
- External name resolution. This is used by people and devices outside of your on-premises networks and virtual networks. These are the names that are visible to the internet, and are used to direct connection to your cloud-based services.

For internal name resolution, you have two options:

- A virtual network DNS server. When you create a new virtual network, a DNS server is created for you. This DNS server can resolve the names of the machines located on that virtual network. This DNS server is not configurable, is managed by the Azure fabric manager, and can therefore help you secure your name resolution solution.
- Bring your own DNS server. You have the option of putting a DNS server of your own choosing on your virtual network. This DNS server can be an Active Directory integrated DNS server, or a dedicated DNS server solution provided by an Azure partner, which you can obtain from the Azure Marketplace.

Learn more:

- [Virtual network overview](#)
- [Manage DNS Servers used by a virtual network](#)

For external name resolution, you have two options:

- Host your own external DNS server on-premises.
- Host your own external DNS server with a service provider.

Many large organizations host their own DNS servers on-premises. They can do this because they have the networking expertise and global presence to do so.

In most cases, it's better to host your DNS name resolution services with a service provider. These service providers have the network expertise and global presence to ensure very high availability for your name resolution services. Availability is essential for DNS services, because if your name resolution services fail, no one will be able to reach your internet facing services.

Azure provides you with a highly available and high-performing external DNS solution in the form of Azure DNS. This external name resolution solution takes advantage of the worldwide Azure DNS infrastructure. It allows you to host your domain in Azure, using the same credentials, APIs, tools, and billing as your other Azure services. As part of Azure, it also inherits the strong security controls built into the platform.

Learn more:

- [Azure DNS overview](#)
- [Azure DNS private zones](#) allows you to configure private DNS names for Azure resources rather than the automatically assigned names without the need to add a custom DNS solution.

Perimeter network architecture

Many large organizations use perimeter networks to segment their networks, and create a buffer-zone between the internet and their services. The perimeter portion of the network is considered a low-security zone, and no high-value assets are placed in that network segment. You'll typically see network security devices that have a network interface on the perimeter network segment. Another network interface is connected to a network that has virtual machines and services that accept inbound connections from the internet.

You can design perimeter networks in a number of different ways. The decision to deploy a perimeter network, and then what type of perimeter network to use if you decide to use one, depends on your network security requirements.

Learn more:

- [Microsoft Cloud Services and Network Security](#)

Azure DDoS protection

Distributed denial of service (DDoS) attacks are some of the largest availability and security concerns facing customers that are moving their applications to the cloud. A DDoS attack attempts to exhaust an application's resources, making the application unavailable to legitimate users. DDoS attacks can be targeted at any endpoint that is publicly reachable through the internet. Microsoft provides DDoS protection known as **Basic** as part of the Azure Platform. This comes at no charge and includes always on monitoring and real-time mitigation of common network level attacks. In addition to the protections included with DDoS protection **Basic** you can enable the **Standard** option. DDoS Protection Standard features include:

- **Native platform integration:** Natively integrated into Azure. Includes configuration through the Azure portal. DDoS Protection Standard understands your resources and resource configuration.
- **Turn-key protection:** Simplified configuration immediately protects all resources on a virtual network as soon as DDoS Protection Standard is enabled. No intervention or user definition is required. DDoS Protection Standard instantly and automatically mitigates the attack, once it is detected.
- **Always-on traffic monitoring:** Your application traffic patterns are monitored 24 hour a day, 7 days a week, looking for indicators of DDoS attacks. Mitigation is performed when protection policies are exceeded.
- **Attack Mitigation Reports** Attack Mitigation Reports use aggregated network flow data to provide detailed information about attacks targeted at your resources.
- **Attack Mitigation Flow Logs** Attack Mitigation Flow Logs allow you to review the dropped traffic, forwarded traffic and other attack data in near real-time during an active DDoS attack.
- **Adaptive tuning:** Intelligent traffic profiling learns your application's traffic over time, and selects and updates the profile that is the most suitable for your service. The profile adjusts as traffic changes over time. Layer 3 to layer 7 protection: Provides full stack DDoS protection, when used with a web application firewall.
- **Extensive mitigation scale:** Over 60 different attack types can be mitigated, with global capacity, to protect against the largest known DDoS attacks.
- **Attack metrics:** Summarized metrics from each attack are accessible through Azure Monitor.
- **Attack alerting:** Alerts can be configured at the start and stop of an attack, and over the attack's duration, using built-in attack metrics. Alerts integrate into your operational software like Microsoft Azure Log Analytics, Splunk, Azure Storage, Email, and the Azure portal.
- **Cost guarantee:** Data-transfer and application scale-out service credits for documented DDoS attacks.
- **DDoS Rapid responsive** DDoS Protection Standard customers now have access to Rapid Response team during an active attack. DRR can help with attack investigation, custom mitigations during an attack and post-

attack analysis.

Learn more:

- [DDoS protection overview](#)

Azure Front Door

Azure Front Door Service enables you to define, manage, and monitor the global routing of your web traffic. It optimizes your traffic's routing for best performance and high availability. Azure Front Door allows you to author custom web application firewall (WAF) rules for access control to protect your HTTP/HTTPS workload from exploitation based on client IP addresses, country code, and http parameters. Additionally, Front Door also enables you to create rate limiting rules to battle malicious bot traffic, it includes SSL offloading and per-HTTP/HTTPS request, application-layer processing.

Front Door platform itself is protected by Azure DDoS Protection Basic. For further protection, Azure DDoS Protection Standard may be enabled at your VNETs and safeguard resources from network layer (TCP/UDP) attacks via auto tuning and mitigation. Front Door is a layer 7 reverse proxy, it only allows web traffic to pass through to back end servers and block other types of traffic by default.

Learn more:

- For more information on the whole set of Azure Front door capabilities you can review the [Azure Front Door overview](#)

Azure Traffic manager

Azure Traffic Manager is a DNS-based traffic load balancer that enables you to distribute traffic optimally to services across global Azure regions, while providing high availability and responsiveness. Traffic Manager uses DNS to direct client requests to the most appropriate service endpoint based on a traffic-routing method and the health of the endpoints. An endpoint is any Internet-facing service hosted inside or outside of Azure. Traffic manager monitors the end points and does not direct traffic to any endpoints that are unavailable.

Learn more:

- [Azure Traffic manager overview](#)

Monitoring and threat detection

Azure provides capabilities to help you in this key area with early detection, monitoring, and collecting and reviewing network traffic.

Azure Network Watcher

Azure Network Watcher can help you troubleshoot, and provides a whole new set of tools to assist with the identification of security issues.

[Security Group View](#) helps with auditing and security compliance of Virtual Machines. Use this feature to perform programmatic audits, comparing the baseline policies defined by your organization to effective rules for each of your VMs. This can help you identify any configuration drift.

[Packet capture](#) allows you to capture network traffic to and from the virtual machine. You can collect network statistics and troubleshoot application issues, which can be invaluable in the investigation of network intrusions. You can also use this feature together with Azure Functions to start network captures in response to specific Azure alerts.

For more information on Network Watcher and how to start testing some of the functionality in your labs, see [Azure network watcher monitoring overview](#).

NOTE

For the most up-to-date notifications on availability and status of this service, check the [Azure updates page](#).

Azure Security Center

Azure Security Center helps you prevent, detect, and respond to threats, and provides you increased visibility into, and control over, the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a large set of security solutions.

Security Center helps you optimize and monitor network security by:

- Providing network security recommendations.
- Monitoring the state of your network security configuration.
- Alerting you to network based threats, both at the endpoint and network levels.

Learn more:

- [Introduction to Azure Security Center](#)

Virtual Network TAP

Azure virtual network TAP (Terminal Access Point) allows you to continuously stream your virtual machine network traffic to a network packet collector or analytics tool. The collector or analytics tool is provided by a network virtual appliance partner. You can use the same virtual network TAP resource to aggregate traffic from multiple network interfaces in the same or different subscriptions.

Learn more:

- [Virtual network TAP](#)

Logging

Logging at a network level is a key function for any network security scenario. In Azure, you can log information obtained for NSGs to get network level logging information. With NSG logging, you get information from:

- **Activity logs.** Use these logs to view all operations submitted to your Azure subscriptions. These logs are enabled by default, and can be used within the Azure portal. They were previously known as audit or operational logs.
- Event logs. These logs provide information about what NSG rules were applied.
- Counter logs. These logs let you know how many times each NSG rule was applied to deny or allow traffic.

You can also use [Microsoft Power BI](#), a powerful data visualization tool, to view and analyze these logs. Learn more:

- [Log Analytics for Network Security Groups \(NSGs\)](#)

Azure Network Security Best Practices

9/18/2018 • 13 minutes to read • [Edit Online](#)

You can connect [Azure virtual machines \(VMs\)](#) and appliances to other networked devices by placing them on [Azure virtual networks](#). That is, you can connect virtual network interface cards to a virtual network to allow TCP/IP-based communications between network-enabled devices. Virtual machines connected to an Azure virtual network can connect to devices on the same virtual network, different virtual networks, the internet, or your own on-premises networks.

This article discusses a collection of Azure network security best practices. These best practices are derived from our experience with Azure networking and the experiences of customers like yourself.

For each best practice, this article explains:

- What the best practice is
- Why you want to enable that best practice
- What might be the result if you fail to enable the best practice
- Possible alternatives to the best practice
- How you can learn to enable the best practice

This Azure Network Security Best Practices article is based on a consensus opinion, and Azure platform capabilities and feature sets, as they exist at the time this article was written. Opinions and technologies change over time and this article will be updated on a regular basis to reflect those changes.

The following sections describe best practices for network security.

Logically segment subnets

Azure virtual networks are similar to a LAN on your on-premises network. The idea behind an Azure virtual network is that you create a single private IP address space-based network on which you can place all your Azure virtual machines. The private IP address spaces available are in the Class A (10.0.0.0/8), Class B (172.16.0.0/12), and Class C (192.168.0.0/16) ranges.

Best practices for logically segmenting subnets include:

Best practice: Segment the larger address space into subnets.

Detail: Use [CIDR](#)-based subnetting principles to create your subnets.

Best practice: Create network access controls between subnets. Routing between subnets happens automatically, and you don't need to manually configure routing tables. By default, there are no network access controls between the subnets that you create on the Azure virtual network.

Detail: Use a [network security group](#) (NSG). NSGs are simple, stateful packet inspection devices that use the 5-tuple (the source IP, source port, destination IP, destination port, and layer 4 protocol) approach to create allow/deny rules for network traffic. You allow or deny traffic to and from a single IP address, to and from multiple IP addresses, or to and from entire subnets.

When you use NSGs for network access control between subnets, you can put resources that belong to the same security zone or role in their own subnets.

Control routing behavior

When you put a virtual machine on an Azure virtual network, the VM can connect to any other VM on the same

virtual network, even if the other VMs are on different subnets. This is possible because a collection of system routes enabled by default allows this type of communication. These default routes allow VMs on the same virtual network to initiate connections with each other, and with the internet (for outbound communications to the internet only).

Although the default system routes are useful for many deployment scenarios, there are times when you want to customize the routing configuration for your deployments. You can configure the next-hop address to reach specific destinations.

We recommend that you configure [user-defined routes](#) when you deploy a security appliance for a virtual network. We talk about this in a later section titled [secure your critical Azure service resources to only your virtual networks](#).

NOTE

User-defined routes are not required, and the default system routes usually work.

Enable forced tunneling

To better understand forced tunneling, it's useful to understand what "split tunneling" is. The most common example of split tunneling is seen with virtual private network (VPN) connections. Imagine that you establish a VPN connection from your hotel room to your corporate network. This connection allows you to access corporate resources. All communications to your corporate network go through the VPN tunnel.

What happens when you want to connect to resources on the internet? When split tunneling is enabled, those connections go directly to the internet and not through the VPN tunnel. Some security experts consider this to be a potential risk. They recommend disabling split tunneling and making sure that all connections, those destined for the internet and those destined for corporate resources, go through the VPN tunnel. The advantage of disabling split tunneling is that connections to the internet are then forced through the corporate network's security devices. That wouldn't be the case if the VPN client connected to the internet outside the VPN tunnel.

Now let's bring this back to VMs on an Azure virtual network. The default routes for an Azure virtual network allow VMs to initiate traffic to the internet. This too can represent a security risk, because these outbound connections might increase the attack surface of a VM and be used by attackers. For this reason, we recommend that you [enable forced tunneling](#) on your VMs when you have cross-premises connectivity between your Azure virtual network and your on-premises network. We talk about cross-premises connectivity later in the networking best practices.

If you don't have a cross-premises connection, be sure to take advantage of NSGs (discussed earlier) or Azure virtual network security appliances (discussed next) to prevent outbound connections to the internet from your Azure virtual machines.

Use virtual network appliances

NSGs and user-defined routing can provide a certain measure of network security at the network and transport layers of the [OSI model](#). But in some situations, you want or need to enable security at high levels of the stack. In such situations, we recommend that you deploy virtual network security appliances provided by Azure partners.

Azure network security appliances can deliver better security than what network-level controls provide. Network security capabilities of virtual network security appliances include:

- Firewalling
- Intrusion detection/intrusion prevention
- Vulnerability management
- Application control
- Network-based anomaly detection

- Web filtering
- Antivirus
- Botnet protection

To find available Azure virtual network security appliances, go to the [Azure Marketplace](#) and search for "security" and "network security."

Deploy perimeter networks for security zones

A [perimeter network](#) (also known as a DMZ) is a physical or logical network segment that provides an additional layer of security between your assets and the internet. Specialized network access control devices on the edge of a perimeter network allow only desired traffic into your virtual network.

Perimeter networks are useful because you can focus your network access control management, monitoring, logging, and reporting on the devices at the edge of your Azure virtual network. Here you typically enable distributed denial of service (DDoS) prevention, intrusion detection/intrusion prevention systems (IDS/IPS), firewall rules and policies, web filtering, network antimalware, and more. The network security devices sit between the internet and your Azure virtual network and have an interface on both networks.

Although this is the basic design of a perimeter network, there are many different designs, such as back-to-back, tri-homed, and multi-homed.

We recommend for all high security deployments that you consider using a perimeter network to enhance the level of network security for your Azure resources.

Avoid exposure to the Internet with dedicated WAN links

Many organizations have chosen the hybrid IT route. In hybrid IT, some of the company's information assets are in Azure, while others remain on-premises. In many cases, some components of a service are running in Azure while other components remain on-premises.

In the hybrid IT scenario, there is usually some type of cross-premises connectivity. Cross-premises connectivity allows the company to connect its on-premises networks to Azure virtual networks. Two cross-premises connectivity solutions are available:

- **Site-to-site VPN:** It's a trusted, reliable, and established technology, but the connection takes place over the internet. Bandwidth is constrained to a maximum of about 200 Mbps. Site-to-site VPN is a desirable option in some scenarios and is discussed further in the section [Disable RDP/SSH access to virtual machines](#).
- **Azure ExpressRoute:** We recommend that you use [ExpressRoute](#) for your cross-premises connectivity. ExpressRoute is a dedicated WAN link between your on-premises location or an Exchange hosting provider. Because this is a telco connection, your data doesn't travel over the internet and therefore is not exposed to the potential risks of internet communications.

Optimize uptime and performance

If a service is down, information can't be accessed. If performance is so poor that the data is unusable, you can consider the data to be inaccessible. From a security perspective, you need to do whatever you can to make sure that your services have optimal uptime and performance.

A popular and effective method for enhancing availability and performance is load balancing. Load balancing is a method of distributing network traffic across servers that are part of a service. For example, if you have front-end web servers as part of your service, you can use load balancing to distribute the traffic across your multiple front-end web servers.

This distribution of traffic increases availability because if one of the web servers becomes unavailable, the load

balancer stops sending traffic to that server and redirects it to the servers that are still online. Load balancing also helps performance, because the processor, network, and memory overhead for serving requests is distributed across all the load-balanced servers.

We recommend that you employ load balancing whenever you can, and as appropriate for your services. Following are scenarios at both the Azure virtual network level and the global level, along with load-balancing options for each.

Scenario: You have an application that:

- Requires requests from the same user/client session to reach the same back-end virtual machine. Examples of this are shopping cart apps and web mail servers.
- Accepts only a secure connection, so unencrypted communication to the server is not an acceptable option.
- Requires multiple HTTP requests on the same long-running TCP connection to be routed or load balanced to different back-end servers.

Load-balancing option: Use [Azure Application Gateway](#), an HTTP web traffic load balancer. Application Gateway supports end-to-end SSL encryption and [SSL termination](#) at the gateway. Web servers can then be unburdened from encryption and decryption overhead and traffic flowing unencrypted to the back-end servers.

Scenario: You need to load balance incoming connections from the internet among your servers located in an Azure virtual network. Scenarios are when you:

- Have stateless applications that accept incoming requests from the internet.
- Don't require sticky sessions or SSL offload. Sticky sessions is a method used with Application Load Balancing, to achieve server-affinity.

Load-balancing option: Use the Azure portal to [create an external load balancer](#) that spreads incoming requests across multiple VMs to provide a higher level of availability.

Scenario: You need to load balance connections from VMs that are not on the internet. In most cases, the connections that are accepted for load balancing are initiated by devices on an Azure virtual network, such as SQL Server instances or internal web servers.

Load-balancing option: Use the Azure portal to [create an internal load balancer](#) that spreads incoming requests across multiple VMs to provide a higher level of availability.

Scenario: You need global load balancing because you:

- Have a cloud solution that is widely distributed across multiple regions and requires the highest level of uptime (availability) possible.
- Need the highest level of uptime possible to make sure that your service is available even if an entire datacenter becomes unavailable.

Load-balancing option: Use Azure Traffic Manager. Traffic Manager makes it possible to load balance connections to your services based on the location of the user.

For example, if the user makes a request to your service from the EU, the connection is directed to your services located in an EU datacenter. This part of Traffic Manager global load balancing helps to improve performance because connecting to the nearest datacenter is faster than connecting to datacenters that are far away.

Disable RDP/SSH Access to virtual machines

It's possible to reach Azure virtual machines by using [Remote Desktop Protocol](#) (RDP) and the [Secure Shell](#) (SSH) protocol. These protocols enable the management VMs from remote locations and are standard in datacenter computing.

The potential security problem with using these protocols over the internet is that attackers can use [brute force](#)

techniques to gain access to Azure virtual machines. After the attackers gain access, they can use your VM as a launch point for compromising other machines on your virtual network or even attack networked devices outside Azure.

We recommend that you disable direct RDP and SSH access to your Azure virtual machines from the internet. After direct RDP and SSH access from the internet is disabled, you have other options that you can use to access these VMs for remote management.

Scenario: Enable a single user to connect to an Azure virtual network over the internet.

Option: [Point-to-site VPN](#) is another term for a remote access VPN client/server connection. After the point-to-site connection is established, the user can use RDP or SSH to connect to any VMs located on the Azure virtual network that the user connected to via point-to-site VPN. This assumes that the user is authorized to reach those VMs.

Point-to-site VPN is more secure than direct RDP or SSH connections because the user has to authenticate twice before connecting to a VM. First, the user needs to authenticate (and be authorized) to establish the point-to-site VPN connection. Second, the user needs to authenticate (and be authorized) to establish the RDP or SSH session.

Scenario: Enable users on your on-premises network to connect to VMs on your Azure virtual network.

Option: A [site-to-site VPN](#) connects an entire network to another network over the internet. You can use a site-to-site VPN to connect your on-premises network to an Azure virtual network. Users on your on-premises network connect by using the RDP or SSH protocol over the site-to-site VPN connection. You don't have to allow direct RDP or SSH access over the internet.

Scenario: Use a dedicated WAN link to provide functionality similar to the site-to-site VPN.

Option: Use [ExpressRoute](#). It provides functionality similar to the site-to-site VPN. The main differences are:

- The dedicated WAN link doesn't traverse the internet.
- Dedicated WAN links are typically more stable and perform better.

Secure your critical Azure service resources to only your virtual networks

Use virtual network service endpoints to extend your virtual network private address space, and the identity of your virtual network to the Azure services, over a direct connection. Endpoints allow you to secure your critical Azure service resources to only your virtual networks. Traffic from your virtual network to the Azure service always remains on the Microsoft Azure backbone network.

Service endpoints provide the following benefits:

- **Improved security for your Azure service resources:** With service endpoints, Azure service resources can be secured to your virtual network. Securing service resources to a virtual network provides improved security by fully removing public internet access to resources, and allowing traffic only from your virtual network.
- **Optimal routing for Azure service traffic from your virtual network:** Any routes in your virtual network that force internet traffic to your on-premises and/or virtual appliances, known as forced tunneling, also force Azure service traffic to take the same route as the internet traffic. Service endpoints provide optimal routing for Azure traffic.

Endpoints always take service traffic directly from your virtual network to the service on the Azure backbone network. Keeping traffic on the Azure backbone network allows you to continue auditing and monitoring outbound internet traffic from your virtual networks, through forced tunneling, without affecting service traffic. Learn more about [user-defined routes and forced tunneling](#).

- **Simple to set up with less management overhead:** You no longer need reserved, public IP addresses in your virtual networks to secure Azure resources through an IP firewall. There are no NAT or gateway devices required to set up the service endpoints. Service endpoints are configured through a simple click on

a subnet. There is no additional overhead to maintain the endpoints.

To learn more about service endpoints and the Azure services and regions that service endpoints are available for, see [Virtual network service endpoints](#).

Next step

See [Azure security best practices and patterns](#) for more security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.

Azure DDoS Protection: Best practices and reference architectures

11/7/2018 • 17 minutes to read • [Edit Online](#)

This article is for IT decision makers and security personnel. It expects that you're familiar with Azure, networking, and security.

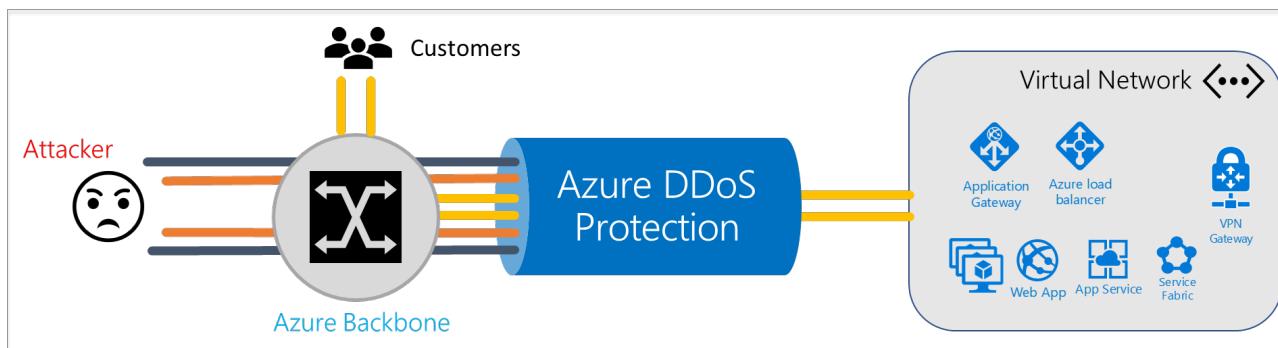
Designing for distributed denial of service (DDoS) resiliency requires planning and designing for a variety of failure modes. This article provides best practices for designing applications in Azure for resiliency against DDoS attacks.

Types of attacks

DDoS is a type of attack that tries to exhaust application resources. The goal is to affect the application's availability and its ability to handle legitimate requests. Attacks are becoming more sophisticated and larger in size and impact. DDoS attacks can be targeted at any endpoint that is publicly reachable through the internet.

Azure provides continuous protection against DDoS attacks. This protection is integrated into the Azure platform by default and at no extra cost.

In addition to the core DDoS protection in the platform, [Azure DDoS Protection Standard](#) provides advanced DDoS mitigation capabilities against network attacks. It's automatically tuned to protect your specific Azure resources. Protection is simple to enable during the creation of new virtual networks. It can also be done after creation and requires no application or resource changes.



DDoS attacks can be classified into three categories: volumetric, protocol, and resource.

Volumetric attacks

Volumetric attacks are the most common type of DDoS attack. Volumetric attacks are brute-force assaults that target the network and transport layers. They try to exhaust resources like network links.

These attacks often use multiple infected systems to flood the network layers with seemingly legitimate traffic. They use network-layer protocols such as Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), and Transmission Control Protocol (TCP).

The most commonly used network-layer DDoS attacks are TCP SYN flooding, ICMP echo, UDP flooding, DNS, and NTP amplification attacks. This type of attack can be used not only to disrupt service, but also as a smokescreen for more nefarious and targeted network intrusion. An example of a recent volumetric attack is the [Memcached exploit](#) that affected GitHub. This attack targeted UDP port 11211 and generated 1.35 Tb/s of attack volume.

Protocol attacks

Protocol attacks target application protocols. They try to use up all the available resources in infrastructure devices such as firewalls, application servers, and load balancers. Protocol attacks use packets that are malformed or contain protocol abnormalities. These attacks operate by sending large numbers of open requests that servers and other communication devices answer and wait for a packet response. The target tries to respond to the open requests, eventually causing the system to crash.

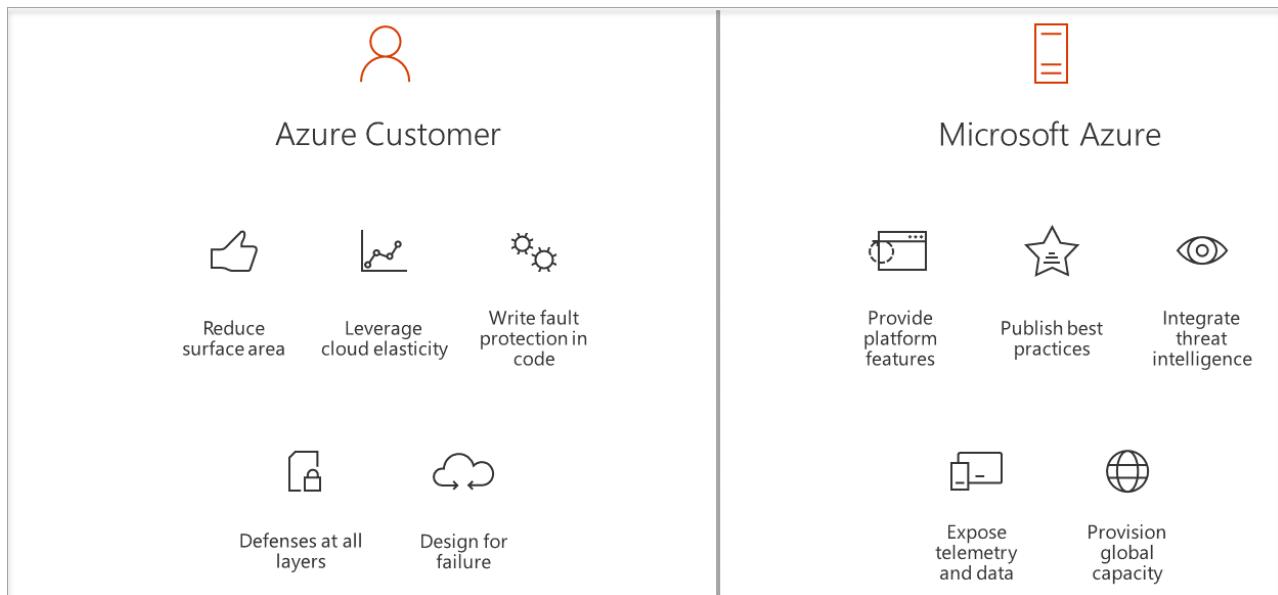
The most common example of a protocol-based DDoS attack is TCP SYN flood. In this attack, a succession of TCP SYN requests tries to overwhelm a target. The goal is to make the target unresponsive. The 2016 Dyn outage, apart from being an application-layer attack, consisted of TCP SYN floods that targeted port 53 of Dyn's DNS servers.

Resource attacks

Resource attacks target the application layer. They trigger back-end processes in an effort to overwhelm a system. Resource attacks abuse traffic that looks normal but that carries CPU-intensive queries to the server. The volume of traffic needed to exhaust resources is lower than that of the other type of attacks. The traffic in a resource attack is indistinguishable from legitimate traffic, making it hard to detect. The most common resource attacks are on HTTP/HTTPS and DNS services.

Shared responsibility in the cloud

A defense-in-depth strategy helps combat the increasing variety and sophistication of attacks. Security is a shared responsibility between the customer and Microsoft. Microsoft calls this a [shared responsibility model](#). The following figure shows this division of responsibility:



Azure customers benefit from reviewing Microsoft best practices and building globally distributed applications that are designed and tested for failure.

Fundamental best practices

The following sections give prescriptive guidance to build DDoS-resilient services on Azure.

Design for security

Ensure that security is a priority throughout the entire lifecycle of an application, from design and implementation to deployment and operations. Applications can have bugs that allow a relatively low volume of requests to use an inordinate amount of resources, resulting in a service outage.

To help protect a service running on Microsoft Azure, you should have a good understanding of your application architecture and focus on the [five pillars of software quality](#). You should know typical traffic volumes, the

connectivity model between the application and other applications, and the service endpoints that are exposed to the public internet.

Ensuring that an application is resilient enough to handle a denial of service that's targeted at the application itself is most important. Security and privacy are built into the Azure platform, beginning with the [Security Development Lifecycle \(SDL\)](#). The SDL addresses security at every development phase and ensures that Azure is continually updated to make it even more secure.

Design for scalability

Scalability is how well a system can handle increased load. You must design your applications to [scale horizontally](#) to meet the demand of an amplified load, specifically in the event of a DDoS attack. If your application depends on a single instance of a service, it creates a single point of failure. Provisioning multiple instances makes your system more resilient and more scalable.

For [Azure App Service](#), select an [App Service plan](#) that offers multiple instances. For Azure Cloud Services, configure each of your roles to use [multiple instances](#). For [Azure Virtual Machines](#), ensure that your virtual machine (VM) architecture includes more than one VM and that each VM is included in an [availability set](#). We recommend using [virtual machine scale sets](#) for autoscaling capabilities.

Defense in depth

The idea behind defense in depth is to manage risk by using diverse defensive strategies. Layering security defenses in an application reduces the chance of a successful attack. We recommend that you implement secure designs for your applications by using the built-in capabilities of the Azure platform.

For example, the risk of attack increases with the size (*surface area*) of the application. You can reduce the surface area by using whitelisting to close down the exposed IP address space and listening ports that are not needed on the load balancers ([Azure Load Balancer](#) and [Azure Application Gateway](#)). [Network security groups \(NSGs\)](#) are another way to reduce the attack surface. You can use [service tags](#) and [application security groups](#) to minimize complexity for creating security rules and configuring network security, as a natural extension of an application's structure.

You should deploy Azure services in a [virtual network](#) whenever possible. This practice allows service resources to communicate through private IP addresses. Azure service traffic from a virtual network uses public IP addresses as source IP addresses by default. Using [service endpoints](#) will switch service traffic to use virtual network private addresses as the source IP addresses when they're accessing the Azure service from a virtual network.

We often see customers' on-premises resources getting attacked along with their resources in Azure. If you're connecting an on-premises environment to Azure, we recommend that you minimize exposure of on-premises resources to the public internet. You can use the scale and advanced DDoS protection capabilities of Azure by deploying your well-known public entities in Azure. Because these publicly accessible entities are often a target for DDoS attacks, putting them in Azure reduces the impact on your on-premises resources.

Azure offerings for DDoS protection

Azure has two DDoS service offerings that provide protection from network attacks (Layer 3 and 4): DDoS Protection Basic and DDoS Protection Standard.

DDoS Protection Basic

Basic protection is integrated into the Azure by default at no additional cost. The scale and capacity of the globally deployed Azure network provides defense against common network-layer attacks through always-on traffic monitoring and real-time mitigation. DDoS Protection Basic requires no user configuration or application changes. DDoS Protection Basic helps protect all Azure services, including PaaS services like Azure DNS.



Basic DDoS protection in Azure consists of both software and hardware components. A software control plane decides when, where, and what type of traffic should be steered through hardware appliances that analyze and remove attack traffic. The control plane makes this decision based on an infrastructure-wide DDoS Protection *policy*. This policy is statically set and universally applied to all Azure customers.

For example, the DDoS Protection policy specifies at what traffic volume the protection should be *triggered*. (That is, the tenant's traffic should be routed through scrubbing appliances.) The policy then specifies how the scrubbing appliances should *mitigate* the attack.

The Azure DDoS Protection Basic service is targeted at protection of the infrastructure and protection of the Azure platform. It mitigates traffic when it exceeds a rate that is so significant that it might affect multiple customers in a multitenant environment. It doesn't provide alerting or per-customer customized policies.

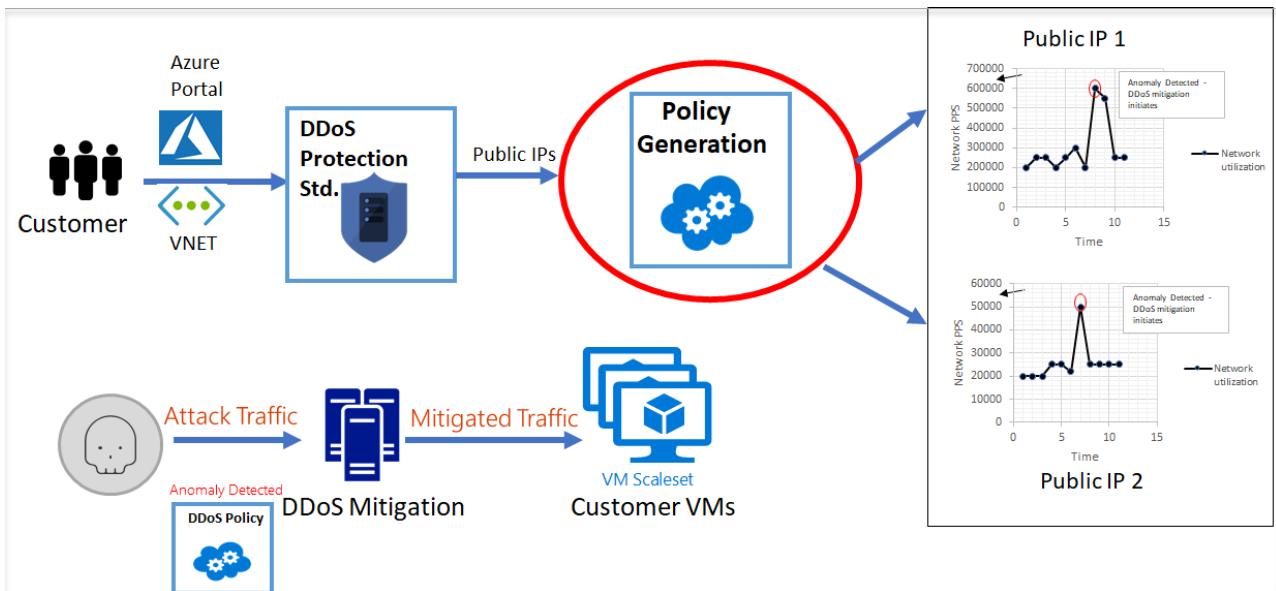
DDoS Protection Standard

Standard protection provides enhanced DDoS mitigation features. It's automatically tuned to help protect your specific Azure resources in a virtual network. Protection is simple to enable on any new or existing virtual network, and it requires no application or resource changes. It has several advantages over the basic service, including logging, alerting, and telemetry. The following sections outline the key features of the Azure DDoS Protection Standard service.

Adaptive real-time tuning

The Azure DDoS Protection Basic service helps protect customers and prevent impacts to other customers. For example, if a service is provisioned for a typical volume of legitimate incoming traffic that's smaller than the *trigger rate* of the infrastructure-wide DDoS Protection policy, a DDoS attack on that customer's resources might go unnoticed. More generally, the complexity of recent attacks (for example, multi-vector DDoS) and the application-specific behaviors of tenants call for per-customer, customized protection policies. The service accomplishes this customization by using two insights:

- Automatic learning of per-customer (per-IP) traffic patterns for Layer 3 and 4.
- Minimizing false positives, considering that the scale of Azure allows it to absorb a significant amount of traffic.



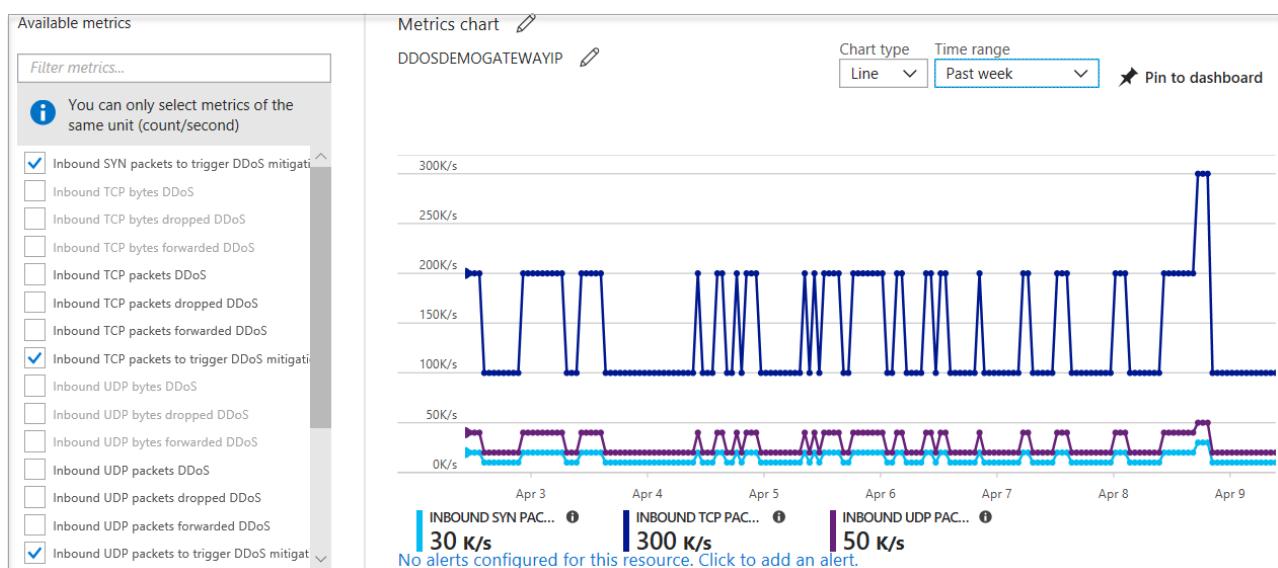
DDoS Protection telemetry, monitoring, and alerting

DDoS Protection Standard exposes rich telemetry via [Azure Monitor](#) for the duration of a DDoS attack. You can configure alerts for any of the Azure Monitor metrics that DDoS Protection uses. You can integrate logging with Splunk (Azure Event Hubs), Azure Log Analytics, and Azure Storage for advanced analysis via the Azure Monitor Diagnostics interface.

DDoS mitigation policies

In the Azure portal, select **Monitor > Metrics**. In the **Metrics** pane, select the resource group, select a resource type of **Public IP Address**, and select your Azure public IP address. DDoS metrics are visible in the **Available metrics** pane.

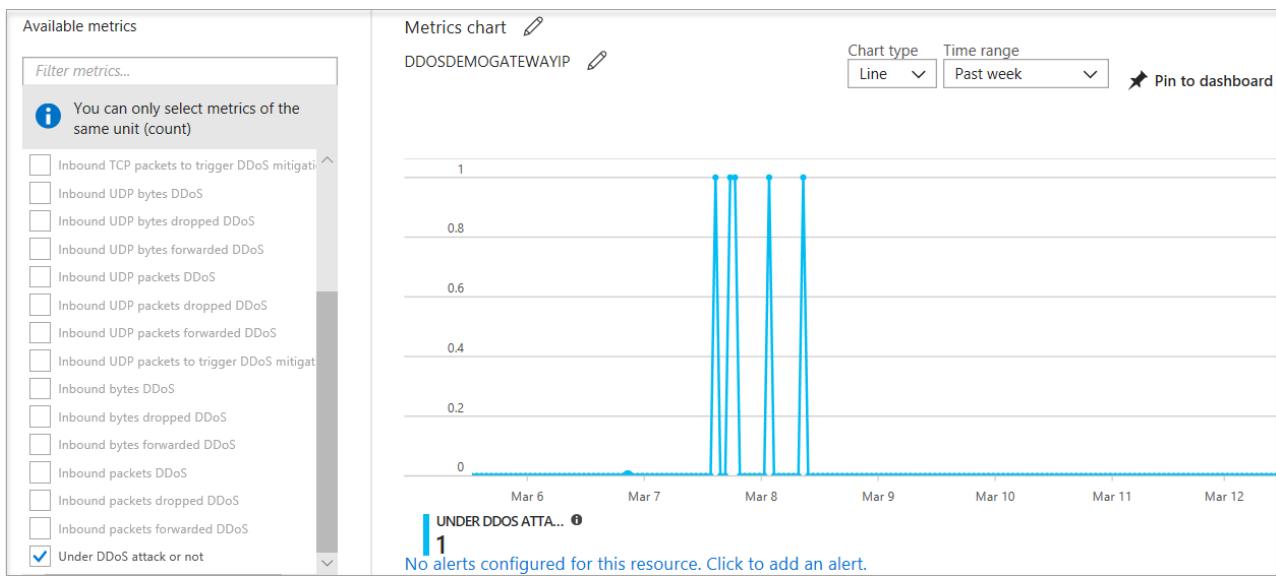
DDoS Protection Standard applies three autotuned mitigation policies (TCP SYN, TCP, and UDP) for each public IP of the protected resource, in the virtual network that has DDoS enabled. You can view the policy thresholds by selecting the metric **Inbound packets to trigger DDoS mitigation**.



The policy thresholds are autoconfigured via machine learning-based network traffic profiling. DDoS mitigation occurs for an IP address under attack only when the policy threshold is exceeded.

Metric for an IP address under DDoS attack

If the public IP address is under attack, the value for the metric **Under DDoS attack or not** changes to 1 as DDoS Protection performs mitigation on the attack traffic.



We recommend configuring an alert on this metric. You'll then be notified when there's an active DDoS mitigation performed on your public IP address.

For more information, see [Manage Azure DDoS Protection Standard using the Azure portal](#).

Web application firewall for resource attacks

Specific to resource attacks at the application layer, you should configure a web application firewall (WAF) to help secure web applications. A WAF inspects inbound web traffic to block SQL injections, cross-site scripting, DDoS, and other Layer 7 attacks. Azure provides [WAF as a feature of Application Gateway](#) for centralized protection of your web applications from common exploits and vulnerabilities. There are other WAF offerings available from Azure partners that might be more suitable for your needs via the [Azure Marketplace](#).

Even web application firewalls are susceptible to volumetric and state exhaustion attacks. We strongly recommend enabling DDoS Protection Standard on the WAF virtual network to help protect from volumetric and protocol attacks. For more information, see the [DDoS Protection reference architectures](#) section.

Protection planning

Planning and preparation are crucial to understand how a system will perform during a DDoS attack. Designing an incident management response plan is part of this effort.

If you have DDoS Protection Standard, make sure that it's enabled on the virtual network of internet-facing endpoints. Configuring DDoS alerts helps you constantly watch for any potential attacks on your infrastructure.

You should monitor your applications independently. Understand the normal behavior of an application. Prepare to act if the application is not behaving as expected during a DDoS attack.

Testing through simulations

It's a good practice to test your assumptions about how your services will respond to an attack by conducting periodic simulations. During testing, validate that your services or applications continue to function as expected and there's no disruption to the user experience. Identify gaps from both a technology and process standpoint and incorporate them in the DDoS response strategy. We recommend that you perform such tests in staging environments or during non-peak hours to minimize the impact to the production environment.

We have partnered with [BreakingPoint Cloud](#) to build an interface where Azure customers can generate traffic against DDoS Protection-enabled public endpoints for simulations. You can use the [BreakingPoint Cloud](#) simulation to:

- Validate how Azure DDoS Protection helps protect your Azure resources from DDoS attacks.
- Optimize your incident response process while under DDoS attack.
- Document DDoS compliance.

- Train your network security teams.

Cybersecurity requires constant innovation in defense. Azure DDoS Standard protection is a state-of-the-art offering with an effective solution to mitigate increasingly complex DDoS attacks.

Components of a DDoS response strategy

A DDoS attack that targets Azure resources usually requires minimal intervention from a user standpoint. Still, incorporating DDoS mitigation as part of an incident response strategy helps minimize the impact to business continuity.

Microsoft threat intelligence

Microsoft has an extensive threat intelligence network. This network uses the collective knowledge of an extended security community that supports Microsoft online services, Microsoft partners, and relationships within the internet security community.

As a critical infrastructure provider, Microsoft receives early warnings about threats. Microsoft gathers threat intelligence from its online services and from its global customer base. Microsoft incorporates all of this threat intelligence back into the Azure DDoS Protection products.

Also, the Microsoft Digital Crimes Unit (DCU) performs offensive strategies against botnets. Botnets are a common source of command and control for DDoS attacks.

Risk evaluation of your Azure resources

It's imperative to understand the scope of your risk from a DDoS attack on an ongoing basis. Periodically ask yourself:

- What new publicly available Azure resources need protection?
- Is there a single point of failure in the service?
- How can services be isolated to limit the impact of an attack while still making services available to valid customers?
- Are there virtual networks where DDoS Protection Standard should be enabled but isn't?
- Are my services active/active with failover across multiple regions?

Customer DDoS response team

Creating a DDoS response team is a key step in responding to an attack quickly and effectively. Identify contacts in your organization who will oversee both planning and execution. This DDoS response team should thoroughly understand the Azure DDoS Protection Standard service. Make sure that the team can identify and mitigate an attack by coordinating with internal and external customers, including the Microsoft support team.

For your DDoS response team, we recommend that you use simulation exercises as a normal part of your service availability and continuity planning. These exercises should include scale testing.

Alerts during an attack

Azure DDoS Protection Standard identifies and mitigates DDoS attacks without any user intervention. To get notified when there's an active mitigation for a protected public IP, you can [configure an alert](#) on the metric **Under DDoS attack or not**. You can choose to create alerts for the other DDoS metrics to understand the scale of the attack, traffic being dropped, and other details.

When to contact Microsoft support

- During a DDoS attack, you find that the performance of the protected resource is severely degraded, or the resource is not available.
- You think the DDoS Protection service is not behaving as expected.

The DDoS Protection service starts mitigation only if the metric value **Policy to trigger DDoS mitigation (TCP/TCP SYN/UDP)** is lower than the traffic received on the protected public IP resource.

- You're planning a viral event that will significantly increase your network traffic.
- An actor has threatened to launch a DDoS attack against your resources.

For attacks that have a critical business impact, create a severity-A [support ticket](#).

Post-attack steps

It's always a good strategy to do a postmortem after an attack and adjust the DDoS response strategy as needed.

Things to consider:

- Was there any disruption to the service or user experience due to lack of scalable architecture?
- Which applications or services suffered the most?
- How effective was the DDoS response strategy, and how can it be improved?

If you suspect you're under a DDoS attack, escalate through your normal Azure Support channels.

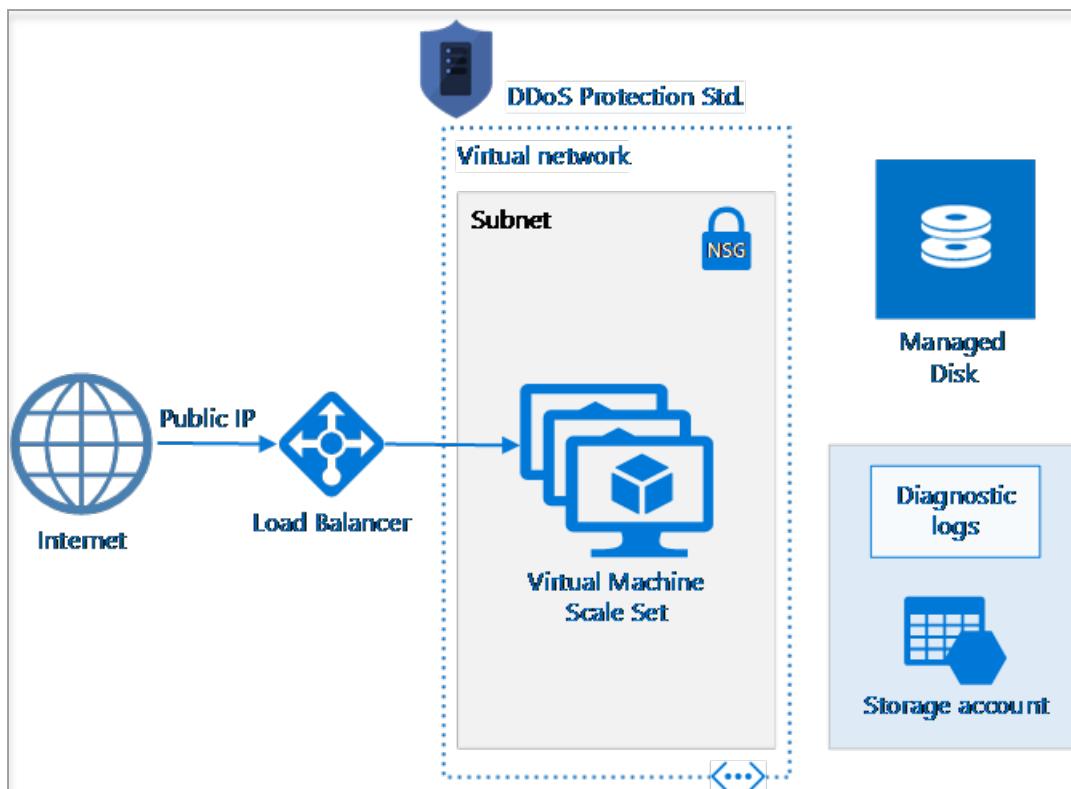
DDoS Protection reference architectures

DDoS Protection Standard is designed [for services that are deployed in a virtual network](#). For other services, the default DDoS Protection Basic service applies. The following reference architectures are arranged by scenarios, with architecture patterns grouped together.

Virtual machine (Windows/Linux) workloads

Application running on load-balanced VMs

This reference architecture shows a set of proven practices for running multiple Windows VMs in a scale set behind a load balancer, to improve availability and scalability. This architecture can be used for any stateless workload, such as a web server.

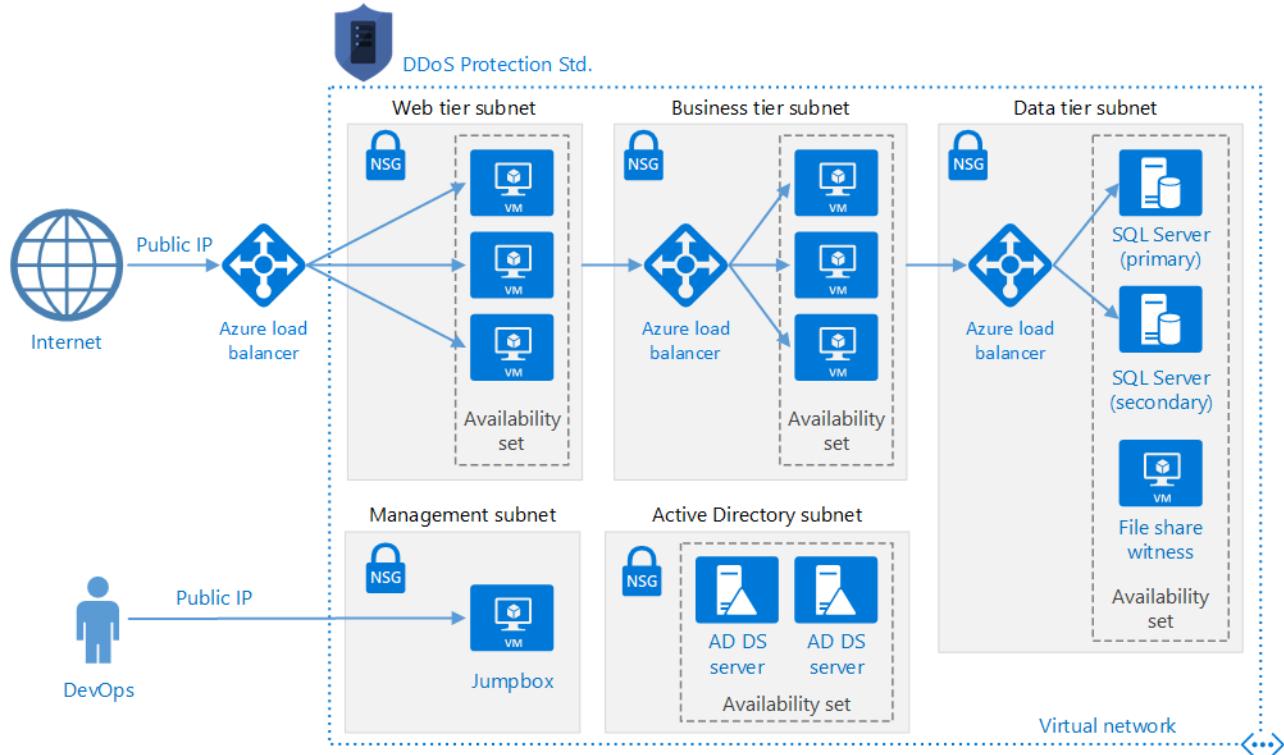


In this architecture, a workload is distributed across multiple VM instances. There is a single public IP address, and internet traffic is distributed to the VMs through a load balancer. DDoS Protection Standard is enabled on the virtual network of the Azure (internet) load balancer that has the public IP associated with it.

The load balancer distributes incoming internet requests to the VM instances. Virtual machine scale sets allow the number of VMs to be scaled in or out manually, or automatically based on predefined rules. This is important if the resource is under DDoS attack. For more information on this reference architecture, see [this article](#).

Application running on Windows N-tier

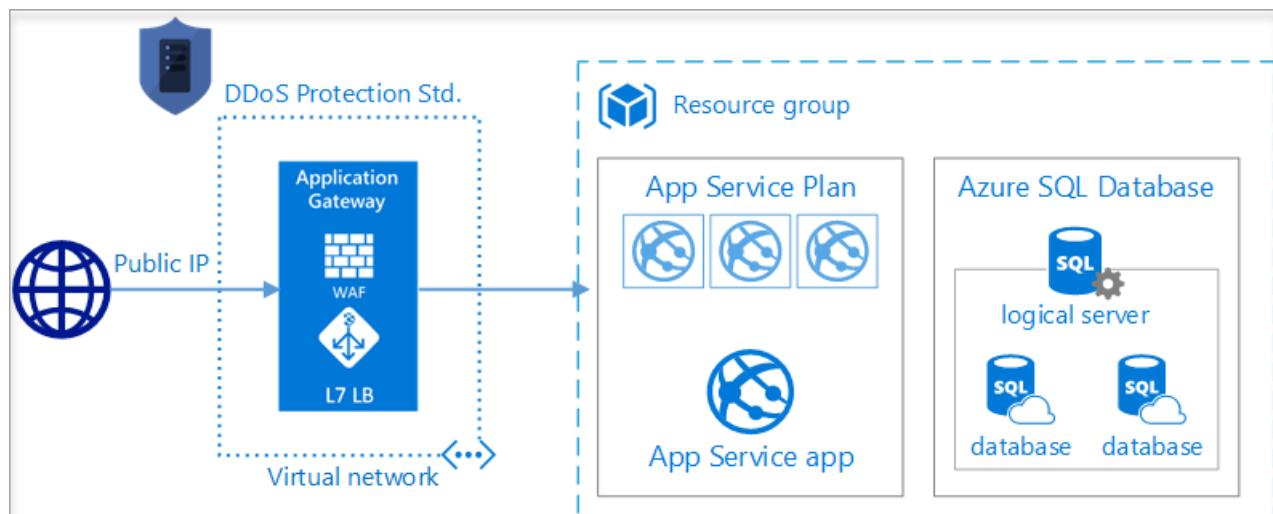
There are many ways to implement an N-tier architecture. The following diagram shows a typical three-tier web application. This architecture builds on the article [Run load-balanced VMs for scalability and availability](#). The web and business tiers use load-balanced VMs.



In this architecture, DDoS Protection Standard is enabled on the virtual network. All public IPs in the virtual network get DDoS protection for Layer 3 and 4. For Layer 7 protection, deploy Application Gateway in the WAF SKU. For more information on this reference architecture, see [this article](#).

PaaS web application

This reference architecture shows running an Azure App Service application in a single region. This architecture shows a set of proven practices for a web application that uses [Azure App Service](#) and [Azure SQL Database](#). A standby region is set up for failover scenarios.



Azure Traffic Manager routes incoming requests to Application Gateway in one of the regions. During normal operations, it routes requests to Application Gateway in the active region. If that region becomes unavailable, Traffic Manager fails over to Application Gateway in the standby region.

All traffic from the internet destined to the web application is routed to the [Application Gateway public IP address](#) via Traffic Manager. In this scenario, the app service (web app) itself is not directly externally facing and is protected by Application Gateway.

We recommend that you configure the Application Gateway WAF SKU (prevent mode) to help protect against Layer 7 (HTTP/HTTPS/WebSocket) attacks. Additionally, web apps are configured to [accept only traffic from the Application Gateway IP address](#).

For more information about this reference architecture, see [this article](#).

Mitigation for non-web PaaS services

HDInsight on Azure

This reference architecture shows configuring DDoS Protection Standard for an [Azure HDInsight cluster](#). Make sure that the HDInsight cluster is linked to a virtual network and that DDoS Protection is enabled on the virtual network.

The screenshot shows the Azure portal interface for creating a new HDInsight cluster. On the left, a sidebar lists five steps: 1. Basics (Configure basic settings), 2. Storage (Set storage settings), 3. Applications (optional) (Productivity through applicatio...), 4. Cluster size (Choose node sizes), and 5. Advanced settings (Configure advanced features). Step 5 is highlighted with a blue background. On the right, the 'Advanced settings' blade is open, containing two main sections: 'Script Actions (optional)' and 'Virtual Network Settings (optional)'. The 'Virtual Network Settings' section includes dropdown menus for 'Virtual network' (set to 'hdinsightvnet/hdinsight') and 'Subnet' (set to 'hdinsight1').

The screenshot shows the Azure portal interface for managing a virtual network. The left sidebar lists various management options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Address space, Connected devices, Subnets, and DDoS protection. The 'DDoS protection' option is highlighted with a blue selection bar at the bottom. The main content area displays information about DDoS protection, including a description of its features and a status summary where 'DDoS protection' is set to 'Enabled'.

In this architecture, traffic destined to the HDInsight cluster from the internet is routed to the public IP associated with the HDInsight gateway load balancer. The gateway load balancer then sends the traffic to the head nodes or the worker nodes directly. Because DDoS Protection Standard is enabled on the HDInsight virtual network, all public IPs in the virtual network get DDoS protection for Layer 3 and 4. This reference architecture can be combined with the N-Tier and multi-region reference architectures.

For more information on this reference architecture, see the [Extend Azure HDInsight using an Azure Virtual Network](#) documentation.

NOTE

Azure App Service Environment for PowerApps or API management in a virtual network with a public IP are both not natively supported.

Next steps

- [Azure DDoS Protection product page](#)
- [Azure DDoS Protection blog](#)
- [Azure DDoS Protection documentation](#)

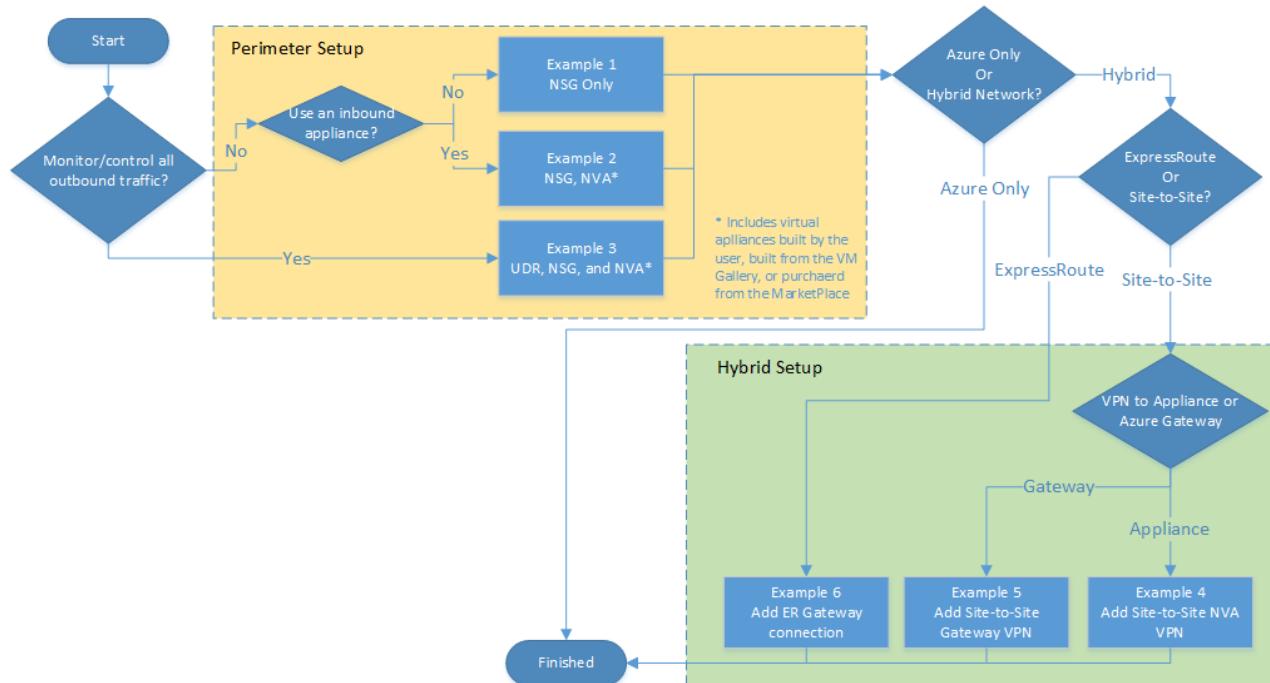
Microsoft cloud services and network security

5/21/2018 • 37 minutes to read • [Edit Online](#)

Microsoft cloud services deliver hyper-scale services and infrastructure, enterprise-grade capabilities, and many choices for hybrid connectivity. Customers can choose to access these services either via the Internet or with Azure ExpressRoute, which provides private network connectivity. The Microsoft Azure platform allows customers to seamlessly extend their infrastructure into the cloud and build multi-tier architectures. Additionally, third parties can enable enhanced capabilities by offering security services and virtual appliances. This white paper provides an overview of security and architectural issues that customers should consider when using Microsoft cloud services accessed via ExpressRoute. It also covers creating more secure services in Azure virtual networks.

Fast start

The following logic chart can direct you to a specific example of the many security techniques available with the Azure platform. For quick reference, find the example that best fits your case. For expanded explanations, continue reading through the paper.



Example 1: Build a perimeter network (also known as DMZ, demilitarized zone, or screened subnet) to help protect applications with network security groups (NSGs).

Example 2: Build a perimeter network to help protect applications with a firewall and NSGs.

Example 3: Build a perimeter network to help protect networks with a firewall, user-defined route (UDR), and NSG.

Example 4: Add a hybrid connection with a site-to-site, virtual appliance virtual private network (VPN).

Example 5: Add a hybrid connection with a site-to-site, Azure VPN gateway.

Example 6: Add a hybrid connection with ExpressRoute.

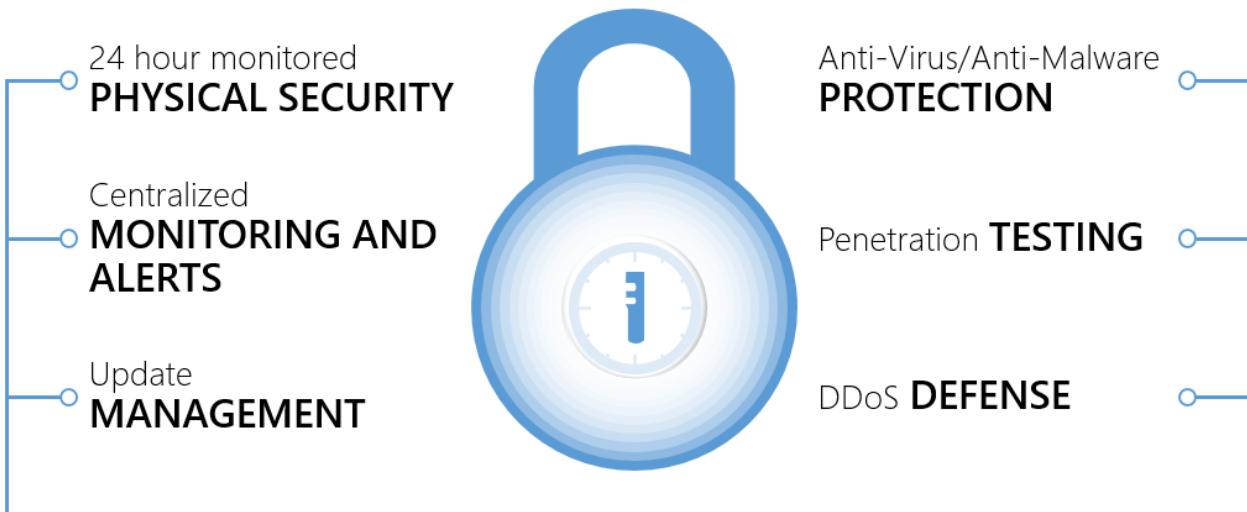
Examples for adding connections between virtual networks, high availability, and service chaining will be added to this document over the next few months.

Microsoft compliance and infrastructure protection

To help organizations comply with national, regional, and industry-specific requirements governing the collection and use of individuals' data, Microsoft offers over 40 certifications and attestations. The most comprehensive set of any cloud service provider.

For more information, see the compliance information on the [Microsoft Trust Center](#).

Microsoft has a comprehensive approach to protect cloud infrastructure needed to run hyper-scale global services. Microsoft cloud infrastructure includes hardware, software, networks, and administrative and operations staff, in addition to the physical data centers.

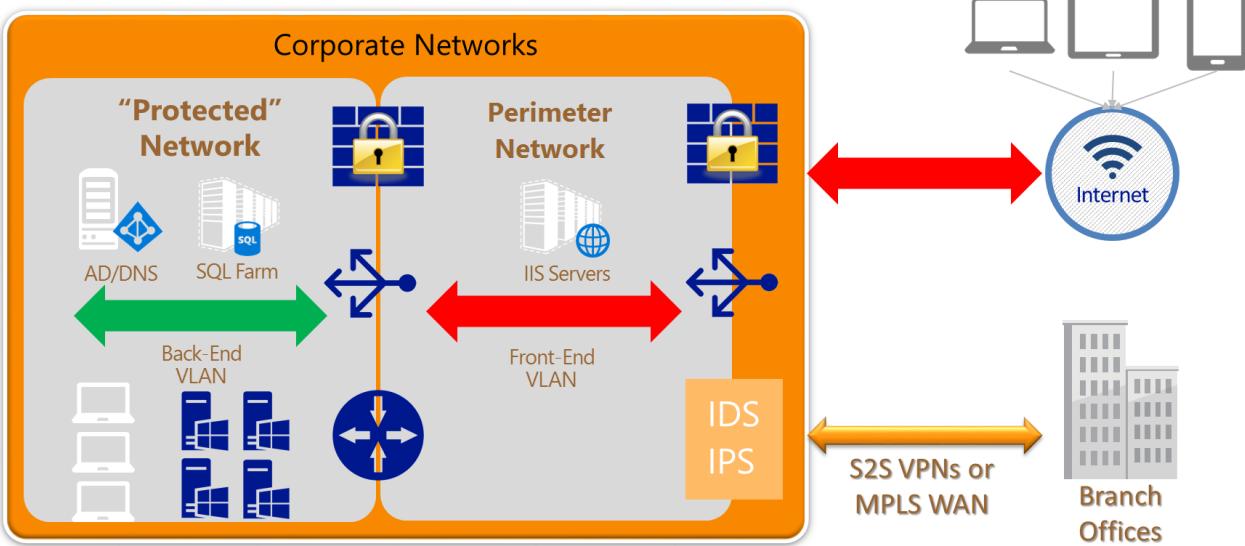


This approach provides a more secure foundation for customers to deploy their services in the Microsoft cloud. The next step is for customers to design and create a security architecture to protect these services.

Traditional security architectures and perimeter networks

Although Microsoft invests heavily in protecting the cloud infrastructure, customers must also protect their cloud services and resource groups. A multilayered approach to security provides the best defense. A perimeter network security zone protects internal network resources from an untrusted network. A perimeter network refers to the edges or parts of the network that sit between the Internet and the protected enterprise IT infrastructure.

In typical enterprise networks, the core infrastructure is heavily fortified at the perimeters, with multiple layers of security devices. The boundary of each layer consists of devices and policy enforcement points. Each layer can include a combination of the following network security devices: firewalls, Denial of Service (DoS) prevention, Intrusion Detection or Protection Systems (IDS/IPS), and VPN devices. Policy enforcement can take the form of firewall policies, access control lists (ACLs), or specific routing. The first line of defense in the network, directly accepting incoming traffic from the Internet, is a combination of these mechanisms to block attacks and harmful traffic while allowing legitimate requests further into the network. This traffic routes directly to resources in the perimeter network. That resource may then "talk" to resources deeper in the network, transiting the next boundary for validation first. The outermost layer is called the perimeter network because this part of the network is exposed to the Internet, usually with some form of protection on both sides. The following figure shows an example of a single subnet perimeter network in a corporate network, with two security boundaries.

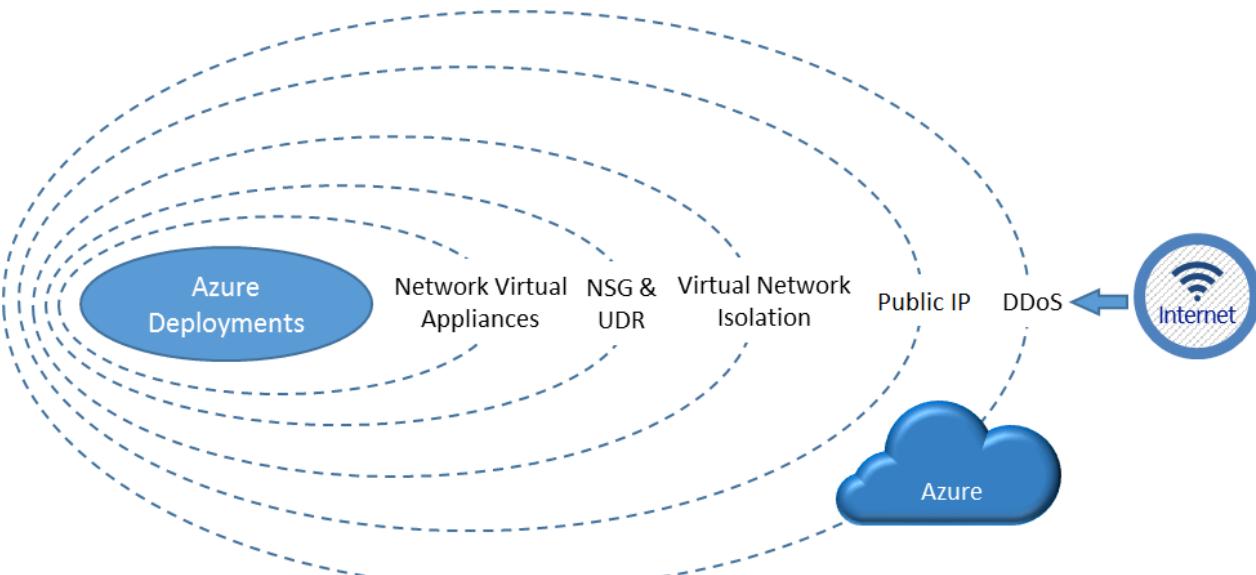


There are many architectures used to implement a perimeter network. These architectures can range from a simple load balancer to a multiple-subnet perimeter network with varied mechanisms at each boundary to block traffic and protect the deeper layers of the corporate network. How the perimeter network is built depends on the specific needs of the organization and its overall risk tolerance.

As customers move their workloads to public clouds, it is critical to support similar capabilities for perimeter network architecture in Azure to meet compliance and security requirements. This document provides guidelines on how customers can build a secure network environment in Azure. It focuses on the perimeter network, but also includes a comprehensive discussion of many aspects of network security. The following questions inform this discussion:

- How can a perimeter network in Azure be built?
- What are some of the Azure features available to build the perimeter network?
- How can back-end workloads be protected?
- How are Internet communications controlled to the workloads in Azure?
- How can the on-premises networks be protected from deployments in Azure?
- When should native Azure security features be used versus third-party appliances or services?

The following diagram shows various layers of security Azure provides to customers. These layers are both native in the Azure platform itself and customer-defined features:



Inbound from the Internet, Azure DDoS helps protect against large-scale attacks against Azure. The next layer is

customer-defined public IP addresses (endpoints), which are used to determine which traffic can pass through the cloud service to the virtual network. Native Azure virtual network isolation ensures complete isolation from all other networks and that traffic only flows through user configured paths and methods. These paths and methods are the next layer, where NSGs, UDR, and network virtual appliances can be used to create security boundaries to protect the application deployments in the protected network.

The next section provides an overview of Azure virtual networks. These virtual networks are created by customers, and are what their deployed workloads are connected to. Virtual networks are the basis of all the network security features required to establish a perimeter network to protect customer deployments in Azure.

Overview of Azure virtual networks

Before Internet traffic can get to the Azure virtual networks, there are two layers of security inherent to the Azure platform:

- 1. DDoS protection:** DDoS protection is a layer of the Azure physical network that protects the Azure platform itself from large-scale Internet-based attacks. These attacks use multiple "bot" nodes in an attempt to overwhelm an Internet service. Azure has a robust DDoS protection mesh on all inbound, outbound, and cross-Azure region connectivity. This DDoS protection layer has no user configurable attributes and is not accessible to the customer. The DDoS protection layer protects Azure as a platform from large-scale attacks, it also monitors out-bound traffic and cross-Azure region traffic. Using network virtual appliances on the VNet, additional layers of resilience can be configured by the customer against a smaller scale attack that doesn't trip the platform level protection. An example of DDoS in action; if an internet facing IP address was attacked by a large-scale DDoS attack, Azure would detect the sources of the attacks and scrub the offending traffic before it reached its intended destination. In almost all cases, the attacked endpoint isn't affected by the attack. In the rare cases that an endpoint is affected, no traffic is affected to other endpoints, only the attacked endpoint. Thus other customers and services would see no impact from that attack. It's critical to note that Azure DDoS is only looking for large-scale attacks. It is possible that your specific service could be overwhelmed before the platform level protection thresholds are exceeded. For example, a web site on a single A0 IIS server, could be taken offline by a DDoS attack before Azure platform level DDoS protection registered a threat.
- 2. Public IP Addresses:** Public IP addresses (enabled via service endpoints, Public IP addresses, Application Gateway, and other Azure features that present a public IP address to the internet routed to your resource) allow cloud services or resource groups to have public Internet IP addresses and ports exposed. The endpoint uses Network Address Translation (NAT) to route traffic to the internal address and port on the Azure virtual network. This path is the primary way for external traffic to pass into the virtual network. The Public IP addresses are configurable to determine which traffic is passed in, and how and where it's translated on to the virtual network.

Once traffic reaches the virtual network, there are many features that come into play. Azure virtual networks are the foundation for customers to attach their workloads and where basic network-level security applies. It is a private network (a virtual network overlay) in Azure for customers with the following features and characteristics:

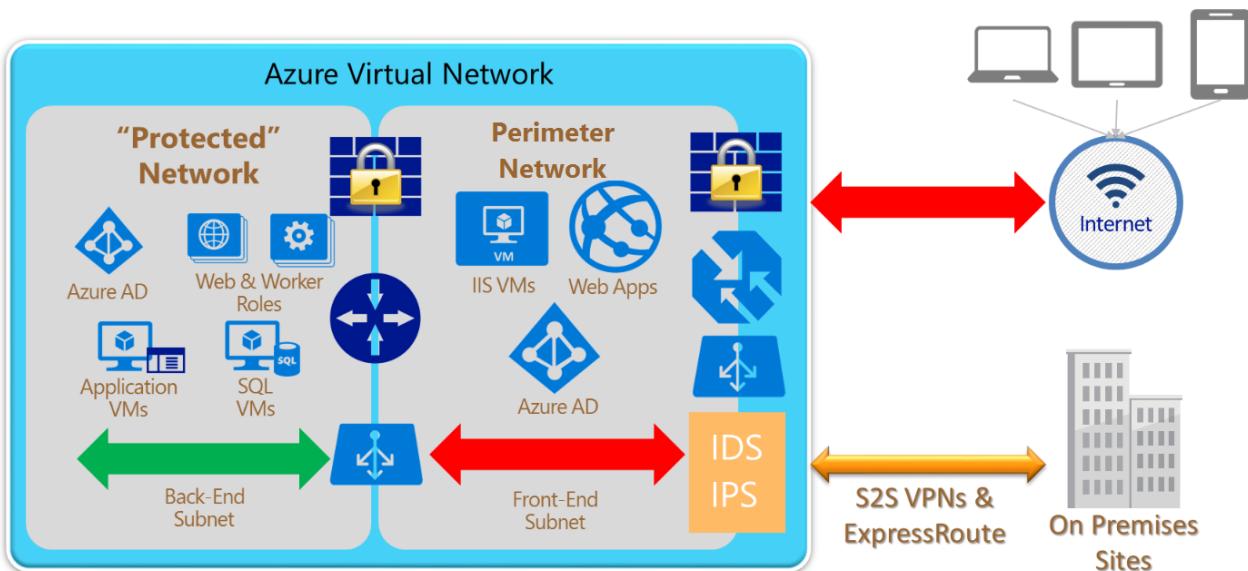
- Traffic isolation:** A virtual network is the traffic isolation boundary on the Azure platform. Virtual machines (VMs) in one virtual network cannot communicate directly to VMs in a different virtual network, even if both virtual networks are created by the same customer. Isolation is a critical property that ensures customer VMs and communication remains private within a virtual network.

NOTE

Traffic isolation refers only to traffic *inbound* to the virtual network. By default outbound traffic from the VNet to the internet is allowed, but can be prevented if desired by NSGs.

- **Multi-tier topology:** Virtual networks allow customers to define multi-tier topology by allocating subnets and designating separate address spaces for different elements or “tiers” of their workloads. These logical groupings and topologies enable customers to define different access policy based on the workload types, and also control traffic flows between the tiers.
- **Cross-premises connectivity:** Customers can establish cross-premises connectivity between a virtual network and multiple on-premises sites or other virtual networks in Azure. To construct a connection, customers can use VNet Peering, Azure VPN Gateways, third-party network virtual appliances, or ExpressRoute. Azure supports site-to-site (S2S) VPNs using standard IPsec/IKE protocols and ExpressRoute private connectivity.
- **NSG** allows customers to create rules (ACLs) at the desired level of granularity: network interfaces, individual VMs, or virtual subnets. Customers can control access by permitting or denying communication between the workloads within a virtual network, from systems on customer’s networks via cross-premises connectivity, or direct Internet communication.
- **UDR and IP Forwarding** allow customers to define the communication paths between different tiers within a virtual network. Customers can deploy a firewall, IDS/IPS, and other virtual appliances, and route network traffic through these security appliances for security boundary policy enforcement, auditing, and inspection.
- **Network virtual appliances** in the Azure Marketplace: Security appliances such as firewalls, load balancers, and IDS/IPS are available in the Azure Marketplace and the VM Image Gallery. Customers can deploy these appliances into their virtual networks, and specifically, at their security boundaries (including the perimeter network subnets) to complete a multi-tiered secure network environment.

With these features and capabilities, one example of how a perimeter network architecture could be constructed in Azure is the following diagram:



Perimeter network characteristics and requirements

The perimeter network is the front end of the network, directly interfacing communication from the Internet. The incoming packets should flow through the security appliances, such as the firewall, IDS, and IPS, before reaching the back-end servers. Internet-bound packets from the workloads can also flow through the security appliances in the perimeter network for policy enforcement, inspection, and auditing purposes, before leaving the network. Additionally, the perimeter network can host cross-premises VPN gateways between customer virtual networks and on-premises networks.

Perimeter network characteristics

Referencing the previous figure, some of the characteristics of a good perimeter network are as follows:

- Internet-facing:
 - The perimeter network subnet itself is Internet-facing, directly communicating with the Internet.

- Public IP addresses, VIPs, and/or service endpoints pass Internet traffic to the front-end network and devices.
- Inbound traffic from the Internet passes through security devices before other resources on the front-end network.
- If outbound security is enabled, traffic passes through security devices, as the final step, before passing to the Internet.
- Protected network:
 - There is no direct path from the Internet to the core infrastructure.
 - Channels to the core infrastructure must traverse through security devices such as NSGs, firewalls, or VPN devices.
 - Other devices must not bridge Internet and the core infrastructure.
 - Security devices on both the Internet-facing and the protected network facing boundaries of the perimeter network (for example, the two firewall icons shown in the previous figure) may actually be a single virtual appliance with differentiated rules or interfaces for each boundary. For example, one physical device, logically separated, handling the load for both boundaries of the perimeter network.
- Other common practices and constraints:
 - Workloads must not store business critical information.
 - Access and updates to perimeter network configurations and deployments are limited to only authorized administrators.

Perimeter network requirements

To enable these characteristics, follow these guidelines on virtual network requirements to implement a successful perimeter network:

- **Subnet architecture:** Specify the virtual network such that an entire subnet is dedicated as the perimeter network, separated from other subnets in the same virtual network. This separation ensures the traffic between the perimeter network and other internal or private subnet tiers flows through a firewall or IDS/IPS virtual appliance. User-defined routes on the boundary subnets are required to forward this traffic to the virtual appliance.
- **NSG:** The perimeter network subnet itself should be open to allow communication with the Internet, but that does not mean customers should be bypassing NSGs. Follow common security practices to minimize the network surfaces exposed to the Internet. Lock down the remote address ranges allowed to access the deployments or the specific application protocols and ports that are open. There may be circumstances, however, in which a complete lock-down is not possible. For example, if customers have an external website in Azure, the perimeter network should allow the incoming web requests from any public IP addresses, but should only open the web application ports: TCP on port 80 and/or TCP on port 443.
- **Routing table:** The perimeter network subnet itself should be able to communicate to the Internet directly, but should not allow direct communication to and from the back end or on-premises networks without going through a firewall or security appliance.
- **Security appliance configuration:** To route and inspect packets between the perimeter network and the rest of the protected networks, the security appliances such as firewall, IDS, and IPS devices may be multi-homed. They may have separate NICs for the perimeter network and the back-end subnets. The NICs in the perimeter network communicate directly to and from the Internet, with the corresponding NSGs and the perimeter network routing table. The NICs connecting to the back-end subnets have more restricted NSGs and routing tables of the corresponding back-end subnets.
- **Security appliance functionality:** The security appliances deployed in the perimeter network typically perform the following functionality:
 - Firewall: Enforcing firewall rules or access control policies for the incoming requests.
 - Threat detection and prevention: Detecting and mitigating malicious attacks from the Internet.
 - Auditing and logging: Maintaining detailed logs for auditing and analysis.

- Reverse proxy: Redirecting the incoming requests to the corresponding back-end servers. This redirection involves mapping and translating the destination addresses on the front-end devices, typically firewalls, to the back-end server addresses.
- Forward proxy: Providing NAT and performing auditing for communication initiated from within the virtual network to the Internet.
- Router: Forwarding incoming and cross-subnet traffic inside the virtual network.
- VPN device: Acting as the cross-premises VPN gateways for cross-premises VPN connectivity between customer on-premises networks and Azure virtual networks.
- VPN server: Accepting VPN clients connecting to Azure virtual networks.

TIP

Keep the following two groups separate: the individuals authorized to access the perimeter network security gear and the individuals authorized as application development, deployment, or operations administrators. Keeping these groups separate allows for a segregation of duties and prevents a single person from bypassing both applications security and network security controls.

Questions to be asked when building network boundaries

In this section, unless specifically mentioned, the term "networks" refers to private Azure virtual networks created by a subscription administrator. The term doesn't refer to the underlying physical networks within Azure.

Also, Azure virtual networks are often used to extend traditional on-premises networks. It is possible to incorporate either site-to-site or ExpressRoute hybrid networking solutions with perimeter network architectures. This hybrid link is an important consideration in building network security boundaries.

The following three questions are critical to answer when you're building a network with a perimeter network and multiple security boundaries.

1) How many boundaries are needed?

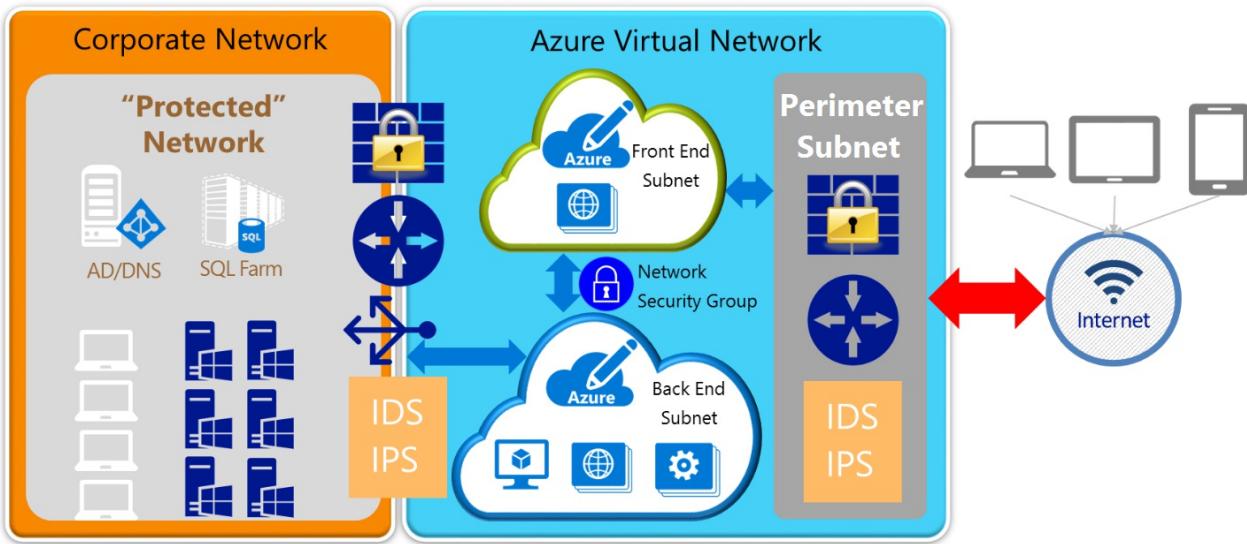
The first decision point is to decide how many security boundaries are needed in a given scenario:

- A single boundary: One on the front-end perimeter network, between the virtual network and the Internet.
- Two boundaries: One on the Internet side of the perimeter network, and another between the perimeter network subnet and the back-end subnets in the Azure virtual networks.
- Three boundaries: One on the Internet side of the perimeter network, one between the perimeter network and back-end subnets, and one between the back-end subnets and the on-premises network.
- N boundaries: A variable number. Depending on security requirements, there is no limit to the number of security boundaries that can be applied in a given network.

The number and type of boundaries needed vary based on a company's risk tolerance and the specific scenario being implemented. This decision is often made together with multiple groups within an organization, often including a risk and compliance team, a network and platform team, and an application development team. People with knowledge of security, the data involved, and the technologies being used should have a say in this decision to ensure the appropriate security stance for each implementation.

TIP

Use the smallest number of boundaries that satisfy the security requirements for a given situation. With more boundaries, operations and troubleshooting can be more difficult, as well as the management overhead involved with managing the multiple boundary policies over time. However, insufficient boundaries increase risk. Finding the balance is critical.



The preceding figure shows a high-level view of a three security boundary network. The boundaries are between the perimeter network and the Internet, the Azure front-end and back-end private subnets, and the Azure back-end subnet and the on-premises corporate network.

2) Where are the boundaries located?

Once the number of boundaries are decided, where to implement them is the next decision point. There are generally three choices:

- Using an Internet-based intermediary service (for example, a cloud-based Web application firewall, which is not discussed in this document)
- Using native features and/or network virtual appliances in Azure
- Using physical devices on the on-premises network

On purely Azure networks, the options are native Azure features (for example, Azure Load Balancers) or network virtual appliances from the rich partner ecosystem of Azure (for example, Check Point firewalls).

If a boundary is needed between Azure and an on-premises network, the security devices can reside on either side of the connection (or both sides). Thus a decision must be made on the location to place security gear.

In the previous figure, the Internet-to-perimeter network and the front-to-back-end boundaries are entirely contained within Azure, and must be either native Azure features or network virtual appliances. Security devices on the boundary between Azure (back-end subnet) and the corporate network could be either on the Azure side or the on-premises side, or even a combination of devices on both sides. There can be significant advantages and disadvantages to either option that must be seriously considered.

For example, using existing physical security gear on the on-premises network side has the advantage that no new gear is needed. It just needs reconfiguration. The disadvantage, however, is that all traffic must come back from Azure to the on-premises network to be seen by the security gear. Thus Azure-to-Azure traffic could incur significant latency, and affect application performance and user experience, if it was forced back to the on-premises network for security policy enforcement.

3) How are the boundaries implemented?

Each security boundary will likely have different capability requirements (for example, IDS and firewall rules on the Internet side of the perimeter network, but only ACLs between the perimeter network and back-end subnet).

Deciding on which device (or how many devices) to use depends on the scenario and security requirements. In the following section, examples 1, 2, and 3 discuss some options that could be used. Reviewing the Azure native network features and the devices available in Azure from the partner ecosystem shows the myriad options available to solve virtually any scenario.

Another key implementation decision point is how to connect the on-premises network with Azure. Should you use the Azure virtual gateway or a network virtual appliance? These options are discussed in greater detail in the

following section (examples 4, 5, and 6).

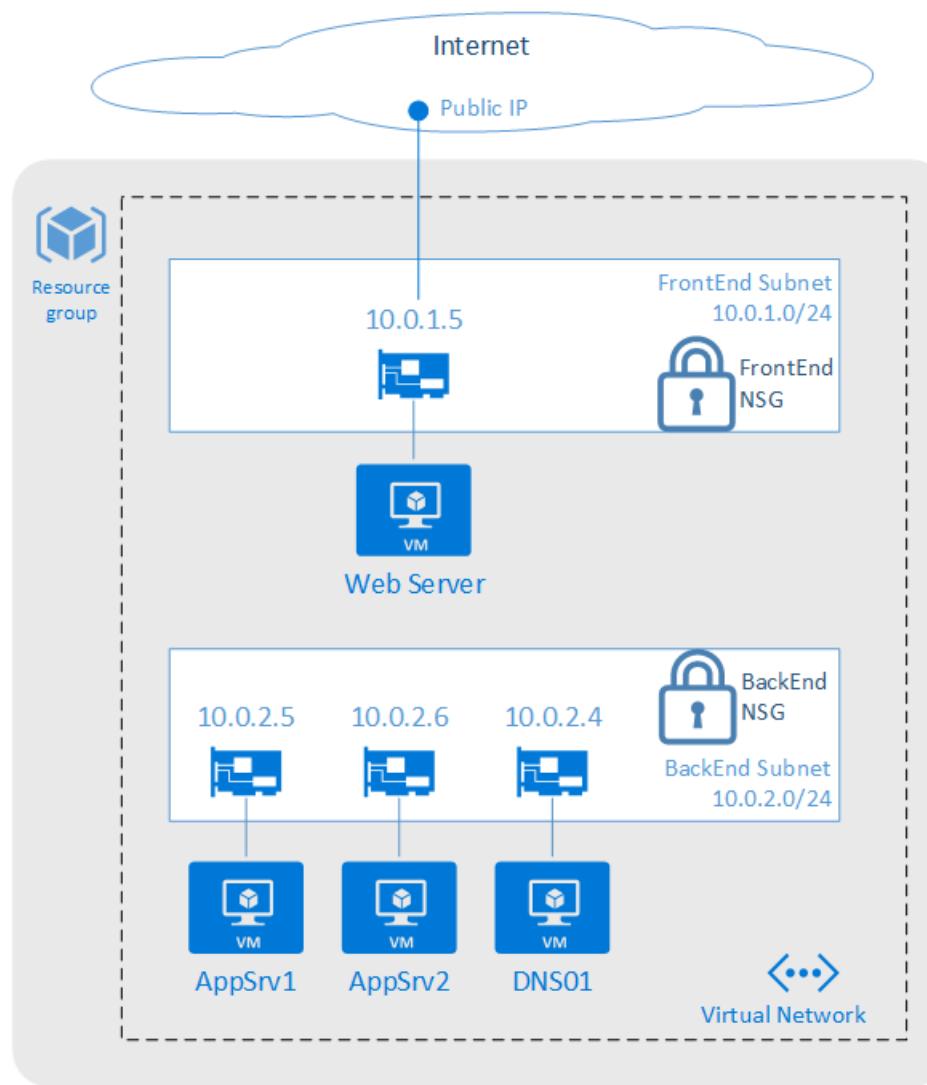
Additionally, traffic between virtual networks within Azure may be needed. These scenarios will be added in the future.

Once you know the answers to the previous questions, the [Fast Start](#) section can help identify which examples are most appropriate for a given scenario.

Examples: Building security boundaries with Azure virtual networks

Example 1 Build a perimeter network to help protect applications with NSGs

[Back to Fast start](#) | [Detailed build instructions for this example](#)



Environment description

In this example, there is a subscription that contains the following resources:

- A single resource group
- A virtual network with two subnets: "FrontEnd" and "BackEnd"
- A Network Security Group that is applied to both subnets
- A Windows server that represents an application web server ("IIS01")
- Two Windows servers that represent application back-end servers ("AppVM01", "AppVM02")
- A Windows server that represents a DNS server ("DNS01")
- A public IP associated with the application web server

For scripts and an Azure Resource Manager template, see the [detailed build instructions](#).

NSG description

In this example, an NSG group is built and then loaded with six rules.

TIP

Generally speaking, you should create your specific "Allow" rules first, followed by the more generic "Deny" rules. The given priority dictates which rules are evaluated first. Once traffic is found to apply to a specific rule, no further rules are evaluated. NSG rules can apply in either the inbound or outbound direction (from the perspective of the subnet).

Declaratively, the following rules are being built for inbound traffic:

1. Internal DNS traffic (port 53) is allowed.
2. RDP traffic (port 3389) from the Internet to any Virtual Machine is allowed.
3. HTTP traffic (port 80) from the Internet to web server (IIS01) is allowed.
4. Any traffic (all ports) from IIS01 to AppVM1 is allowed.
5. Any traffic (all ports) from the Internet to the entire virtual network (both subnets) is denied.
6. Any traffic (all ports) from the front-end subnet to the back-end subnet is denied.

With these rules bound to each subnet, if an HTTP request was inbound from the Internet to the web server, both rules 3 (allow) and 5 (deny) would apply. But because rule 3 has a higher priority, only it would apply, and rule 5 would not come into play. Thus the HTTP request would be allowed to the web server. If that same traffic was trying to reach the DNS01 server, rule 5 (deny) would be the first to apply, and the traffic would not be allowed to pass to the server. Rule 6 (deny) blocks the front-end subnet from talking to the back-end subnet (except for allowed traffic in rules 1 and 4). This rule-set protects the back-end network in case an attacker compromises the web application on the front end. The attacker would have limited access to the back-end "protected" network (only to resources exposed on the AppVM01 server).

There is a default outbound rule that allows traffic out to the Internet. For this example, we're allowing outbound traffic and not modifying any outbound rules. To lock down traffic in both directions, user-defined routing is required (see example 3).

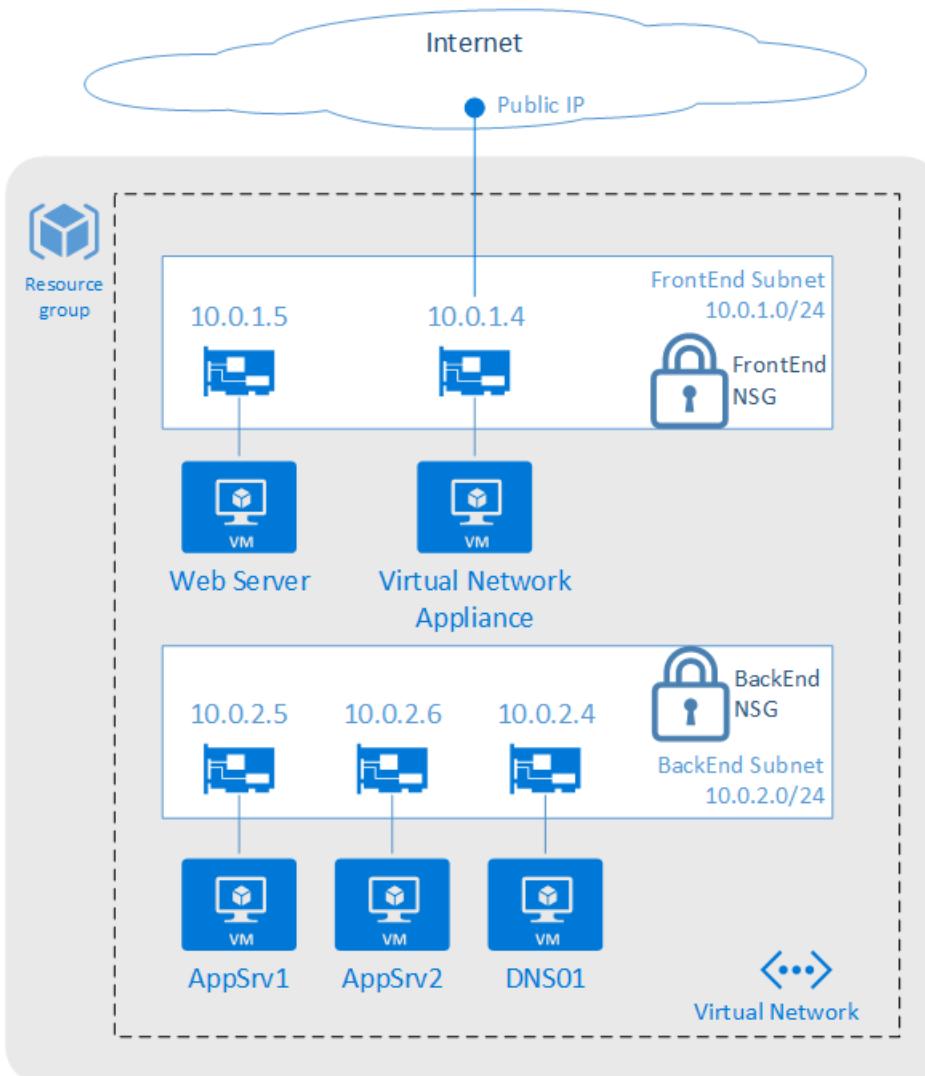
Conclusion

This example is a relatively simple and straightforward way of isolating the back-end subnet from inbound traffic. For more information, see the [detailed build instructions](#). These instructions include:

- How to build this perimeter network with classic PowerShell scripts.
- How to build this perimeter network with an Azure Resource Manager template.
- Detailed descriptions of each NSG command.
- Detailed traffic flow scenarios, showing how traffic is allowed or denied in each layer.

Example 2 Build a perimeter network to help protect applications with a firewall and NSGs

[Back to Fast start](#) | [Detailed build instructions for this example](#)



Environment description

In this example, there is a subscription that contains the following resources:

- A single resource group
- A virtual network with two subnets: "FrontEnd" and "BackEnd"
- A Network Security Group that is applied to both subnets
- A network virtual appliance, in this case a firewall, connected to the front-end subnet
- A Windows server that represents an application web server ("IIS01")
- Two Windows servers that represent application back-end servers ("AppVM01", "AppVM02")
- A Windows server that represents a DNS server ("DNS01")

For scripts and an Azure Resource Manager template, see the [detailed build instructions](#).

NSG description

In this example, an NSG group is built and then loaded with six rules.

TIP

Generally speaking, you should create your specific "Allow" rules first, followed by the more generic "Deny" rules. The given priority dictates which rules are evaluated first. Once traffic is found to apply to a specific rule, no further rules are evaluated. NSG rules can apply in either the inbound or outbound direction (from the perspective of the subnet).

Declaratively, the following rules are being built for inbound traffic:

1. Internal DNS traffic (port 53) is allowed.

2. RDP traffic (port 3389) from the Internet to any Virtual Machine is allowed.
3. Any Internet traffic (all ports) to the network virtual appliance (firewall) is allowed.
4. Any traffic (all ports) from IIS01 to AppVM1 is allowed.
5. Any traffic (all ports) from the Internet to the entire virtual network (both subnets) is denied.
6. Any traffic (all ports) from the front-end subnet to the back-end subnet is denied.

With these rules bound to each subnet, if an HTTP request was inbound from the Internet to the firewall, both rules 3 (allow) and 5 (deny) would apply. But because rule 3 has a higher priority, only it would apply, and rule 5 would not come into play. Thus the HTTP request would be allowed to the firewall. If that same traffic was trying to reach the IIS01 server, even though it's on the front-end subnet, rule 5 (deny) would apply, and the traffic would not be allowed to pass to the server. Rule 6 (deny) blocks the front-end subnet from talking to the back-end subnet (except for allowed traffic in rules 1 and 4). This rule-set protects the back-end network in case an attacker compromises the web application on the front end. The attacker would have limited access to the back-end "protected" network (only to resources exposed on the AppVM01 server).

There is a default outbound rule that allows traffic out to the Internet. For this example, we're allowing outbound traffic and not modifying any outbound rules. To lock down traffic in both directions, user-defined routing is required (see example 3).

Firewall rule description

On the firewall, forwarding rules should be created. Since this example only routes Internet traffic in-bound to the firewall and then to the web server, only one forwarding network address translation (NAT) rule is needed.

The forwarding rule accepts any inbound source address that hits the firewall trying to reach HTTP (port 80 or 443 for HTTPS). It's sent out of the firewall's local interface and redirected to the web server with the IP Address of 10.0.1.5.

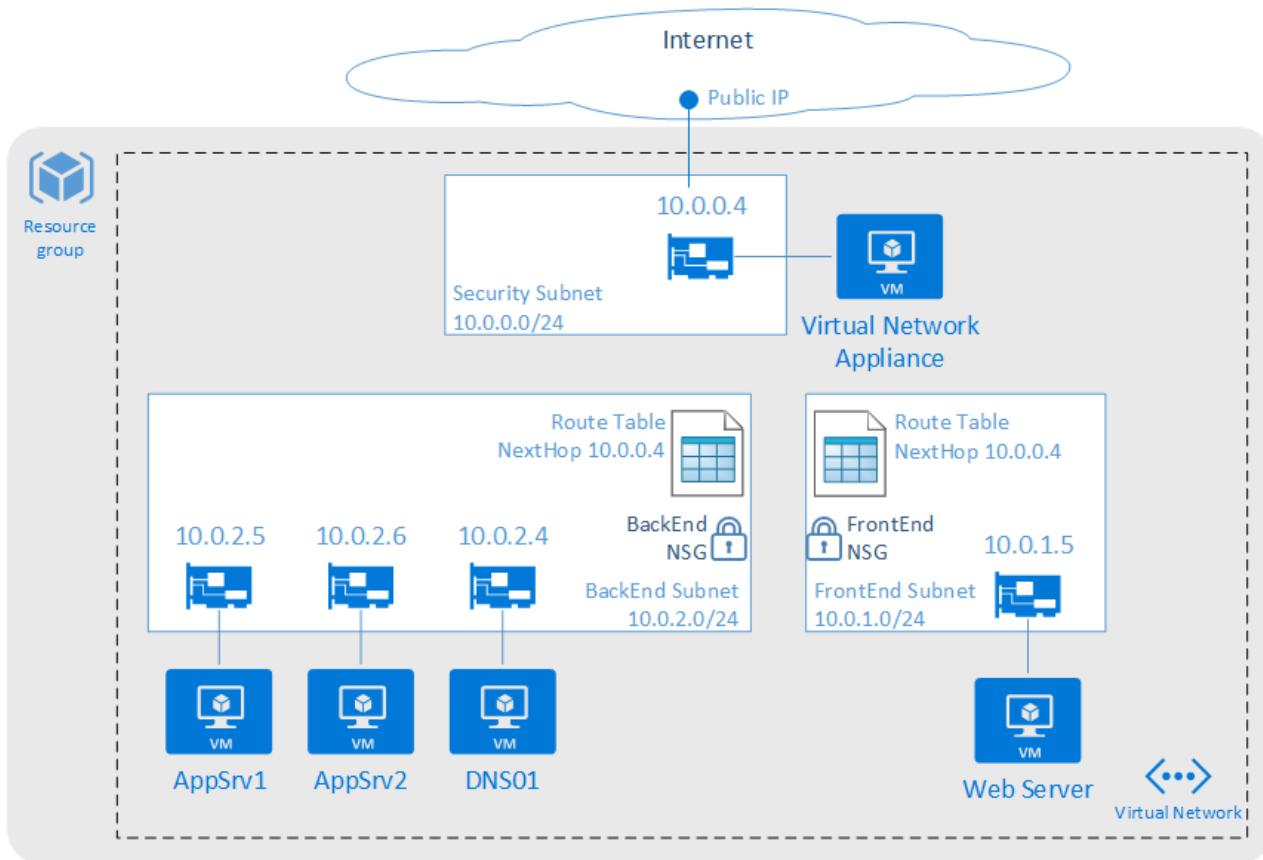
Conclusion

This example is a relatively straightforward way of protecting your application with a firewall and isolating the back-end subnet from inbound traffic. For more information, see the [detailed build instructions](#). These instructions include:

- How to build this perimeter network with classic PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed descriptions of each NSG command and firewall rule.
- Detailed traffic flow scenarios, showing how traffic is allowed or denied in each layer.

Example 3 Build a perimeter network to help protect networks with a firewall and UDR and NSG

[Back to Fast start](#) | [Detailed build instructions for this example](#)



Environment description

In this example, there is a subscription that contains the following resources:

- A single resource group
- A virtual network with three subnets: "SecNet", "FrontEnd", and "BackEnd"
- A network virtual appliance, in this case a firewall, connected to the SecNet subnet
- A Windows server that represents an application web server ("IIS01")
- Two Windows servers that represent application back-end servers ("AppVM01", "AppVM02")
- A Windows server that represents a DNS server ("DNS01")

For scripts and an Azure Resource Manager template, see the [detailed build instructions](#).

UDR description

By default, the following system routes are defined as:

Effective routes :					
Address	Prefix	Next hop type	Next hop IP address	Status	Source
{10.0.0.0/16}		VNETLocal		Active	Default
{0.0.0.0/0}		Internet		Active	Default
{10.0.0.0/8}		Null		Active	Default
{100.64.0.0/10}		Null		Active	Default
{172.16.0.0/12}		Null		Active	Default
{192.168.0.0/16}		Null		Active	Default

The VNETLocal is always one or more defined address prefixes that make up the virtual network for that specific network (that is, it changes from virtual network to virtual network, depending on how each specific virtual network is defined). The remaining system routes are static and default as indicated in the table.

In this example, two routing tables are created, one each for the front-end and back-end subnets. Each table is loaded with static routes appropriate for the given subnet. In this example, each table has three routes that direct all traffic (0.0.0.0/0) through the firewall (Next hop = Virtual Appliance IP address):

1. Local subnet traffic with no Next Hop defined to allow local subnet traffic to bypass the firewall.
2. Virtual network traffic with a Next Hop defined as firewall. This next hop overrides the default rule that allows local virtual network traffic to route directly.
3. All remaining traffic (0/0) with a Next Hop defined as the firewall.

TIP

Not having the local subnet entry in the UDR breaks local subnet communications.

- In our example, 10.0.1.0/24 pointing to VNETLocal is critical! Without it, packet leaving the Web Server (10.0.1.4) destined to another local server (for example) 10.0.1.25 will fail as they will be sent to the NVA. The NVA will send it to the subnet, and the subnet will resend it to the NVA in an infinite loop.
- The chances of a routing loop are typically higher on appliances with multiple NICs that are connected to separate subnets, which is often of traditional, on-premises appliances.

Once the routing tables are created, they must be bound to their subnets. The front-end subnet routing table, once created and bound to the subnet, would look like this output:

Effective routes :					
Address Prefix	Next hop type	Next hop IP address	Status	Source	
{10.0.1.0/24}	VNETLocal		Active		
{10.0.0.0/16}	VirtualAppliance	10.0.0.4	Active		
{0.0.0.0/0}	VirtualAppliance	10.0.0.4	Active		

NOTE

UDR can now be applied to the gateway subnet on which the ExpressRoute circuit is connected.

Examples of how to enable your perimeter network with ExpressRoute or site-to-site networking are shown in examples 3 and 4.

IP Forwarding description

IP Forwarding is a companion feature to UDR. IP Forwarding is a setting on a virtual appliance that allows it to receive traffic not specifically addressed to the appliance, and then forward that traffic to its ultimate destination.

For example, if AppVM01 makes a request to the DNS01 server, UDR would route this traffic to the firewall. With IP Forwarding enabled, the traffic for the DNS01 destination (10.0.2.4) is accepted by the appliance (10.0.0.4) and then forwarded to its ultimate destination (10.0.2.4). Without IP Forwarding enabled on the firewall, traffic would not be accepted by the appliance even though the route table has the firewall as the next hop. To use a virtual appliance, it's critical to remember to enable IP Forwarding along with UDR.

NSG description

In this example, an NSG group is built and then loaded with a single rule. This group is then bound only to the front-end and back-end subnets (not the SecNet). Declaratively the following rule is being built:

- Any traffic (all ports) from the Internet to the entire virtual network (all subnets) is denied.

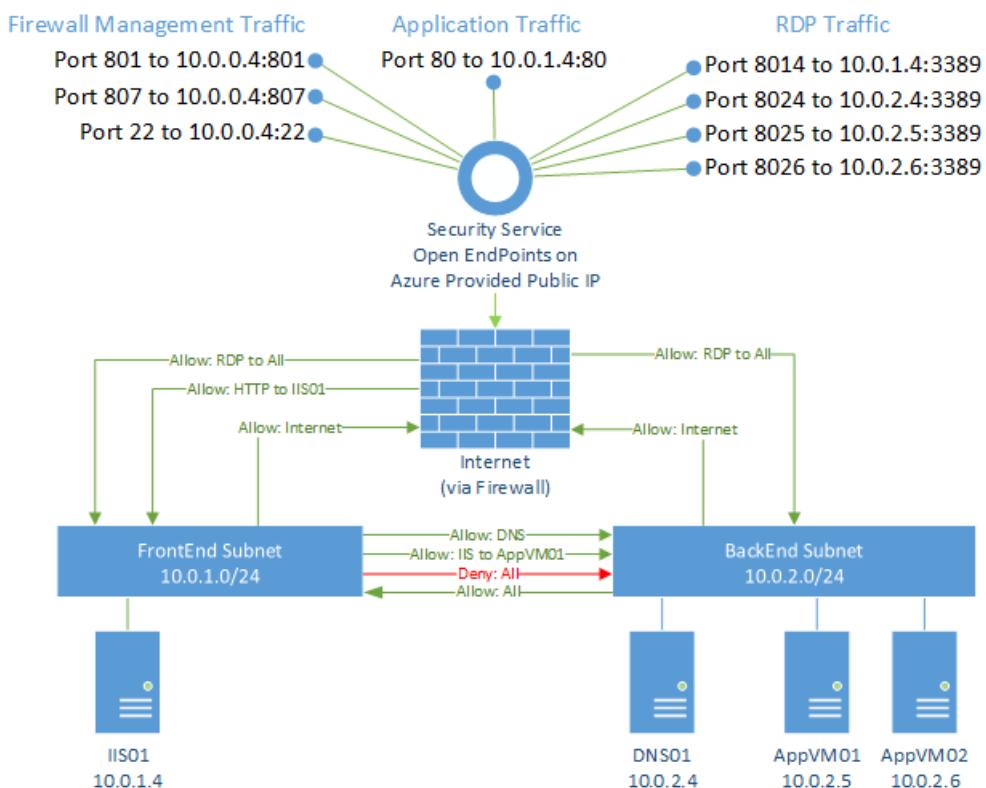
Although NSGs are used in this example, its main purpose is as a secondary layer of defense against manual misconfiguration. The goal is to block all inbound traffic from the Internet to either the front-end or back-end subnets. Traffic should only flow through the SecNet subnet to the firewall (and then, if appropriate, on to the front-end or back-end subnets). Plus, with the UDR rules in place, any traffic that did make it into the front-end or back-end subnets would be directed out to the firewall (thanks to UDR). The firewall would see this traffic as an asymmetric flow and would drop the outbound traffic. Thus there are three layers of security protecting the subnets:

- No Public IP addresses on any FrontEnd or BackEnd NICs.
- NSGs denying traffic from the Internet.
- The firewall dropping asymmetric traffic.

One interesting point regarding the NSG in this example is that it contains only one rule, which is to deny Internet traffic to the entire virtual network, including the Security subnet. However, since the NSG is only bound to the front-end and back-end subnets, the rule isn't processed on traffic inbound to the Security subnet. As a result, traffic flows to the Security subnet.

Firewall rules

On the firewall, forwarding rules should be created. Since the firewall is blocking or forwarding all inbound, outbound, and intra-virtual network traffic, many firewall rules are needed. Also, all inbound traffic hits the Security Service public IP address (on different ports), to be processed by the firewall. A best practice is to diagram the logical flows before setting up the subnets and firewall rules, to avoid rework later. The following figure is a logical view of the firewall rules for this example:



NOTE

Based on the Network Virtual Appliance used, the management ports vary. In this example, a Barracuda NextGen Firewall is referenced, which uses ports 22, 801, and 807. Consult the appliance vendor documentation to find the exact ports used for management of the device being used.

Firewall rules description

In the preceding logical diagram, the security subnet is not shown because the firewall is the only resource on that subnet. The diagram is showing the firewall rules and how they logically allow or deny traffic flows, not the actual routed path. Also, the external ports selected for the RDP traffic are higher ranged ports (8014 – 8026) and were selected to loosely align with the last two octets of the local IP address for easier readability (for example, local server address 10.0.1.4 is associated with external port 8014). Any higher non-conflicting ports, however, could be used.

For this example, we need seven types of rules:

- External rules (for inbound traffic):

1. Firewall management rule: This App Redirect rule allows traffic to pass to the management ports of the network virtual appliance.
2. RDP rules (for each Windows server): These four rules (one for each server) allow management of the individual servers via RDP. The four RDP rules could also be collapsed into one rule, depending on the capabilities of the network virtual appliance being used.
3. Application traffic rules: There are two of these rules, the first for the front-end web traffic, and the second for the back-end traffic (for example, web server to data tier). The configuration of these rules depends on the network architecture (where your servers are placed) and traffic flows (which direction the traffic flows, and which ports are used).
 - o The first rule allows the actual application traffic to reach the application server. While the other rules allow for security and management, application traffic rules are what allow external users or services to access the applications. For this example, there is a single web server on port 80. Thus a single firewall application rule redirects inbound traffic to the external IP, to the web servers internal IP address. The redirected traffic session would be translated via NAT to the internal server.
 - o The second rule is the back-end rule to allow the web server to talk to the AppVM01 server (but not AppVM02) via any port.
- Internal rules (for intra-virtual network traffic)
 1. Outbound to Internet rule: This rule allows traffic from any network to pass to the selected networks. This rule is usually a default rule already on the firewall, but in a disabled state. This rule should be enabled for this example.
 2. DNS rule: This rule allows only DNS (port 53) traffic to pass to the DNS server. For this environment, most traffic from the front end to the back end is blocked. This rule specifically allows DNS from any local subnet.
 3. Subnet to subnet rule: This rule is to allow any server on the back-end subnet to connect to any server on the front-end subnet (but not the reverse).
- Fail-safe rule (for traffic that doesn't meet any of the previous):
 1. Deny all traffic rule: This deny rule should always be the final rule (in terms of priority), and as such if a traffic flow fails to match any of the preceding rules it is dropped by this rule. This rule is a default rule and usually in-place and active. No modifications are usually needed to this rule.

TIP

On the second application traffic rule, to simplify this example, any port is allowed. In a real scenario, the most specific port and address ranges should be used to reduce the attack surface of this rule.

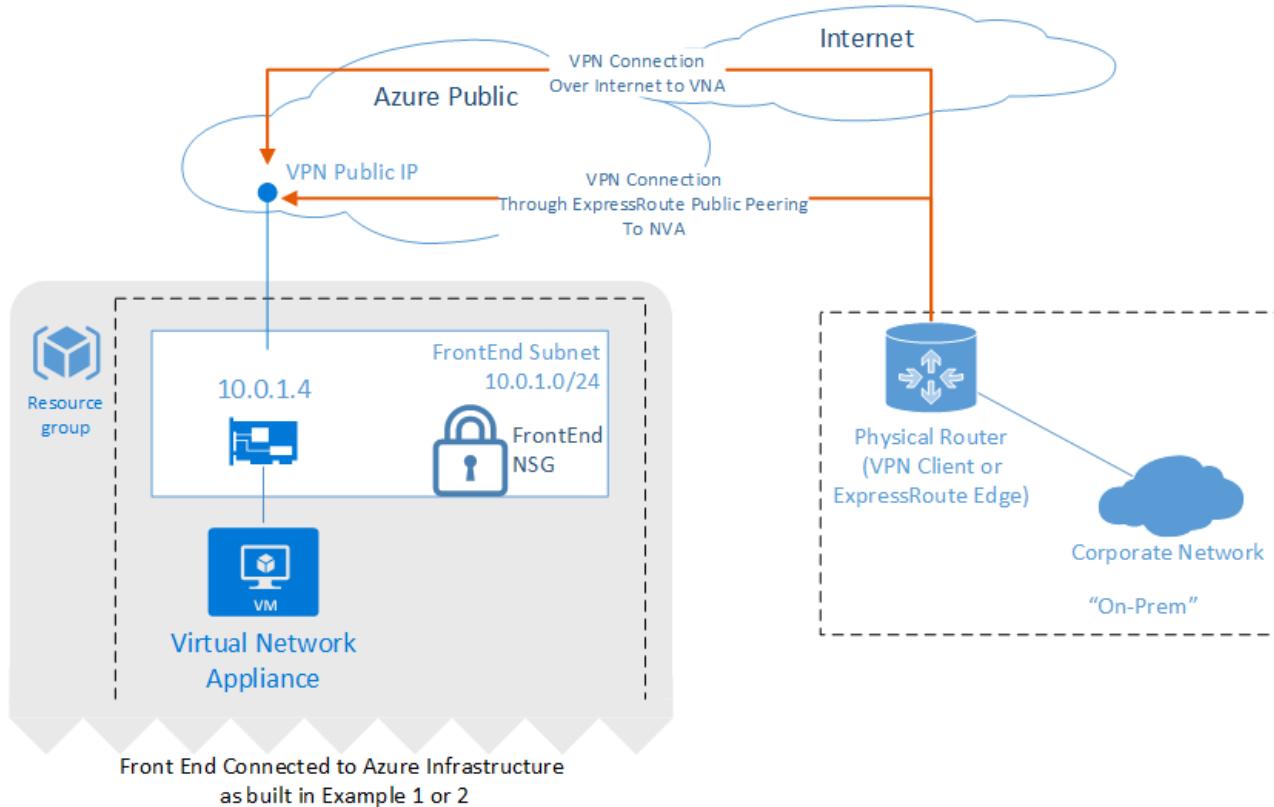
Once the previous rules are created, it's important to review the priority of each rule to ensure traffic is allowed or denied as desired. For this example, the rules are in priority order.

Conclusion

This example is a more complex but complete way of protecting and isolating the network than the previous examples. (Example 2 protects just the application, and Example 1 just isolates subnets). This design allows for monitoring traffic in both directions, and protects not just the inbound application server but enforces network security policy for all servers on this network. Also, depending on the appliance used, full traffic auditing and awareness can be achieved. For more information, see the [detailed build instructions](#). These instructions include:

- How to build this example perimeter network with classic PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed descriptions of each UDR, NSG command, and firewall rule.
- Detailed traffic flow scenarios, showing how traffic is allowed or denied in each layer.

Example 4 Add a hybrid connection with a site-to-site, virtual appliance VPN



Environment description

Hybrid networking using a network virtual appliance (NVA) can be added to any of the perimeter network types described in examples 1, 2, or 3.

As shown in the previous figure, a VPN connection over the Internet (site-to-site) is used to connect an on-premises network to an Azure virtual network via an NVA.

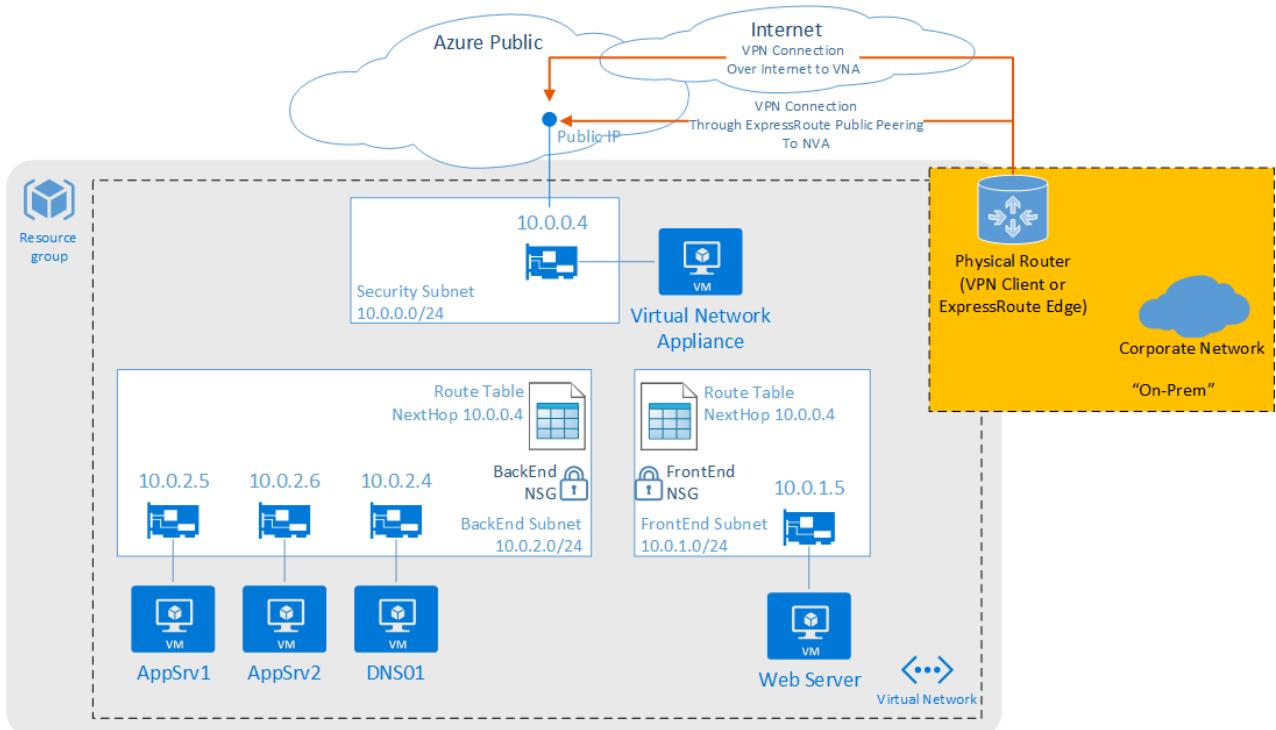
NOTE

If you use ExpressRoute with the Azure Public Peering option enabled, a static route should be created. This static route should route to the NVA VPN IP address out your corporate Internet and not via the ExpressRoute connection. The NAT required on the ExpressRoute Azure Public Peering option can break the VPN session.

Once the VPN is in place, the NVA becomes the central hub for all networks and subnets. The firewall forwarding rules determine which traffic flows are allowed, are translated via NAT, are redirected, or are dropped (even for traffic flows between the on-premises network and Azure).

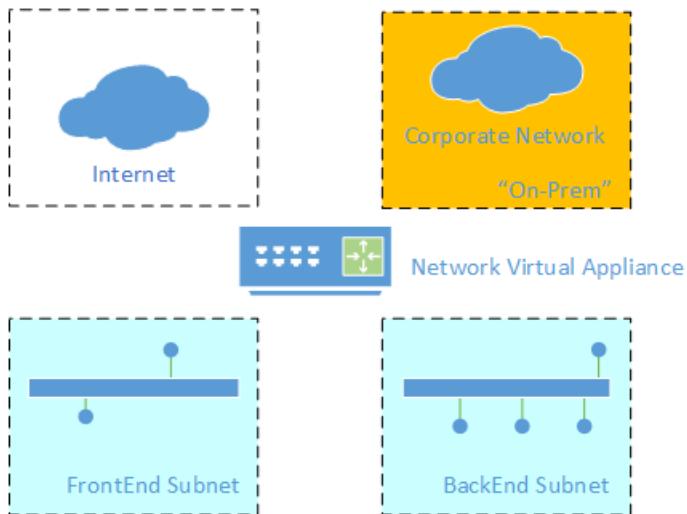
Traffic flows should be considered carefully, as they can be optimized or degraded by this design pattern, depending on the specific use case.

Using the environment built in example 3, and then adding a site-to-site VPN hybrid network connection, produces the following design:



The on-premises router, or any other network device that is compatible with your NVA for VPN, would be the VPN client. This physical device would be responsible for initiating and maintaining the VPN connection with your NVA.

Logically to the NVA, the network looks like four separate “security zones” with the rules on the NVA being the primary director of traffic between these zones:



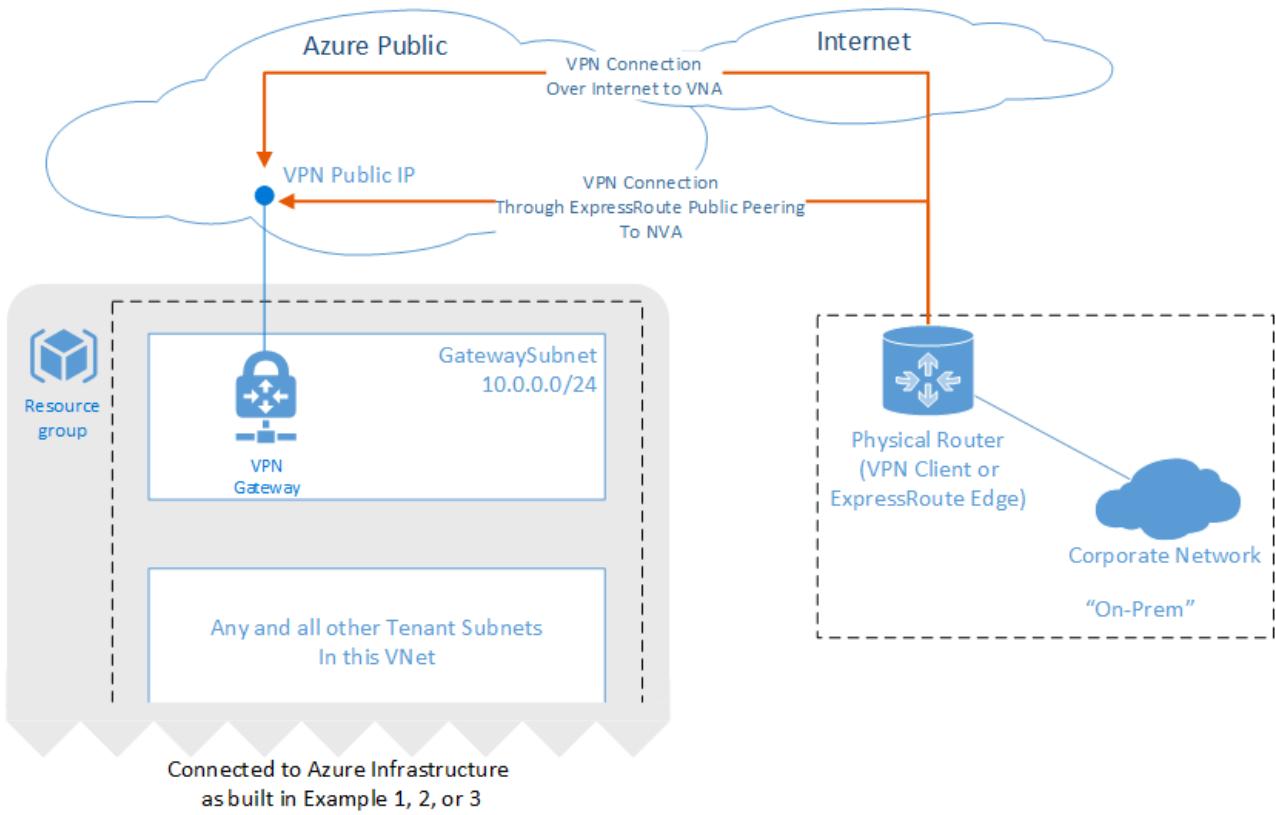
Conclusion

The addition of a site-to-site VPN hybrid network connection to an Azure virtual network can extend the on-premises network into Azure in a secure manner. In using a VPN connection, your traffic is encrypted and routes via the Internet. The NVA in this example provides a central location to enforce and manage the security policy. For more information, see the detailed build instructions (forthcoming). These instructions include:

- How to build this example perimeter network with PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed traffic flow scenarios, showing how traffic flows through this design.

Example 5 Add a hybrid connection with a site-to-site, Azure VPN gateway

[Back to Fast start](#) | Detailed build instructions available soon



Environment description

Hybrid networking using an Azure VPN gateway can be added to either perimeter network type described in examples 1 or 2.

As shown in the preceding figure, a VPN connection over the Internet (site-to-site) is used to connect an on-premises network to an Azure virtual network via an Azure VPN gateway.

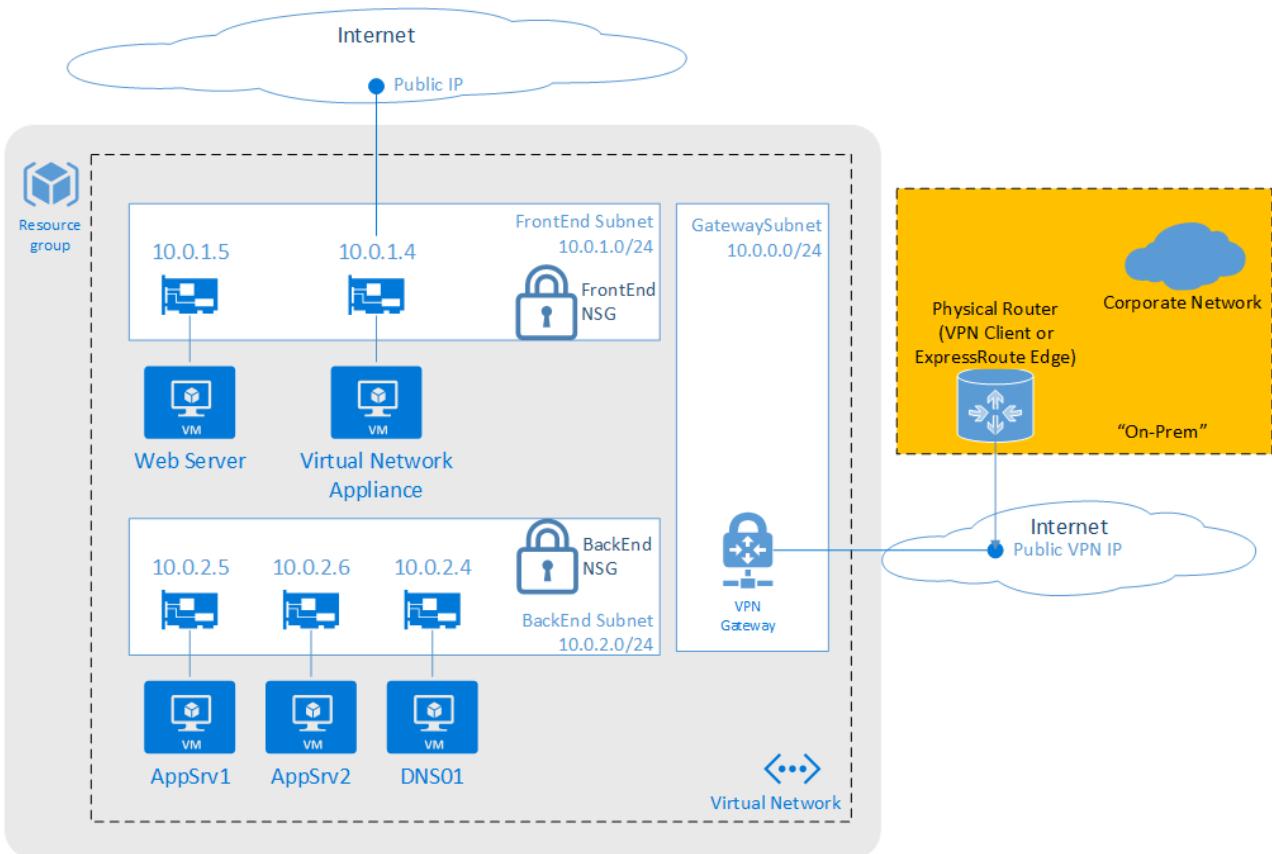
NOTE

If you use ExpressRoute with the Azure Public Peering option enabled, a static route should be created. This static route should route to the NVA VPN IP address out your corporate Internet and not via the ExpressRoute WAN. The NAT required on the ExpressRoute Azure Public Peering option can break the VPN session.

The following figure shows the two network edges in this example. On the first edge, the NVA and NSGs control traffic flows for intra-Azure networks and between Azure and the Internet. The second edge is the Azure VPN gateway, which is a separate and isolated network edge between on-premises and Azure.

Traffic flows should be considered carefully, as they can be optimized or degraded by this design pattern, depending on the specific use case.

Using the environment built in example 1, and then adding a site-to-site VPN hybrid network connection, produces the following design:



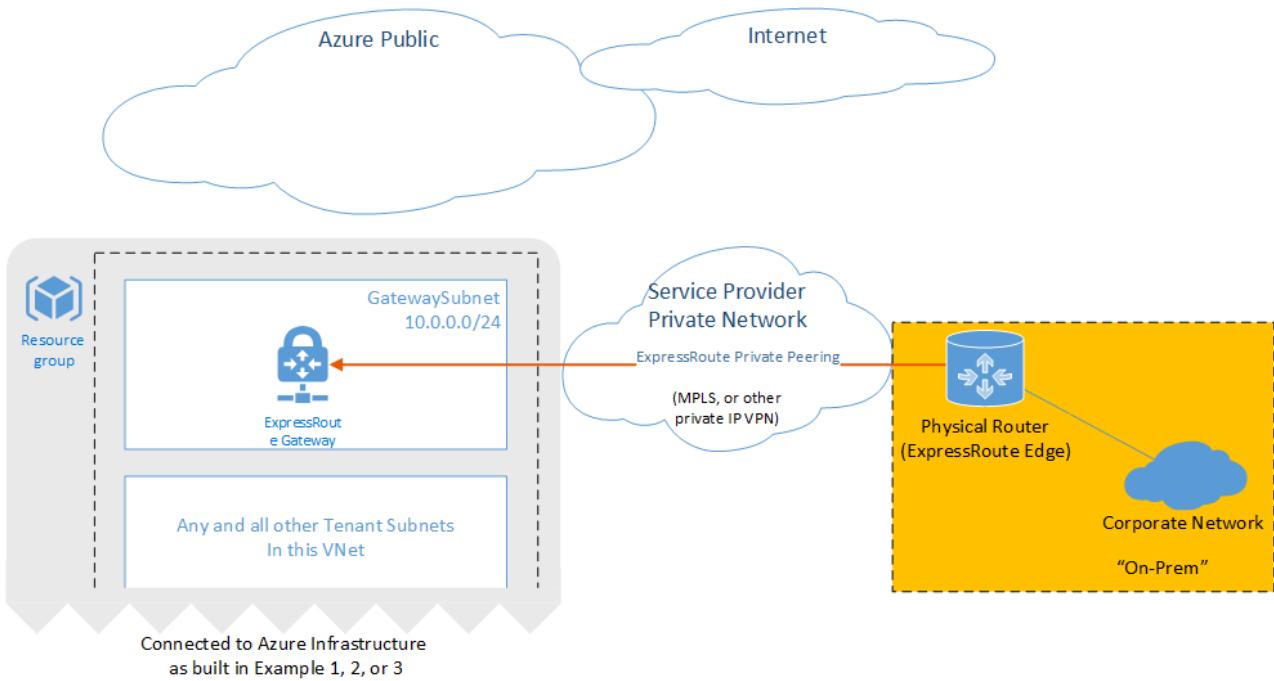
Conclusion

The addition of a site-to-site VPN hybrid network connection to an Azure virtual network can extend the on-premises network into Azure in a secure manner. Using the native Azure VPN gateway, your traffic is IPSec encrypted and routes via the Internet. Also, using the Azure VPN gateway can provide a lower-cost option (no additional licensing cost as with third-party NVAs). This option is most economical in example 1, where no NVA is used. For more information, see the detailed build instructions (forthcoming). These instructions include:

- How to build this example perimeter network with PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed traffic flow scenarios, showing how traffic flows through this design.

Example 6 Add a hybrid connection with ExpressRoute

[Back to Fast start](#) | Detailed build instructions available soon



Environment description

Hybrid networking using an ExpressRoute private peering connection can be added to either perimeter network type described in examples 1 or 2.

As shown in the preceding figure, ExpressRoute private peering provides a direct connection between your on-premises network and the Azure virtual network. Traffic transits only the service provider network and the Microsoft Azure network, never touching the Internet.

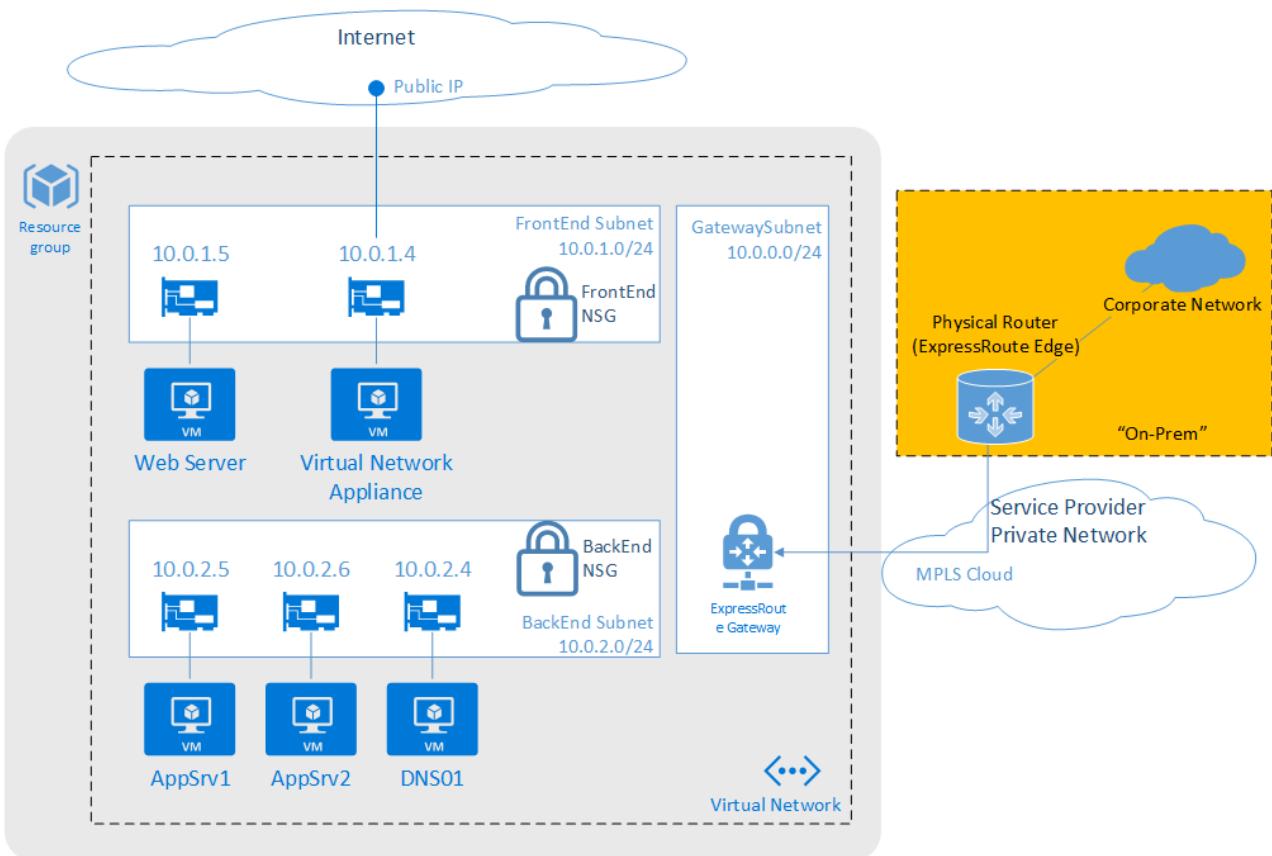
TIP

Using ExpressRoute keeps corporate network traffic off the Internet. It also allows for service level agreements from your ExpressRoute provider. The Azure Gateway can pass up to 10 Gbps with ExpressRoute, whereas with site-to-site VPNs, the Azure Gateway maximum throughput is 200 Mbps.

As seen in the following diagram, with this option the environment now has two network edges. The NVA and NSG control traffic flows for intra-Azure networks and between Azure and the Internet, while the gateway is a separate and isolated network edge between on-premises and Azure.

Traffic flows should be considered carefully, as they can be optimized or degraded by this design pattern, depending on the specific use case.

Using the environment built in example 1, and then adding an ExpressRoute hybrid network connection, produces the following design:



Conclusion

The addition of an ExpressRoute Private Peering network connection can extend the on-premises network into Azure in a secure, lower latency, higher performing manner. Also, using the native Azure Gateway, as in this example, provides a lower-cost option (no additional licensing as with third-party NVAs). For more information, see the detailed build instructions (forthcoming). These instructions include:

- How to build this example perimeter network with PowerShell scripts.
- How to build this example with an Azure Resource Manager template.
- Detailed traffic flow scenarios, showing how traffic flows through this design.

References

Helpful websites and documentation

- Access Azure with Azure Resource Manager: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/quickstart-create-a-new-resource-group>
- Accessing Azure with PowerShell: <https://docs.microsoft.com/en-us/powershell/azureps-cmdlets-docs/>
- Virtual networking documentation: <https://docs.microsoft.com/en-us/azure/virtual-network/>
- Network security group documentation: <https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-nsg>
- User-defined routing documentation: <https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview>
- Azure virtual gateways: <https://docs.microsoft.com/en-us/azure/vpn-gateway/>
- Site-to-Site VPNs: <https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-create-site-to-site-rm-powershell>
- ExpressRoute documentation (be sure to check out the "Getting Started" and "How To" sections): <https://docs.microsoft.com/en-us/azure/expressroute/>

Securing PaaS deployments

12/6/2018 • 11 minutes to read • [Edit Online](#)

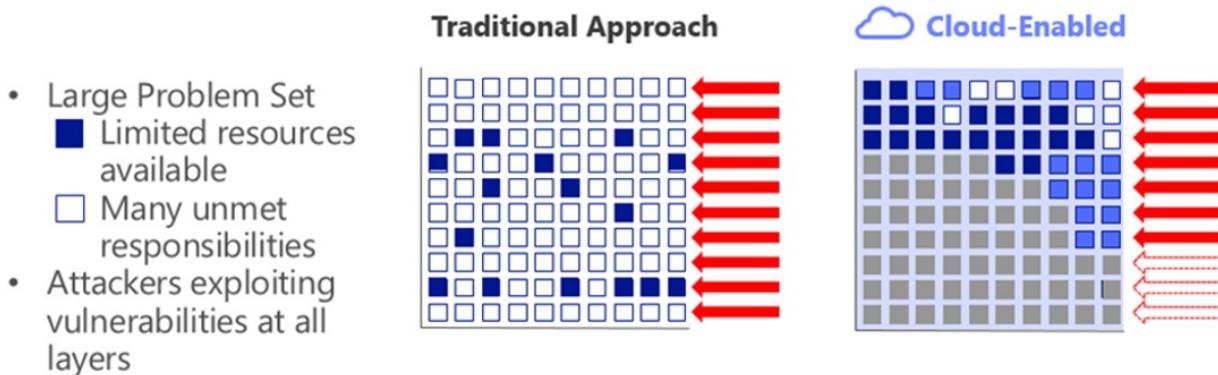
This article provides information that helps you:

- Understand the security advantages of hosting applications in the cloud
- Evaluate the security advantages of platform as a service (PaaS) versus other cloud service models
- Change your security focus from a network-centric to an identity-centric perimeter security approach
- Implement general PaaS security best practices recommendations

Cloud security advantages

There are security advantages to being in the cloud. In an on-premises environment, organizations likely have unmet responsibilities and limited resources available to invest in security, which creates an environment where attackers are able to exploit vulnerabilities at all layers.

Security Advantages of Cloud Era

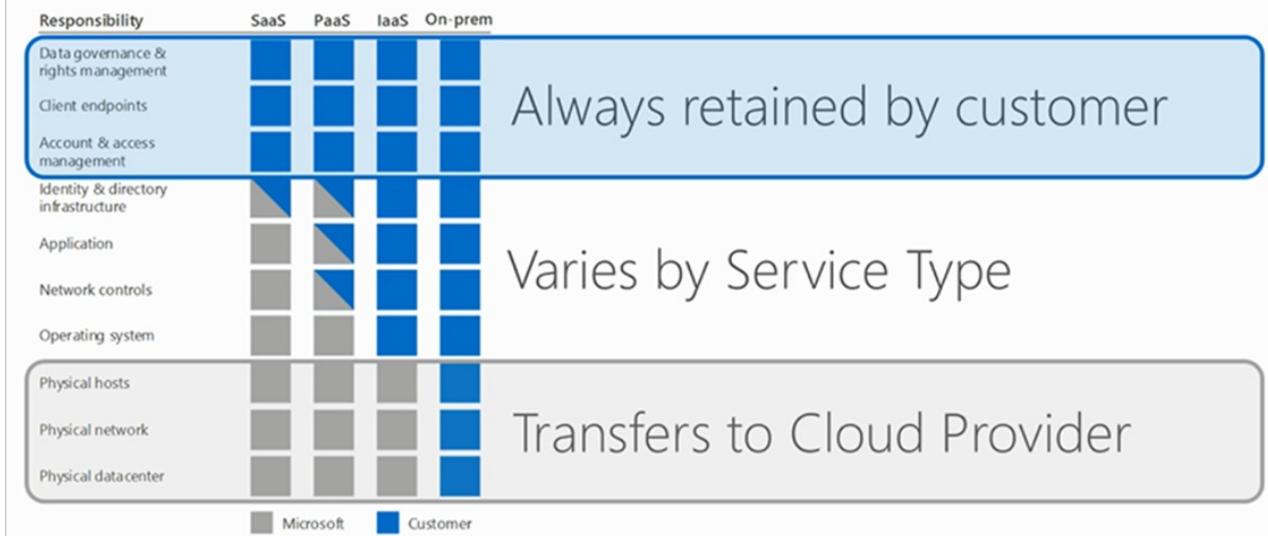


Organizations are able to improve their threat detection and response times by using a provider's cloud-based security capabilities and cloud intelligence. By shifting responsibilities to the cloud provider, organizations can get more security coverage, which enables them to reallocate security resources and budget to other business priorities.

Division of responsibility

It's important to understand the division of responsibility between you and Microsoft. On-premises, you own the whole stack but as you move to the cloud some responsibilities transfer to Microsoft. The following responsibility matrix shows the areas of the stack in a SaaS, PaaS, and IaaS deployment that you are responsible for and Microsoft is responsible for.

Responsibility Zones



For all cloud deployment types, you own your data and identities. You are responsible for protecting the security of your data and identities, on-premises resources, and the cloud components you control (which varies by service type).

Responsibilities that are always retained by you, regardless of the type of deployment, are:

- Data
- Endpoints
- Account
- Access management

Security advantages of a PaaS cloud service model

Using the same responsibility matrix, let's look at the security advantages of an Azure PaaS deployment versus on-premises.

Security advantages of PaaS

Responsibility	On-prem	PaaS		
Data governance & rights management	Customer	Customer	⚠️ Application data –	Depends on key/data management
Client endpoints	Customer	Customer	⚠️ User/endpoints –	Depends on least privilege design
Account & access management	Customer	Customer	⚠️ Admin access –	One account → access to all apps / data / infra
Identity & directory infrastructure	Customer	Customer	⚠️ Directory –	Depends on identity system / app authentication
Application	Customer	Customer	⚠️ Application code –	One exploit can lead to access of all data
Network controls	Customer	Customer	⚠️ Network configuration –	Depends on TLS usage
Operating system	Customer	Customer	⚠️ Attack Azure Infrastructure – Extremely low attack return on investment (ROI) for a single tenant <ul style="list-style-type: none">• Active security monitoring & engineering make attack very expensive• Expense limits potential attackers to small pool with larger budgets	
Physical hosts	Customer	Customer		
Physical network	Customer	Customer		
Physical datacenter	Customer	Customer		

Legend:

- Red circle: Always attractive target
- Blue circle: App design can quickly deter attacker

Starting at the bottom of the stack, the physical infrastructure, Microsoft mitigates common risks and responsibilities. Because the Microsoft cloud is continually monitored by Microsoft, it is hard to attack. It doesn't

make sense for an attacker to pursue the Microsoft cloud as a target. Unless the attacker has lots of money and resources, the attacker is likely to move on to another target.

In the middle of the stack, there is no difference between a PaaS deployment and on-premises. At the application layer and the account and access management layer, you have similar risks. In the next steps section of this article, we will guide you to best practices for eliminating or minimizing these risks.

At the top of the stack, data governance and rights management, you take on one risk that can be mitigated by key management. (Key management is covered in best practices.) While key management is an additional responsibility, you have areas in a PaaS deployment that you no longer have to manage so you can shift resources to key management.

The Azure platform also provides you strong DDoS protection by using various network-based technologies. However, all types of network-based DDoS protection methods have their limits on a per-link and per-datacenter basis. To help avoid the impact of large DDoS attacks, you can take advantage of Azure's core cloud capability of enabling you to quickly and automatically scale out to defend against DDoS attacks. We'll go into more detail on how you can do this in the recommended practices articles.

Modernizing the defender's mindset

With PaaS deployments come a shift in your overall approach to security. You shift from needing to control everything yourself to sharing responsibility with Microsoft.

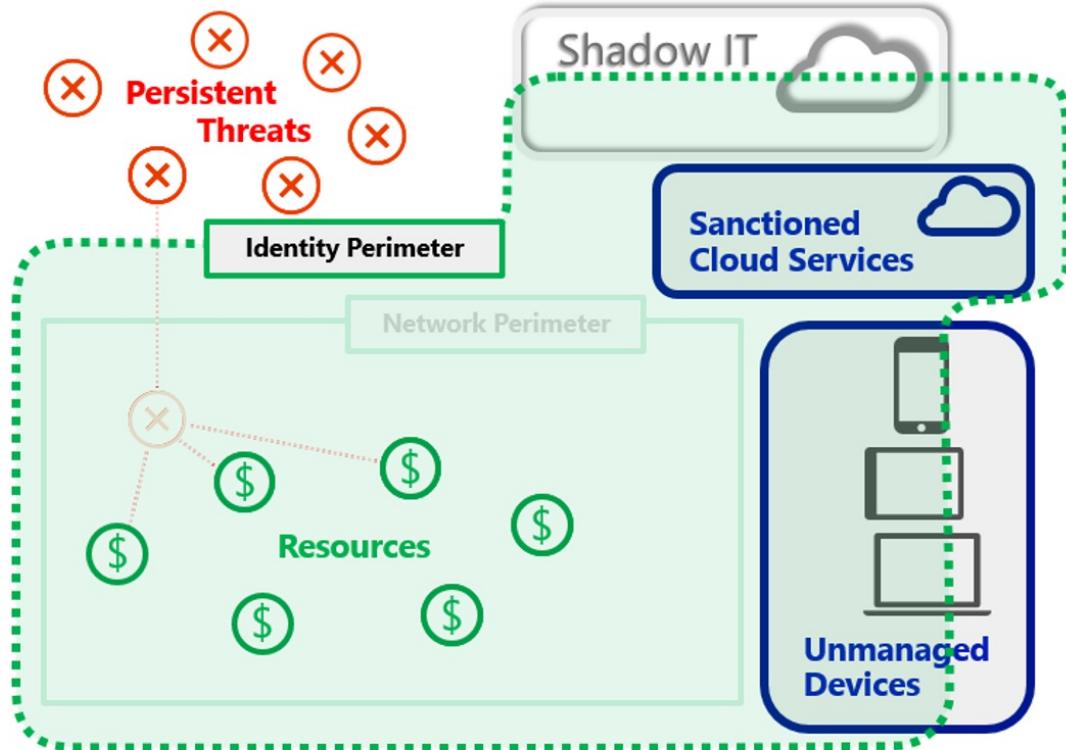
Another significant difference between PaaS and traditional on-premises deployments, is a new view of what defines the primary security perimeter. Historically, the primary on-premises security perimeter was your network and most on-premises security designs use the network as its primary security pivot. For PaaS deployments, you are better served by considering identity to be the primary security perimeter.

Adopt a policy of identity as the primary security perimeter

One of the five essential characteristics of cloud computing is broad network access, which makes network-centric thinking less relevant. The goal of much of cloud computing is to allow users to access resources regardless of location. For most users, their location is going to be somewhere on the Internet.

The following figure shows how the security perimeter has evolved from a network perimeter to an identity perimeter. Security becomes less about defending your network and more about defending your data, as well as managing the security of your apps and users. The key difference is that you want to push security closer to what's important to your company.

The Evolving Security Perimeter



Initially, Azure PaaS services (for example, web roles and Azure SQL) provided little or no traditional network perimeter defenses. It was understood that the element's purpose was to be exposed to the Internet (web role) and that authentication provides the new perimeter (for example, BLOB or Azure SQL).

Modern security practices assume that the adversary has breached the network perimeter. Therefore, modern defense practices have moved to identity. Organizations must establish an identity-based security perimeter with strong authentication and authorization hygiene (best practices).

Principles and patterns for the network perimeter have been available for decades. In contrast, the industry has relatively less experience with using identity as the primary security perimeter. With that said, we have accumulated enough experience to provide some general recommendations that are proven in the field and apply to almost all PaaS services.

The following are best practices for managing the identity perimeter.

Best practice: Secure your keys and credentials to secure your PaaS deployment.

Detail: Losing keys and credentials is a common problem. You can use a centralized solution where keys and secrets can be stored in hardware security modules. Azure provides you an HSM in the cloud with [Azure Key Vault](#).

Best practice: Don't put credentials and other secrets in source code or GitHub.

Detail: The only thing worse than losing your keys and credentials is having an unauthorized party gain access to them. Attackers can take advantage of bot technologies to find keys and secrets stored in code repositories such as GitHub. Do not put key and secrets in these public code repositories.

Best practice: Protect your VM management interfaces on hybrid PaaS and IaaS services by using a management interface that enables you to remote manage these VMs directly.

Detail: Remote management protocols such as [SSH](#), [RDP](#), and [PowerShell remoting](#) can be used. In general, we recommend that you do not enable direct remote access to VMs from the internet.

If possible, use alternate approaches like using virtual private networks in an Azure virtual network. If alternative approaches are not available, ensure that you use complex passphrases and two-factor authentication (such as [Azure Multi-Factor Authentication](#)).

Best practice: Use strong authentication and authorization platforms.

Detail: Use federated identities in Azure AD instead of custom user stores. When you use federated identities, you take advantage of a platform-based approach and you delegate the management of authorized identities to your partners. A federated identity approach is especially important when employees are terminated and that information needs to be reflected through multiple identity and authorization systems.

Use platform-supplied authentication and authorization mechanisms instead of custom code. The reason is that developing custom authentication code can be error prone. Most of your developers are not security experts and are unlikely to be aware of the subtleties and the latest developments in authentication and authorization.

Commercial code (for example, from Microsoft) is often extensively security reviewed.

Use two-factor authentication. Two-factor authentication is the current standard for authentication and authorization because it avoids the security weaknesses inherent in username and password types of authentication. Access to both the Azure management (portal/remote PowerShell) interfaces and customer-facing services should be designed and configured to use [Azure Multi-Factor Authentication](#).

Use standard authentication protocols, such as OAuth2 and Kerberos. These protocols have been extensively peer reviewed and are likely implemented as part of your platform libraries for authentication and authorization.

Use threat modeling during application design

The Microsoft [Security Development Lifecycle](#) specifies that teams should engage in a process called threat modeling during the design phase. To help facilitate this process, Microsoft has created the [SDL Threat Modeling Tool](#). Modeling the application design and enumerating [STRIDE](#) threats across all trust boundaries can catch design errors early on.

The following table lists the STRIDE threats and gives some example mitigations that use Azure features. These mitigations won't work in every situation.

THREAT	SECURITY PROPERTY	POTENTIAL AZURE PLATFORM MIGRATION
Spoofing	Authentication	Require HTTPS connections.
Tampering	Integrity	Validate SSL certificates.
Repudiation	Non-repudiation	Enable Azure monitoring and diagnostics .
Information disclosure	Confidentiality	Encrypt sensitive data at rest by using service certificates .
Denial of service	Availability	Monitor performance metrics for potential denial-of-service conditions. Implement connection filters.
Elevation of privilege	Authorization	Use Privileged Identity Management .

Develop on Azure App Service

[Azure App Service](#) is a PaaS offering that lets you create web and mobile apps for any platform or device and connect to data anywhere, in the cloud or on-premises. App Service includes the web and mobile capabilities that

were previously delivered separately as Azure Websites and Azure Mobile Services. It also includes new capabilities for automating business processes and hosting cloud APIs. As a single integrated service, App Service brings a rich set of capabilities to web, mobile, and integration scenarios.

Following are best practices for using App Service.

Best practice: [Authenticate through Azure Active Directory](#).

Detail: App Service provides an OAuth 2.0 service for your identity provider. OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, and mobile phones. Azure AD uses OAuth 2.0 to enable you to authorize access to mobile and web applications.

Best practice: Restrict access based on the need to know and least privilege security principles.

Detail: Restricting access is imperative for organizations that want to enforce security policies for data access. You can use RBAC to assign permissions to users, groups, and applications at a certain scope. To learn more about granting users access to applications, see [Get started with access management](#).

Best practice: Protect your keys.

Detail: Azure Key Vault helps safeguard cryptographic keys and secrets that cloud applications and services use. With Key Vault, you can encrypt keys and secrets (such as authentication keys, storage account keys, data encryption keys, .PFX files, and passwords) by using keys that are protected by hardware security modules (HSMs). For added assurance, you can import or generate keys in HSMs. See [Azure Key Vault](#) to learn more. You can also use Key Vault to manage your TLS certificates with auto-renewal.

Best practice: Restrict incoming source IP addresses.

Detail: [App Service Environment](#) has a virtual network integration feature that helps you restrict incoming source IP addresses through network security groups. Virtual networks enable you to place Azure resources in a non-internet, routable network that you control access to. To learn more, see [Integrate your app with an Azure virtual network](#).

Best practice: Monitor the security state of your App Service environments.

Detail: Use Azure Security Center to monitor your App Service environments. When Security Center identifies potential security vulnerabilities, it creates [recommendations](#) that guide you through the process of configuring the needed controls.

NOTE

Monitoring App Service is in preview and available only on the [Standard tier](#) of Security Center.

Install a web application firewall

Web applications are increasingly targets of malicious attacks that exploit common known vulnerabilities. Common among these exploits are SQL injection attacks, cross site scripting attacks to name a few. Preventing such attacks in application code can be challenging and may require rigorous maintenance, patching and monitoring at many layers of the application topology. A centralized web application firewall helps make security management much simpler and gives better assurance to application administrators against threats or intrusions. A WAF solution can also react to a security threat faster by patching a known vulnerability at a central location versus securing each of individual web applications. Existing application gateways can be converted to a web application firewall enabled application gateway easily.

[Web application firewall \(WAF\)](#) is a feature of Application Gateway that provides centralized protection of your web applications from common exploits and vulnerabilities. WAF is based on rules from the [Open Web Application Security Project \(OWASP\) core rule sets](#) 3.0 or 2.2.9.

Monitor the performance of your applications

Monitoring is the act of collecting and analyzing data to determine the performance, health, and availability of your application. An effective monitoring strategy helps you understand the detailed operation of the components of your application. It helps you increase your uptime by notifying you of critical issues so that you can resolve them before they become problems. It also helps you detect anomalies that might be security related.

Use [Azure Application Insights](#) to monitor availability, performance, and usage of your application, whether it's hosted in the cloud or on-premises. By using Application Insights, you can quickly identify and diagnose errors in your application without waiting for a user to report them. With the information that you collect, you can make informed choices on your application's maintenance and improvements.

Application Insights has extensive tools for interacting with the data that it collects. Application Insights stores its data in a common repository. It can take advantage of shared functionality such as alerts, dashboards, and deep analysis with the Log Analytics query language.

Next steps

In this article, we focused on security advantages of an Azure PaaS deployment and security best practices for cloud applications. Next, learn recommended practices for securing your PaaS web and mobile solutions using specific Azure services. We'll start with Azure App Service, Azure SQL Database and Azure SQL Data Warehouse, and Azure Storage. As articles on recommended practices for other Azure services become available, links will be provided in the following list:

- [Azure App Service](#)
- [Azure SQL Database and Azure SQL Data Warehouse](#)
- [Azure Storage](#)
- [Azure Cache for Redis](#)
- [Azure Service Bus](#)
- [Web Application Firewalls](#)

See [Azure security best practices and patterns](#) for more security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.

The following resources are available to provide more general information about Azure security and related Microsoft services:

- [Azure Security Team Blog](#) - for up to date information on the latest in Azure Security
- [Microsoft Security Response Center](#) - where Microsoft security vulnerabilities, including issues with Azure, can be reported or via email to secure@microsoft.com

Best practices for securing PaaS web and mobile applications using Azure App Service

9/28/2018 • 2 minutes to read • [Edit Online](#)

In this article, we discuss a collection of [Azure App Service](#) security best practices for securing your PaaS web and mobile applications. These best practices are derived from our experience with Azure and the experiences of customers like yourself.

Azure App Service is a platform-as-a-service (PaaS) offering that lets you create web and mobile apps for any platform or device and connect to data anywhere, in the cloud or on-premises. App Service includes the web and mobile capabilities that were previously delivered separately as Azure Websites and Azure Mobile Services. It also includes new capabilities for automating business processes and hosting cloud APIs. As a single integrated service, App Service brings a rich set of capabilities to web, mobile, and integration scenarios.

Authenticate through Azure Active Directory (AD)

App Service provides an OAuth 2.0 service for your identity provider. OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications, and mobile phones. Azure AD uses OAuth 2.0 to enable you to authorize access to mobile and web applications. To learn more, see [Authentication and authorization in Azure App Service](#).

Restrict access based on role

Restricting access is imperative for organizations that want to enforce security policies for data access. You can use role-based access control (RBAC) to assign permissions to users, groups, and applications at a certain scope, such as the need to know and least privilege security principles. To learn more about granting users access to applications, see [What is role-based access control](#).

Protect your keys

It doesn't matter how good your security is if you lose your subscription keys. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. With Key Vault, you can encrypt keys and secrets (such as authentication keys, storage account keys, data encryption keys, .PFX files, and passwords) by using keys that are protected by hardware security modules (HSMs). For added assurance, you can import or generate keys in HSMs. You can also use Key Vault to manage your TLS certificates with auto-renewal. See [What is Azure Key Vault](#) to learn more.

Restrict incoming source IP addresses

[App Service Environments](#) has a virtual network integration feature that helps you restrict incoming source IP addresses through network security groups (NSGs). If you are unfamiliar with Azure Virtual Networks (VNets), this is a capability that allows you to place many of your Azure resources in a non-internet, routable network that you control access to. To learn more, see [Integrate your app with an Azure Virtual Network](#).

Next steps

This article introduced you to a collection of App Service security best practices for securing your PaaS web and mobile applications. To learn more about securing your PaaS deployments, see:

- Securing PaaS deployments
- Securing PaaS databases in Azure

Best practices for securing PaaS web and mobile applications using Azure Storage

9/28/2018 • 5 minutes to read • [Edit Online](#)

In this article, we discuss a collection of Azure Storage security best practices for securing your platform-as-a-service (PaaS) web and mobile applications. These best practices are derived from our experience with Azure and the experiences of customers like yourself.

Azure makes it possible to deploy and use storage in ways not easily achievable on-premises. With Azure storage, you can reach high levels of scalability and availability with relatively little effort. Not only is Azure Storage the foundation for Windows and Linux Azure Virtual Machines, it can also support large distributed applications.

Azure Storage provides the following four services: Blob storage, Table storage, Queue storage, and File storage. To learn more, see [Introduction to Microsoft Azure Storage](#).

The [Azure Storage security guide](#) is a great source for detailed information about Azure Storage and security. This best practices article addresses at a high level some of the concepts found in the security guide and links to the security guide, as well as other sources, for more information.

This article addresses the following best practices:

- Shared access signatures (SAS)
- Role-based access control (RBAC)
- Client side encryption for high value data
- Storage Service Encryption

Use a shared access signature instead of a storage account key

Access control is critical. To help you control access to Azure Storage, Azure generates two 512-bit storage account keys (SAKs) when you create a storage account. The level of key redundancy makes it possible for you to avoid service interruptions during routine key rotation.

Storage access keys are high priority secrets and should only be accessible to those responsible for storage access control. If the wrong people get access to these keys, they will have complete control of storage and could replace, delete, or add files to storage. This includes malware and other types of content that can potentially compromise your organization or your customers.

You still need a way to provide access to objects in storage. To provide more granular access you can take advantage of shared access signature (SAS). The SAS makes it possible for you to share specific objects in storage for a pre-defined time-interval and with specific permissions. A shared access signature allows you to define:

- The interval over which the SAS is valid, including the start time and the expiry time.
- The permissions granted by the SAS. For example, a SAS on a blob might grant a user read and write permissions to that blob, but not delete permissions.
- An optional IP address or range of IP addresses from which Azure Storage accepts the SAS. For example, you might specify a range of IP addresses belonging to your organization. This provides another measure of security for your SAS.
- The protocol over which Azure Storage accepts the SAS. You can use this optional parameter to restrict access to clients using HTTPS.

SAS allows you to share content the way you want to share it without giving away your storage account keys.

Always using SAS in your application is a secure way to share your storage resources without compromising your storage account keys.

To learn more about shared access signature, see [Using shared access signatures](#).

Use role-based access control

Another way to manage access is to use [role-based access control](#) (RBAC). With RBAC, you focus on giving employees the exact permissions they need, based on the need to know and least privilege security principles. Too many permissions can expose an account to attackers. Too few permissions means that employees can't get their work done efficiently. RBAC helps address this problem by offering fine-grained access management for Azure. This is imperative for organizations that want to enforce security policies for data access.

You can use built-in RBAC roles in Azure to assign privileges to users. For example, use Storage Account Contributor for cloud operators that need to manage storage accounts and Classic Storage Account Contributor role to manage classic storage accounts. For cloud operators that need to manage VMs but not the virtual network or storage account to which they are connected, you can add them to the Virtual Machine Contributor role.

Organizations that do not enforce data access control by using capabilities such as RBAC may be giving more privileges than necessary for their users. This can lead to data compromise by allowing some users access to data they shouldn't have in the first place.

To learn more about RBAC see:

- [Manage access using RBAC and the Azure portal](#)
- [Built-in roles for Azure resources](#)
- [Azure Storage security guide](#)

Use client-side encryption for high value data

Client-side encryption enables you to programmatically encrypt data in transit before uploading to Azure Storage, and programmatically decrypt data when retrieving it. This provides encryption of data in transit but it also provides encryption of data at rest. Client-side encryption is the most secure method of encrypting your data but it does require you to make programmatic changes to your application and put key management processes in place.

Client-side encryption also enables you to have sole control over your encryption keys. You can generate and manage your own encryption keys. It uses an envelope technique where the Azure storage client library generates a content encryption key (CEK) that is then wrapped (encrypted) using the key encryption key (KEK). The KEK is identified by a key identifier and can be an asymmetric key pair or a symmetric key and can be managed locally or stored in [Azure Key Vault](#).

Client-side encryption is built into the Java and the .NET storage client libraries. See [Client-side encryption and Azure Key Vault for Microsoft Azure Storage](#) for information on encrypting data within client applications and generating and managing your own encryption keys.

Enable Storage Service Encryption for data at rest

When [Storage Service Encryption](#) for File storage is enabled, the data is encrypted automatically using AES-256 encryption. Microsoft handles all the encryption, decryption, and key management. This feature is available for LRS and GRS redundancy types.

Next steps

This article introduced you to a collection of Azure Storage security best practices for securing your PaaS web and mobile applications. To learn more about securing your PaaS deployments, see:

- Securing PaaS deployments
- Securing PaaS web and mobile applications using Azure App Services
- Securing PaaS databases in Azure

Best practices for securing PaaS databases in Azure

9/28/2018 • 4 minutes to read • [Edit Online](#)

In this article, we discuss a collection of [Azure SQL Database](#) and [SQL Data Warehouse](#) security best practices for securing your platform-as-a-service (PaaS) web and mobile applications. These best practices are derived from our experience with Azure and the experiences of customers like yourself.

Azure SQL Database and SQL Data Warehouse provide a relational database service for your internet-based applications. Let's look at services that help protect your applications and data when using Azure SQL Database and SQL Data Warehouse in a PaaS deployment:

- Azure Active Directory authentication (instead of SQL Server authentication)
- Azure SQL firewall
- Transparent Data Encryption (TDE)

Use a centralized identity repository

Azure SQL databases can be configured to use one of two types of authentication:

- **SQL authentication** uses a username and password. When you created the logical server for your database, you specified a "server admin" login with a username and password. Using these credentials, you can authenticate to any database on that server as the database owner.
- **Azure Active Directory authentication** uses identities managed by Azure Active Directory and is supported for managed and integrated domains. To use Azure Active Directory Authentication, you must create another server admin called the "Azure AD admin," which is allowed to administer Azure AD users and groups. This admin can also perform all operations that a regular server admin can.

[Azure Active Directory authentication](#) is a mechanism of connecting to Azure SQL Database and SQL Data Warehouse by using identities in Azure Active Directory (AD). Azure AD provides an alternative to SQL Server authentication so you can stop the proliferation of user identities across database servers. Azure AD authentication enables you to centrally manage the identities of database users and other Microsoft services in one central location. Central ID management provides a single place to manage database users and simplifies permission management.

Benefits of using Azure AD instead of SQL authentication

- Allows password rotation in a single place.
- Manages database permissions using external Azure AD groups.
- Eliminates storing passwords by enabling integrated Windows authentication and other forms of authentication supported by Azure AD.
- Uses contained database users to authenticate identities at the database level.
- Supports token-based authentication for applications connecting to SQL Database.
- Supports domain federation with Active Directory Federation Services (ADFS) or native user/password authentication for a local Azure AD without domain synchronization.
- Supports connections from SQL Server Management Studio that use Active Directory Universal Authentication, which includes [Multi-Factor Authentication \(MFA\)](#). MFA includes strong authentication with a range of easy verification options — phone call, text message, smart cards with pin, or mobile app notification. For more information, see [Universal Authentication with SQL Database and SQL Data Warehouse](#).

To learn more about Azure AD authentication, see:

- [Use Azure Active Directory Authentication for authentication with SQL Database, Managed Instance, or SQL Data Warehouse](#)
- [Authentication to Azure SQL Data Warehouse](#)
- [Token-based authentication support for Azure SQL DB using Azure AD authentication](#)

NOTE

To ensure that Azure Active Directory is a good fit for your environment, see [Azure AD features and limitations](#).

Restrict access based on IP address

You can create firewall rules that specify ranges of acceptable IP addresses. These rules can be targeted at both the server and database levels. We recommend using database-level firewall rules whenever possible to enhance security and to make your database more portable. Server-level firewall rules are best used for administrators and when you have many databases that have the same access requirements but you don't want to spend time configuring each database individually.

SQL Database default source IP address restrictions allow access from any Azure address, including other subscriptions and tenants. You can restrict this to only allow your IP addresses to access the instance. Even with your SQL firewall and IP address restrictions, strong authentication is still needed. See the recommendations made earlier in this article.

To learn more about Azure SQL Firewall and IP restrictions, see:

- [Azure SQL Database and SQL Data Warehouse access control](#)
- [Azure SQL Database and SQL Data Warehouse firewall rules](#)

Encrypt data at rest

[Transparent Data Encryption \(TDE\)](#) is enabled by default. TDE transparently encrypts SQL Server, Azure SQL Database, and Azure SQL Data Warehouse data and log files. TDE protects against a compromise of direct access to the files or their backup. This enables you to encrypt data at rest without changing existing applications. TDE should always stay enabled; however, this will not stop an attacker using the normal access path. TDE provides the ability to comply with many laws, regulations, and guidelines established in various industries.

Azure SQL manages key related issues for TDE. As with TDE, on-premises special care must be taken to ensure recoverability and when moving databases. In more sophisticated scenarios, the keys can be explicitly managed in Azure Key Vault through extensible key management. See [Enable TDE on SQL Server Using EKM](#). This also allows for Bring Your Own Key (BYOK) through Azure Key Vaults BYOK capability.

Azure SQL provides encryption for columns through [Always Encrypted](#). This allows only authorized applications access to sensitive columns. Using this kind of encryption limits SQL queries for encrypted columns to equality-based values.

Application level encryption should also be used for selective data. Data sovereignty concerns can sometimes be mitigated by encrypting data with a key that is kept in the correct country. This prevents even accidental data transfer from causing an issue since it is impossible to decrypt the data without the key, assuming a strong algorithm is used (such as AES 256).

You can use additional precautions to help secure the database, such as designing a secure system, encrypting confidential assets, and building a firewall around the database servers.

Next steps

This article introduced you to a collection of SQL Database and SQL Data Warehouse security best practices for

securing your PaaS web and mobile applications. To learn more about securing your PaaS deployments, see:

- [Securing PaaS deployments](#)
- [Securing PaaS web and mobile applications using Azure App Services](#)

Internet of Things security overview

6/18/2018 • 2 minutes to read • [Edit Online](#)

Azure internet of things (IoT) services offer a broad range of capabilities. These enterprise grade services enable you to:

- Collect data from devices
- Analyze data streams in-motion
- Store and query large data sets
- Visualize both real-time and historical data
- Integrate with back-office systems

To deliver these capabilities, Azure IoT solution accelerators package together multiple Azure services with custom extensions as preconfigured solutions. These preconfigured solutions are base implementations of common IoT solution patterns that help to reduce the time you take to deliver your IoT solutions. Using the IoT software development kits, you can customize and extend these solutions to meet your own requirements. You can also use these solutions as examples or templates when you are developing new IoT solutions.

The Azure IoT solution accelerators are powerful solutions for your IoT needs. However, it's of upmost importance that your IoT solutions are designed with security in mind from the start. Because of the sheer number of IoT devices, any security incident can quickly become a widespread event with significant consequences.

To help you understand how to secure your IoT solutions, we have the following information.

Security architecture

When designing a system, it is important to understand the potential threats to that system, and add appropriate defenses accordingly, as the system is designed and architected. It is important to design the product from the start with security in mind because understanding how an attacker might be able to compromise a system helps make sure appropriate mitigations are in place from the beginning.

You can learn about IoT security architecture by reading [Internet of Things Security Architecture](#).

This article discusses the following topics:

- [Security Starts with a Threat Model](#)
- [Security in IoT](#)
- [Threat Modeling the Azure IoT Reference Architecture](#)

Security from the ground up

The IoT poses unique security, privacy, and compliance challenges to businesses worldwide. Unlike traditional cyber technology where these issues revolve around software and how it is implemented, IoT concerns what happens when the cyber and the physical worlds converge. Protecting IoT solutions requires ensuring secure provisioning of devices, secure connectivity between these devices and the cloud, and secure data protection in the cloud during processing and storage. Working against such functionality, however, are resource-constrained devices, geographic distribution of deployments, and many devices within a solution.

You can learn how to handle security in these areas by reading [Internet of Things security from the ground up](#).

The article discusses the following topics:

- [Secure infrastructure from the ground up](#)
- [Microsoft Azure – secure IoT infrastructure for your business](#)

Best Practices

Securing an IoT infrastructure requires a rigorous security-in-depth strategy. From securing data in the cloud, protecting data integrity while in transit over the public internet, to securely provisioning devices, each layer builds greater security assurance in the overall infrastructure.

You can learn about Internet of Things security best practices by reading [Internet of Things security best practices](#).

The article discusses the following topics:

- [IoT hardware manufacturer/integrator](#)
- [IoT solution developer](#)
- [IoT solution deployer](#)
- [IoT solution operator](#)

2 minutes to read

2 minutes to read

2 minutes to read

Azure Service Fabric security overview

7/3/2018 • 9 minutes to read • [Edit Online](#)

[Azure Service Fabric](#) is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices. Service Fabric addresses the challenges of developing and managing cloud applications. Developers and administrators can avoid complex infrastructure problems and focus on implementing mission-critical, demanding workloads that are scalable and reliable.

This article is an overview of security considerations for a Service Fabric deployment.

Secure your cluster

Azure Service Fabric orchestrates services across a cluster of machines. Clusters must be secured to prevent unauthorized users from connecting to them, especially when they're running production workloads. Although it's possible to create an unsecured cluster, this might allow anonymous users to connect to the cluster (if it exposes management endpoints to the public internet).

For clusters that are running either standalone or on Azure, two scenarios to consider are node-to-node security and client-to-node security. You can use various technologies to implement those scenarios.

Node-to-node security

Node-to-node security applies to communication between the VMs or machines in a cluster. With node-to-node security, only computers that are authorized to join the cluster can participate in hosting applications and services in the cluster.

Clusters that are running on Azure or standalone clusters that are running on Windows can use either [certificate security](#) or [Windows security](#) for Windows Server machines.

Node-to-node certificate security

Service Fabric uses X.509 server certificates that you specify when you create a cluster. For a quick overview of what these certificates are and how you can acquire or create them, see [Working with certificates](#).

You configure certificate security when you create the cluster through the Azure portal, Azure Resource Manager templates, or a standalone JSON template. You can specify a primary certificate and an optional secondary certificate that's used for certificate rollovers. The primary and secondary certificates that you specify should be different from the admin client and read-only client certificates that you specify for [client-to-node security](#).

Client-to-node security

You configure client-to-node security by using client identities. To establish trust between a client and a cluster, you must configure the cluster to know which client identities it can trust.

Service Fabric supports two access control types for clients that are connected to a Service Fabric cluster:

- **Administrator:** Full access to management capabilities, including read/write capabilities.
- **User:** Only read access to management capabilities (for example, query capabilities), and the ability to resolve applications and services.

By using access control, cluster administrators can limit access to certain types of cluster operations. This makes the cluster more secure.

Client-to-node certificate security

You configure client-to-node certificate security when you create a cluster through the Azure portal, Resource Manager templates, or a standalone JSON template. You need to specify an admin client certificate and/or a user

client certificate. Make sure that these certificates are different from the primary and secondary certificates that you specify for node-to-node security.

Clients that connect to the cluster by using the admin certificate have full access to management capabilities. Clients that connect to the cluster by using the read-only user client certificate have only read access to management capabilities. In other words, these certificates are used for role-based access control (RBAC).

To learn how to configure certificate security in a cluster, see [Set up a cluster by using an Azure Resource Manager template](#).

Client-to-node Azure Active Directory security

Clusters that are running on Azure can also secure access to the management endpoints by using Azure Active Directory (Azure AD). For information about how to create the necessary Azure Active Directory artifacts, how to populate them during cluster creation, and how to connect to those clusters, see [Set up a cluster by using an Azure Resource Manager template](#).

Azure AD enables organizations (known as tenants) to manage user access to applications. There are applications with a web-based sign-in UI, and applications with a native client experience.

A Service Fabric cluster offers several entry points to its management functionality, including the web-based Service Fabric Explorer and Visual Studio. As a result, you create two Azure AD applications to control access to the cluster: one web application and one native application.

For Azure clusters, we recommend that you use Azure AD security to authenticate clients and certificates for node-to-node security.

For standalone Windows Server clusters with Windows Server 2012 R2 and Active Directory, we recommend that you use Windows security with group Managed Service Accounts (gMSAs). Otherwise, use Windows security with Windows accounts.

Understand monitoring and diagnostics in Service Fabric

[Monitoring and diagnostics](#) are critical to developing, testing, and deploying applications and services in any environment. Service Fabric solutions work best when you implement monitoring and diagnostics to ensure that applications and services work as expected in a local development environment or in production.

From a security perspective, the main goals of monitoring and diagnostics are:

- Detect and diagnose hardware and infrastructure problems that might be caused by a security event.
- Detect software and app issues that might be an indicator of compromise (IoC).
- Understand resource consumption to help prevent inadvertent denial of service.

The workflow of monitoring and diagnostics consists of three steps:

1. **Event generation:** Event generation includes events (logs, traces, custom events) at both the infrastructure (cluster) level and the application/service level. Read more about [infrastructure-level events](#) and [application-level events](#) to understand what's provided and how to add further instrumentation.
2. **Event aggregation:** Generated events need to be collected and aggregated before they can be displayed. We typically recommend using [Azure Diagnostics](#) (similar to agent-based log collection) or [EventFlow](#) (in-process log collection).
3. **Analysis:** Events need to be visualized and accessible in some format, to allow for analysis and display. There are several platforms for the analysis and visualization of monitoring and diagnostics data. We recommend [Azure Log Analytics](#) and [Azure Application Insights](#) because they integrate well with Service Fabric.

You can also use [Azure Monitor](#) to monitor many of the Azure resources on which a Service Fabric cluster is built.

A watchdog is a separate service that can watch health and load across services, and report health for anything in the health model hierarchy. Using a watchdog can help prevent errors that would not be detected based on the view of a single service.

Watchdogs are also a good place to host code that performs remedial actions without user interaction. An example is cleaning up log files in storage at certain time intervals. You can find a sample watchdog service implementation at [Azure Service Fabric watchdog sample](#).

Secure communication by using certificates

Certificates help you secure the communication between the various nodes of your standalone Windows cluster. By using X.509 certificates, you can also authenticate clients that are connecting to this cluster. This ensures that only authorized users can access the cluster. We recommend that you enable a certificate on the cluster when you create it.

X.509 digital certificates are commonly used to authenticate clients and servers. They're also used to encrypt and digitally sign messages.

The following table lists the certificates that you need on your cluster setup:

CERTIFICATE INFORMATION SETTING	DESCRIPTION
ClusterCertificate	This certificate is required to secure the communication between the nodes on a cluster. You can use two cluster certificates: a primary certificate, and a secondary for upgrade.
ServerCertificate	This certificate is presented to the client when it tries to connect to this cluster. You can use two server certificates: a primary certificate, and a secondary for upgrade.
ClientCertificateThumbprints	This is a set of certificates to install on the authenticated clients.
ClientCertificateCommonNames	This is the common name of the first client certificate for CertificateCommonName. CertificateIssuerThumbprint is the thumbprint for the issuer of this certificate.
ReverseProxyCertificate	This is an optional certificate that you can specify to secure your reverse proxy .

For more information about securing certificates, see [Secure a standalone cluster on Windows by using X.509 certificates](#).

Understand role-based access control

You specify the administrator and user client roles at the time of cluster creation by providing separate identities (including certificates) for each. For more information about the default access control settings and how to change the default settings, see [Role-based access control for Service Fabric clients](#).

Secure standalone clusters by using Windows security

To prevent unauthorized access to a Service Fabric cluster, you must secure the cluster. Security is especially important when the cluster runs production workloads. You configure node-to-node and client-to-node security by using Windows security in the ClusterConfig.JSON file.

When Service Fabric needs to run under a gMSA, you configure node-to-node security by setting

[ClusterIdentity](#). To build trust relationships between nodes, you must make them aware of each other.

If you want to use a machine group within an Active Directory domain, you configure node-to-node security by setting ClusterIdentity. For more information, see [Create a machine group in Active Directory](#).

You configure client-to-node security by using ClientIdentities. You must configure the cluster to recognize which client identities it can trust. You can establish trust in two ways:

- Specify the domain group users that can connect.
- Specify the domain node users that can connect.

Configure application security in Service Fabric

Manage secrets in Service Fabric applications

Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text.

You can use [Azure Key Vault](#) to manage keys and secrets. However, the use of secrets in an application doesn't rely on a specific cloud platform. You can deploy applications to a cluster that's hosted anywhere. There are four main steps in this flow:

1. Get a data encipherment certificate.
2. Install the certificate on your cluster.
3. Encrypt secret values when deploying an application with the certificate and inject them into a service's Settings.xml configuration file.
4. Read encrypted values out of Settings.xml by decrypting them with the same encipherment certificate.

For more information, see [Manage secrets in Service Fabric applications](#).

Configure security policies for an application

By using Azure Service Fabric security, you can help secure applications that are running in the cluster under different user accounts. Service Fabric security also helps secure the resources that applications use at the time of deployment under the user accounts--for example, files, directories, and certificates. This makes running applications, even in a shared hosted environment, more secure.

Tasks for configuring security policies include:

- Configuring the policy for a service setup entry point
- Starting PowerShell commands from a setup entry point
- Using console redirection for local debugging
- Configuring a policy for service code packages
- Assigning a security access policy for HTTP and HTTPS endpoints

Secure communication for services

Security is one of the most important aspects of communication. The Reliable Services application framework provides a few prebuilt communication stacks and tools that you can use to improve security. For more information, see [Secure service remoting communications for a service](#).

Next steps

- For conceptual information about cluster security, see [Create a Service Fabric cluster by using Azure Resource Manager](#) and [Create a Service Fabric cluster by using the Azure portal](#).
- To learn more about cluster security in Service Fabric, see [Service Fabric cluster security scenarios](#).

Azure Service Fabric security best practices

5/7/2018 • 10 minutes to read • [Edit Online](#)

Deploying an application on Azure is fast, easy, and cost-effective. Before you deploy your cloud application into production, review our list of essential and recommended best practices for implementing secure clusters in your application.

Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices. Service Fabric also addresses the significant challenges in developing and managing cloud applications. Developers and administrators can avoid complex infrastructure problems and focus on implementing mission-critical, demanding workloads that are scalable, reliable, and manageable.

For each best practice, we explain:

- What the best practice is.
- Why you should implement the best practice.
- What might happen if you don't implement the best practice.
- How you can learn to implement the best practice.

We recommend the following Azure Service Fabric security best practices:

- Use Azure Resource Manager templates and the Service Fabric PowerShell module to create secure clusters.
- Use X.509 certificates.
- Configure security policies.
- Implement the Reliable Actors security configuration.
- Configure SSL for Azure Service Fabric.
- Use network isolation and security with Azure Service Fabric.
- Configure Azure Key Vault for security.
- Assign users to roles.

Best practices for securing your clusters

Always use a secure cluster:

- Implement cluster security by using certificates.
- Provide client access (admin and read-only) by using Azure Active Directory (Azure AD).

Use automated deployments:

- Use scripts to generate, deploy, and roll over the secrets.
- Store the secrets in Azure Key Vault and use Azure AD for all other client access.
- Require authentication for human access to the secrets.

Additionally, consider the following configuration options:

- Create perimeter networks (also known as demilitarized zones, DMZs, and screened subnets) by using Azure Network Security Groups (NSGs).
- Access cluster virtual machines (VMs) or manage your cluster by using jump servers with Remote Desktop Connection.

Your clusters must be secured to prevent unauthorized users from connecting, especially when a cluster is running

in production. Although it's possible to create an unsecured cluster, anonymous users can connect to your cluster if the cluster exposes management endpoints to the public internet.

There are three [scenarios](#) for implementing cluster security by using various technologies:

- Node-to-node security: This scenario secures communication between the VMs and the computers in the cluster. This form of security ensures that only those computers that are authorized to join the cluster can host applications and services in the cluster. In this scenario, the clusters that run on Azure, or standalone clusters that run on Windows, can use either [certificate security](#) or [Windows security](#) for Windows Server machines.
- Client-to-node security: This scenario secures communication between a Service Fabric client and the individual nodes in the cluster.
- Role-Based Access Control (RBAC): This scenario uses separate identities (certificates, Azure AD, and so on) for each administrator and user client role that accesses the cluster. You specify the role identities when you create the cluster.

NOTE

Security recommendation for Azure clusters: Use Azure AD security to authenticate clients and certificates for node-to-node security.

To configure a standalone Windows cluster, see [Configure settings for a standalone Windows cluster](#).

Use Azure Resource Manager templates and the Service Fabric PowerShell module to create a secure cluster. For step-by-step instructions to create a secure Service Fabric cluster by using Azure Resource Manager templates, see [Creating a Service Fabric cluster](#).

Use the Azure Resource Manager template:

- Customize your cluster by using the template to configure managed storage for VM virtual hard disks (VHDs).
- Drive changes to your resource group by using the template for easy configuration management and auditing.

Treat your cluster configuration as code:

- Be thorough when checking your deployment configurations.
- Avoid using implicit commands to directly modify your resources.

Many aspects of the [Service Fabric application lifecycle](#) can be automated. The [Service Fabric PowerShell module](#) automates common tasks for deploying, upgrading, removing, and testing Azure Service Fabric applications.

Managed APIs and HTTP APIs for application management are also available.

Use X.509 certificates

Always secure your clusters by using X.509 certificates or Windows security. Security is only configured at cluster creation time. It's not possible to turn on security after the cluster is created.

To specify a [cluster certificate](#), set the value of the **ClusterCredentialType** property to X509. To specify a server certificate for outside connections, set the **ServerCredentialType** property to X509.

In addition, follow these practices:

- Create the certificates for production clusters by using a correctly configured Windows Server certificate service. You can also obtain the certificates from an approved certificate authority (CA).
- Never use a temporary or test certificate for production clusters if the certificate was created by using MakeCert.exe or a similar tool.
- Use a self-signed certificate for test clusters, but not for production clusters.

If the cluster is unsecure, anyone can connect to the cluster anonymously and perform management operations. For this reason, always secure production clusters by using X.509 certificates or Windows security.

To learn more about using X.509 certificates, see [Add or remove certificates for a Service Fabric cluster](#).

Configure security policies

Service Fabric also secures the resources that are used by applications. Resources like files, directories, and certificates are stored under the user accounts when the application is deployed. This feature makes running applications more secure from one another, even in a shared hosted environment.

- Use an Active Directory domain group or user: Run the service under the credentials for an Active Directory user or group account. Be sure to use Active Directory on-premises within your domain and not Azure Active Directory. Access other resources in the domain that have been granted permissions by using a domain user or group. For example, resources such as file shares.
- Assign a security access policy for HTTP and HTTPS endpoints: Specify the **SecurityAccessPolicy** property to apply a **RunAs** policy to a service when the service manifest declares endpoint resources with HTTP. Ports allocated to the HTTP endpoints are correctly access-controlled lists for the RunAs user account that the service runs under. When the policy isn't set, http.sys doesn't have access to the service and you can get failures with calls from the client.

To learn how to use security policies in a Service Fabric cluster, see [Configure security policies for your application](#).

Implement the Reliable Actors security configuration

Service Fabric Reliable Actors is an implementation of the actor design pattern. As with any software design pattern, the decision to use a specific pattern is based on whether a software problem fits the pattern.

In general, use the actor design pattern to help model solutions for the following software problems or security scenarios:

- Your problem space involves a large number (thousands or more) of small, independent, and isolated units of state and logic.
- You're working with single-threaded objects that don't require significant interaction from external components, including querying state across a set of actors.
- Your actor instances don't block callers with unpredictable delays by issuing I/O operations.

In Service Fabric, actors are implemented in the Reliable Actors application framework. This framework is based on the actor pattern and built on top of [Service Fabric Reliable Services](#). Each reliable actor service that you write is a partitioned stateful reliable service.

Every actor is defined as an instance of an actor type, identical to the way a .NET object is an instance of a .NET type. For example, an **actor type** that implements the functionality of a calculator can have many actors of that type that are distributed on various nodes across a cluster. Each of the distributed actors is uniquely characterized by an actor identifier.

[Replicator security configurations](#) are used to secure the communication channel that is used during replication. This configuration prevents services from seeing each other's replication traffic and ensures that highly available data is secure. By default, an empty security configuration section prevents replication security. Replicator configurations configure the replicator that is responsible for making the Actor State Provider state highly reliable.

Configure SSL for Azure Service Fabric

The server authentication process [authenticates](#) the cluster management endpoints to a management client. The management client then recognizes that it's talking to the real cluster. This certificate also provides an [SSL](#) for the

HTTPS management API and for Service Fabric Explorer over HTTPS. You must obtain a custom domain name for your cluster. When you request a certificate from a certificate authority, the certificate's subject name must match the custom domain name that you use for your cluster.

To configure SSL for an application, you first need to obtain an SSL certificate that has been signed by a CA. The CA is a trusted third party that issues certificates for SSL security purposes. If you don't already have an SSL certificate, you need to obtain one from a company that sells SSL certificates.

The certificate must meet the following requirements for SSL certificates in Azure:

- The certificate must contain a private key.
- The certificate must be created for key exchange and be exportable to a personal information exchange (.pfx) file.
- The certificate's subject name must match the domain name that is used to access your cloud service.
 - Acquire a custom domain name to use for accessing your cloud service.
 - Request a certificate from a CA with a subject name that matches your service's custom domain name.
For example, if your custom domain name is **contoso.com**, the certificate from your CA should have the subject name **.contoso.com** or **www.contoso.com**.

NOTE

You cannot obtain an SSL certificate from a CA for the **cloudapp.net** domain.

- The certificate must use a minimum of 2,048-bit encryption.

The HTTP protocol is unsecure and subject to eavesdropping attacks. Data that is transmitted over HTTP is sent as plain text from the web browser to the web server or between other endpoints. Attackers can intercept and view sensitive data that is sent via HTTP, such as credit card details and account logins. When data is sent or posted through a browser via HTTPS, SSL ensures that sensitive information is encrypted and secure from interception.

To learn more about using SSL certificates, see [Configure SSL for Azure applications](#).

Use network isolation and security with Azure Service Fabric

Set up a 3 nodetype secure cluster by using the [Azure Resource Manager template](#) as a sample. Control the inbound and outbound network traffic by using the template and Network Security Groups.

The template has an NSG for each of the virtual machine scale sets and is used to control the traffic in and out of the set. The rules are configured by default to allow all traffic necessary for the system services and the application ports specified in the template. Review these rules and make any changes to fit your needs, including adding new rules for your applications.

For more information, see [Common networking scenarios for Azure Service Fabric](#).

Set up Azure Key Vault for security

Service Fabric uses certificates to provide authentication and encryption for securing a cluster and its applications.

Service Fabric uses X.509 certificates to secure a cluster and to provide application security features. You use Azure Key Vault to [manage certificates](#) for Service Fabric clusters in Azure. The Azure resource provider that creates the clusters pulls the certificates from a key vault. The provider then installs the certificates on the VMs when the cluster is deployed on Azure.

A certificate relationship exists between [Azure Key Vault](#), the Service Fabric cluster, and the resource provider that uses the certificates. When the cluster is created, information about the certificate relationship is stored in a key

vault.

There are two basic steps to set up a key vault:

1. Create a resource group specifically for your key vault.

We recommend that you put the key vault in its own resource group. This action helps to prevent the loss of your keys and secrets if other resource groups are removed, such as storage, compute, or the group that contains your cluster. The resource group that contains your key vault must be in the same region as the cluster that is using it.

2. Create a key vault in the new resource group.

The key vault must be enabled for deployment. The compute resource provider can then get the certificates from the vault and install them on the VM instances.

To learn more about how to set up a key vault, see [Get started with Azure Key Vault](#).

Assign users to roles

After you've created the applications to represent your cluster, assign your users to the roles that are supported by Service Fabric: read-only and admin. You can assign these roles by using the Azure portal.

NOTE

For more information about using roles in Service Fabric, see [Role-Based Access Control for Service Fabric clients](#).

Azure Service Fabric supports two access control types for clients that are connected to a [Service Fabric cluster](#): administrator and user. The cluster administrator can use access control to limit access to certain cluster operations for different groups of users. Access control makes the cluster more secure.

Next steps

- Set up your Service Fabric [development environment](#).
- Learn about [Service Fabric support options](#).

Azure Service Fabric security checklist

10/17/2018 • 2 minutes to read • [Edit Online](#)

This article provides an easy-to-use checklist that will help you secure your Azure Service Fabric environment.

Introduction

Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices. Service Fabric also addresses the significant challenges in developing and managing cloud applications. Developers and administrators can avoid complex infrastructure problems and focus on implementing mission-critical, demanding workloads that are scalable, reliable, and manageable.

Checklist

Use the following checklist to help you make sure that you haven't overlooked any important issues in management and configuration of a secure Azure Service Fabric solution.

CHECKLIST CATEGORY	DESCRIPTION
Role based access control (RBAC)	<ul style="list-style-type: none">Access control allows the cluster administrator to limit access to certain cluster operations for different groups of users, making the cluster more secure.Administrators have full access to management capabilities (including read/write capabilities).Users, by default, have only read access to management capabilities (for example, query capabilities), and the ability to resolve applications and services.
X.509 certificates and Service Fabric	<ul style="list-style-type: none">Certificates used in clusters running production workloads should be created by using a correctly configured Windows Server certificate service or obtained from an approved Certificate Authority (CA).Never use any temporary or test certificates in production that are created with tools such as MakeCert.exe.You can use a self-signed certificate but, should only do so for test clusters and not in production.
Cluster Security	<ul style="list-style-type: none">The cluster security scenarios include Node-to-node security, Client-to-node security, Role-based access control (RBAC).
Cluster authentication	<ul style="list-style-type: none">Authenticates node-to-node communication for cluster federation.
Server authentication	<ul style="list-style-type: none">Authenticates the cluster management endpoints to a management client.

CHECKLIST CATEGORY	DESCRIPTION
Application security	<ul style="list-style-type: none"> Encryption and decryption of application configuration values. Encryption of data across nodes during replication.
Cluster Certificate	<ul style="list-style-type: none"> This certificate is required to secure the communication between the nodes on a cluster. Set the thumbprint of the primary certificate in the Thumbprint section and that of the secondary in the ThumbprintSecondary variables.
ServerCertificate	<ul style="list-style-type: none"> This certificate is presented to the client when it tries to connect to this cluster. You can use two different server certificates, a primary and a secondary for upgrade.
ClientCertificateThumbprints	<ul style="list-style-type: none"> This is a set of certificates that you want to install on the authenticated clients.
ClientCertificateCommonNames	<ul style="list-style-type: none"> Set the common name of the first client certificate for the CertificateCommonName. The CertificateIssuerThumbprint is the thumbprint for the issuer of this certificate.
ReverseProxyCertificate	<ul style="list-style-type: none"> This is an optional certificate that can be specified if you want to secure your Reverse Proxy.
Key Vault	<ul style="list-style-type: none"> Used to manage certificates for Service Fabric clusters in Azure.

Next steps

- [Service Fabric Cluster upgrade process and expectations from you](#)
- [Managing your Service Fabric applications in Visual Studio.](#)
- [Service Fabric Health model introduction.](#)

Azure security management and monitoring overview

10/30/2018 • 6 minutes to read • [Edit Online](#)

Azure provides security mechanisms to aid in the management and monitoring of Azure cloud services and virtual machines (VMs). This article provides an overview of these core security features and services. Links are provided to articles that give details of each so you can learn more.

The security of your Microsoft cloud services is a partnership and a shared responsibility between you and Microsoft. Microsoft is responsible for the Azure platform and the physical security of its datacenters (by using security protections such as locked badge-entry doors, fences, and guards). Azure provides strong levels of cloud security at the software layer that meets the security, privacy, and compliance needs of its customers.

You own your data and identities, the responsibility for protecting them, the security of your on-premises resources, and the security of cloud components over which you have control. Microsoft gives you security controls and capabilities to help you protect your data and applications. Your degree of responsibility for security is based on the type of cloud service.

The following chart summarizes the balance of responsibility between Microsoft and the customer.

Responsibility	SaaS	PaaS	IaaS	On-prem
Data governance & rights management	Customer	Customer	Customer	Customer
Client endpoints	Customer	Customer	Customer	Customer
Account & access management	Customer	Customer	Customer	Customer
Identity & directory infrastructure	Microsoft	Microsoft	Customer	Customer
Application	Microsoft	Microsoft	Customer	Customer
Network controls	Microsoft	Microsoft	Customer	Customer
Operating system	Microsoft	Microsoft	Customer	Customer
Physical hosts	Microsoft	Microsoft	Microsoft	Customer
Physical network	Microsoft	Microsoft	Microsoft	Customer
Physical datacenter	Microsoft	Microsoft	Microsoft	Customer
		Microsoft	Customer	

For more information about security management, see [Security management in Azure](#).

Role-Based Access Control

Role-Based Access Control (RBAC) provides detailed access management for Azure resources. By using RBAC, you can grant people only the amount of access that they need to perform their jobs. RBAC can also help you ensure that when people leave the organization, they lose access to resources in the cloud.

Learn more:

- [Active Directory team blog on RBAC](#)
- [Azure Role-Based Access Control](#)

Antimalware

With Azure, you can use antimalware software from major security vendors such as Microsoft, Symantec, Trend Micro, McAfee, and Kaspersky. This software helps protect your virtual machines from malicious files, adware, and other threats.

Microsoft Antimalware for Azure Cloud Services and Virtual Machines offers you the ability to install an antimalware agent for both PaaS roles and virtual machines. Based on System Center Endpoint Protection, this feature brings proven on-premises security technology to the cloud.

We also offer deep integration for Trend's [Deep Security](#) and [SecureCloud](#) products in the Azure platform. Deep Security is an antivirus solution, and SecureCloud is an encryption solution. Deep Security is deployed inside VMs through an extension model. By using the Azure portal UI and PowerShell, you can choose to use Deep Security inside new VMs that are being spun up, or existing VMs that are already deployed.

Symantec Endpoint Protection (SEP) is also supported on Azure. Through portal integration, you can specify that you intend to use SEP on a VM. SEP can be installed on a new VM via the Azure portal, or it can be installed on an existing VM via PowerShell.

Learn more:

- [Deploying Antimalware Solutions on Azure Virtual Machines](#)
- [Microsoft Antimalware for Azure Cloud Services and Virtual Machines](#)
- [How to install and configure Trend Micro Deep Security as a Service on a Windows VM](#)
- [How to install and configure Symantec Endpoint Protection on a Windows VM](#)
- [New Antimalware Options for Protecting Azure Virtual Machines](#)

Multi-Factor Authentication

Azure Multi-Factor Authentication is a method of authentication that requires the use of more than one verification method. It adds a critical second layer of security to user sign-ins and transactions.

Multi-Factor Authentication helps safeguard access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication via a range of verification options (phone call, text message, or mobile app notification or verification code) and third-party OATH tokens.

Learn more:

- [Multi-Factor Authentication](#)
- [What is Azure Multi-Factor Authentication?](#)
- [How Azure Multi-Factor Authentication works](#)

ExpressRoute

You can use Azure ExpressRoute to extend your on-premises networks into the Microsoft Cloud over a dedicated private connection that's facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services such as Azure, Office 365, and CRM Online. Connectivity can be from:

- An any-to-any (IP VPN) network.
- A point-to-point Ethernet network.
- A virtual cross-connection through a connectivity provider at a co-location facility.

ExpressRoute connections don't go over the public internet. They can offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the internet.

Learn more:

- [ExpressRoute technical overview](#)

Virtual network gateways

VPN gateways, also called Azure virtual network gateways, are used to send network traffic between virtual networks and on-premises locations. They are also used to send traffic between multiple virtual networks within Azure (network to network). VPN gateways provide secure cross-premises connectivity between Azure and your infrastructure.

Learn more:

- [About VPN gateways](#)
- [Azure network security overview](#)

Privileged Identity Management

Sometimes users need to carry out privileged operations in Azure resources or other SaaS applications. This often means organizations give them permanent privileged access in Azure Active Directory (Azure AD).

This is a growing security risk for cloud-hosted resources because organizations can't sufficiently monitor what those users are doing with their privileged access. Additionally, if a user account with privileged access is compromised, that one breach can affect an organization's overall cloud security. Azure AD Privileged Identity Management helps to resolve this risk by lowering the exposure time of privileges and increasing visibility into usage.

Privileged Identity Management introduces the concept of a temporary admin for a role or "just in time" administrator access. This kind of admin is a user who needs to complete an activation process for that assigned role. The activation process changes the assignment of the user to a role in Azure AD from inactive to active, for a specified time period.

Learn more:

- [Azure AD Privileged Identity Management](#)
- [Get started with Azure AD Privileged Identity Management](#)

Identity Protection

Azure AD Identity Protection provides a consolidated view of suspicious sign-in activities and potential vulnerabilities to help protect your business. Identity Protection detects suspicious activities for users and privileged (admin) identities, based on signals like:

- Brute-force attacks.
- Leaked credentials.
- Sign-ins from unfamiliar locations and infected devices.

By providing notifications and recommended remediation, Identity Protection helps to mitigate risks in real time. It calculates user risk severity. You can configure risk-based policies to automatically help safeguard application access from future threats.

Learn more:

- [Azure Active Directory Identity Protection](#)
- [Channel 9: Azure AD and Identity Show: Identity Protection Preview](#)

Security Center

Azure Security Center helps you prevent, detect, and respond to threats. Security Center gives you increased visibility into, and control over, the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions. It helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Security Center helps you optimize and monitor the security of your Azure resources by:

- Enabling you to define policies for your Azure subscription resources according to:
 - Your company's security needs.

- The type of applications or sensitivity of the data in each subscription.
- Monitoring the state of your Azure virtual machines, networking, and applications.
- Providing a list of prioritized security alerts, including alerts from integrated partner solutions. It also provides the information that you need to quickly investigate an attack and recommendations on how to remediate it.

Learn more:

- [Introduction to Azure Security Center](#)
- [Improve your secure score in Azure Security Center](#)

Intelligent Security Graph

Intelligent Security Graph provides real-time threat protection in Microsoft products and services. It uses advanced analytics that link a massive amount of threat intelligence and security data to provide insights that can strengthen organizational security. Microsoft uses advanced analytics—processing more than 450 billion authentications per month, scanning 400 billion emails for malware and phishing, and updating one billion devices—to deliver richer insights. These insights can help your organization detect and respond to attacks quickly.

- [Intelligent Security Graph](#)

What is Azure Security Center?

12/13/2018 • 8 minutes to read • [Edit Online](#)

Azure Security Center is a unified infrastructure security management system that strengthens the security posture of your data centers, and provides advanced threat protection across your hybrid workloads in the cloud - whether they're in Azure or not - as well as on premises.

Keeping your resources safe is a joint effort between your cloud provider, Azure, and you, the customer. You have to make sure your workloads are secure as you move to the cloud, and at the same time, when you move to IaaS (infrastructure as a service) there is more customer responsibility than there was in PaaS (platform as a service), and SaaS (software as a service). Azure Security Center provides you the tools needed to harden your network, secure your services and make sure you're on top of your security posture.

Azure Security Center addresses the three most urgent security challenges:

- **Rapidly changing workloads** – It's both a strength and a challenge of the cloud. On the one hand, end users are empowered to do more. On the other, how do you make sure that the ever-changing services people are using and creating are up to your security standards and follow security best practices?
- **Increasingly sophisticated attacks** - Wherever you run your workloads, the attacks keep getting more sophisticated. You have to secure your public cloud workloads, which are, in effect, an Internet facing workload that can leave you even more vulnerable if you don't follow security best practices.
- **Security skills are in short supply** - The number of security alerts and alerting systems far outnumbers the number of administrators with the necessary background and experience far to make sure your environments are protected. Staying up-to-date with the latest attacks is a constant challenge, making it impossible to stay in place while the world of security is an ever-changing front.

To help you protect yourself against these challenges, Security Center provides you with the tools to:

- **Strengthen security posture:** Security Center assesses your environment and enables you to understand the status of your resources, are they secure or not?
- **Protect against threats:** Security Center assesses your workloads and raises threat prevention recommendations and threat detection alerts.
- **Get secure faster:** In Security Center, everything is done in cloud speed. Because it is natively integrated, deployment of Security Center is easy, providing you with autoprovisioning and protection with Azure services.

Architecture

Because Security Center is natively part of Azure, PaaS services in Azure - including Service Fabric, SQL databases, and storage accounts - are monitored and protected by Security Center without necessitating any deployment.

In addition, Security Center protects non-Azure servers and virtual machines in the cloud or on premises, for both Windows and Linux servers, by installing the Microsoft Monitoring Agent on them. Azure virtual machines are auto-provisioned in Security Center.

The events collected from the agents and from Azure are correlated in the security analytics engine to provide you tailored recommendations (hardening tasks), that you should follow to make sure your workloads are secure, and threat detection alerts. You should investigate such alerts as soon as possible to make sure malicious attacks

aren't taking place on your workloads.

Strengthen security posture

Azure Security Center enables you to strengthen your security posture. This means it helps you identify and perform the hardening tasks recommended as security best practices and implement them across your machines, data services, and apps. This includes managing and enforcing your security policies, and making sure your Azure virtual machines, non-Azure servers, and Azure PaaS services are compliant. Security Center provides you with the tools you need to have a bird's eye view on your workloads, with focused visibility on your network security estate.

Manage organization security policy and compliance

It's a security basic to know and make sure your workloads are secure, and it starts with having tailored security policies in place. Because all the policies in Security Center are built on top of Azure policy controls, you're getting the full range and flexibility of a **world-class policy solution**. In Security Center, you can set your policies to run on management groups, across subscriptions, and even for a whole tenant.

Policy Management

Manage the security policies by choosing a subscription or management group from the list below. In order to define additional policies, manage exclusions and advanced settings, go to Azure policies > [Click here to learn more >](#)

7 MANAGEMENT GROUPS 4 SUBSCRIPTIONS 20 WORKSPACES

NAME	POLICY INITIATIVE ASSIGNMENT(S)	COMPLIANCE	COVERAGE	SETTINGS
Rome ILDC - Detection Prod Test 1	ASC Default (subscription: 845d028d-fc71-4c45-b41d-a47b)	32%	Standard	Edit settings >
Visual Studio Enterprise		...	Free	Edit settings >
72f988bf-86f1-41af-91ab-2d7cd011db47 (2 of 8 subscriptions)	Limited permissions	...	Edit settings >	
BenKligerMG (1 of 1 subscriptions)	Limited permissions	...		
ASC DEMO	[Preview]: Enable Monitoring in Azure Security Center	27%	Standard	Edit settings >
Contoso (1 of 7 subscriptions)	Limited permissions	...		
Applications (0 of 5 subscriptions)	Limited permissions	...		
IT (1 of 2 subscriptions)	Limited permissions	...		
Application Team (1 of 1 subscriptions)	Limited permissions	...		
Contoso IT - demo	[Preview]: Enable Monitoring in Azure Security Center	13%	Standard	Edit settings >
Infrastructure Team (0 of 1 subscriptions)	Limited permissions	...		

Security Center helps you **identify Shadow IT subscriptions**. By looking at subscriptions labeled **not covered** in your dashboard, you can know immediately when there are newly created subscriptions and make sure they are covered by your policies, and protected by Azure Security Center.

Policy & compliance

Subscription coverage

CATEGORY	NUMBER
Covered (standard)	3
Covered (free)	1
Not covered	0

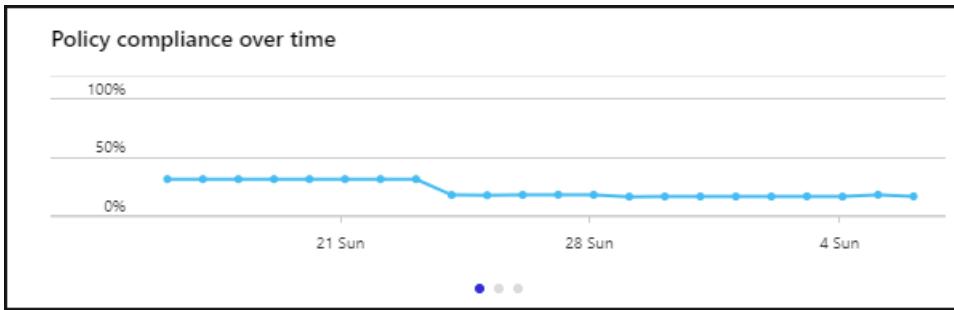
394 Covered resources

Policy compliance

TYPE	PERCENTAGE	DESCRIPTION
Overall compliance	17%	Contoso IT - demo (13%)
Least compliant subscriptions	27%	ASC DEMO (27%)

[Show policy compliance of your environment >](#)

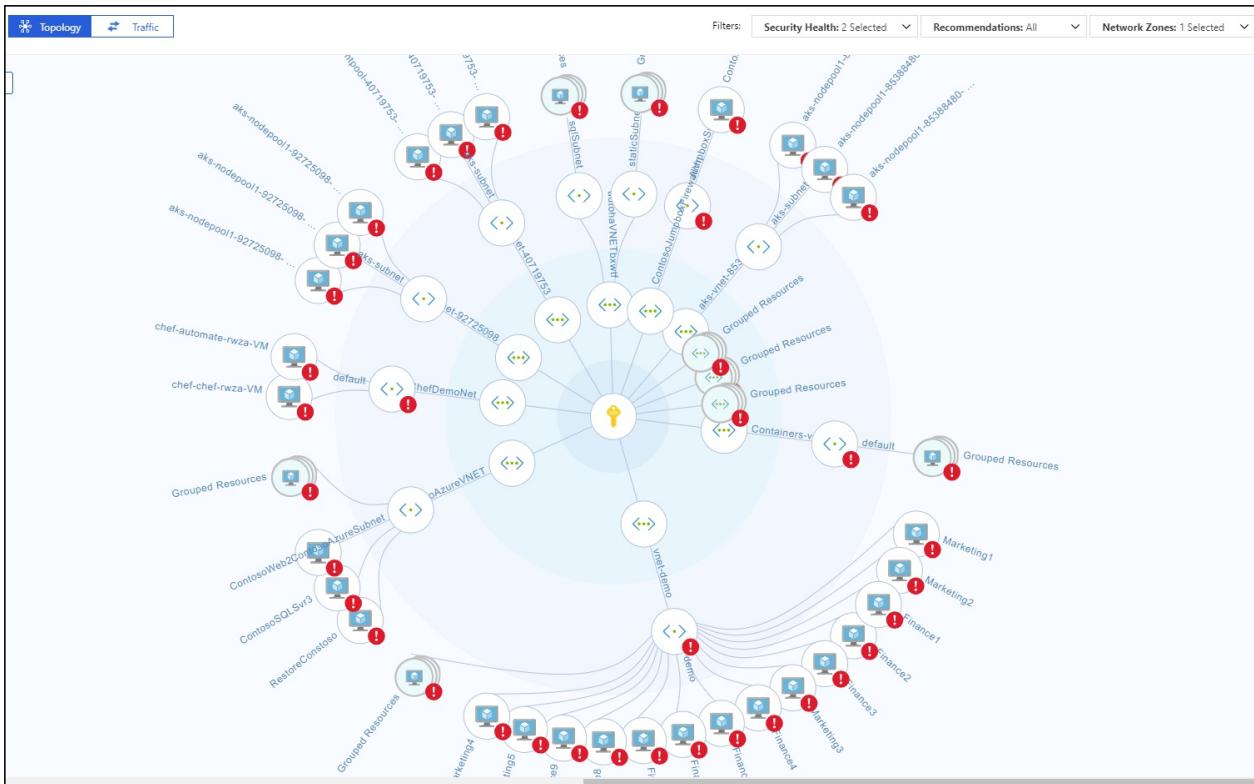
The advanced monitoring capabilities in Security Center also let you **track and manage compliance and governance over time**. The **overall compliance** provides you with a measure of how much your subscriptions are compliant with policies associated with your workload.



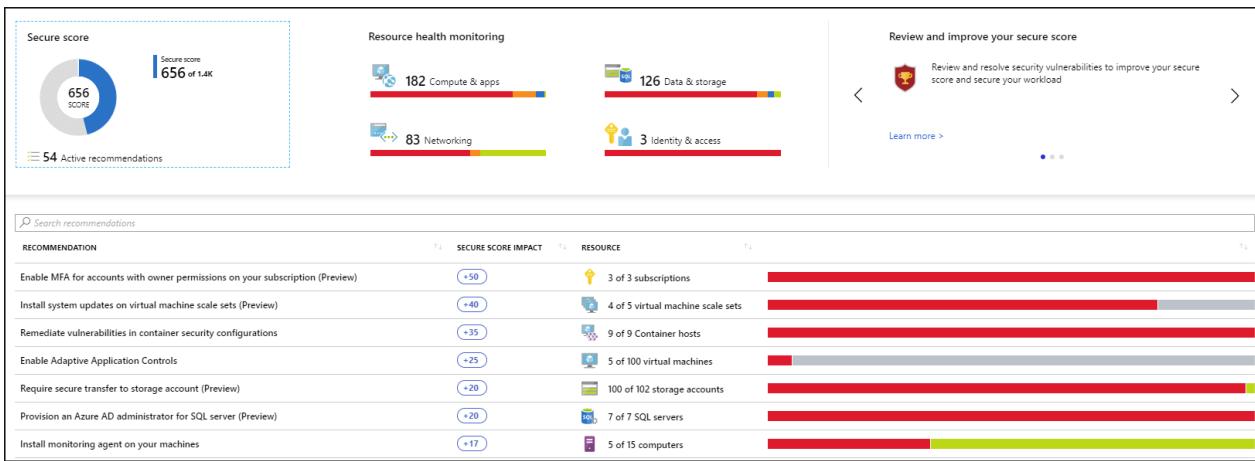
Continuous assessments

Security Center continuously discovers new resources that are being deployed across your workloads and assesses whether they are configured according to security best practices, if not, they're flagged and you get a prioritized list of recommendations for what you need to fix in order to protect your machines.

One of the most powerful tools Security Center provides for continuously monitoring the security status of your network is the **Network map**. The map enables you to see the topology of your workloads, so you can see if each node is properly configured. You can see how your nodes are connected, which helps you block unwanted connections that could potentially make it easier for an attacker to creep along your network.

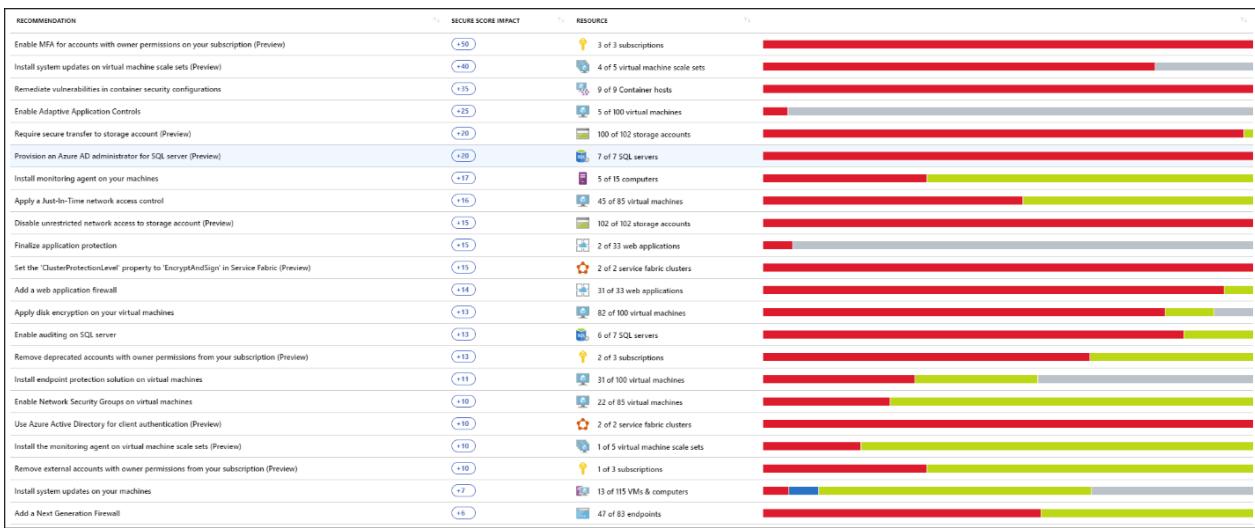


Security Center makes mitigating your security alerts one step easier, by adding a **Secure score**. The secure scores are now associated with each recommendation you receive to help you understand how important each recommendation is to your overall security posture. This is crucial in enabling you to **prioritize your security work**.



Optimize and improve security by configuring recommended controls

The heart of Azure Security Center's value lies in its recommendations. The recommendations are tailored to the particular security concerns found on your workloads, and Security Center does the security admin work for you, by not only finding your vulnerabilities, but providing you with specific instructions for how to get rid of them.



In this way, Security Center enables you not just to set security policies, but to apply secure configuration standards across your resources.

The recommendations help you to reduce the attack surface across each of your resources. That includes Azure virtual machines, non-Azure servers, and Azure PaaS services such as SQL and Storage accounts and more - where each type of resource is assessed differently and has its own standards.

Enable auditing on SQL server

Description
Enable auditing on your SQL Server to track database activities across all databases on the server and save them in an audit log.

General Information

RECOMMENDATION SCORE	2/15
RECOMMENDATION IMPACT	+13
USER IMPACT	Low
IMPLEMENTATION COST	Low

Threats

- Data exfiltration
- Data spillage
- Malicious insider
- Threat resistance

Remediation steps

To enable SQL server auditing:

1. Select the SQL server.
2. Under Auditing, select On.
3. Select Storage details and configure a storage account for the audit log.
4. Click Save.

Unhealthy resources Healthy resources
6 **1**

Unhealthy resources (6) Healthy resources (1) Unscanned resources (0)

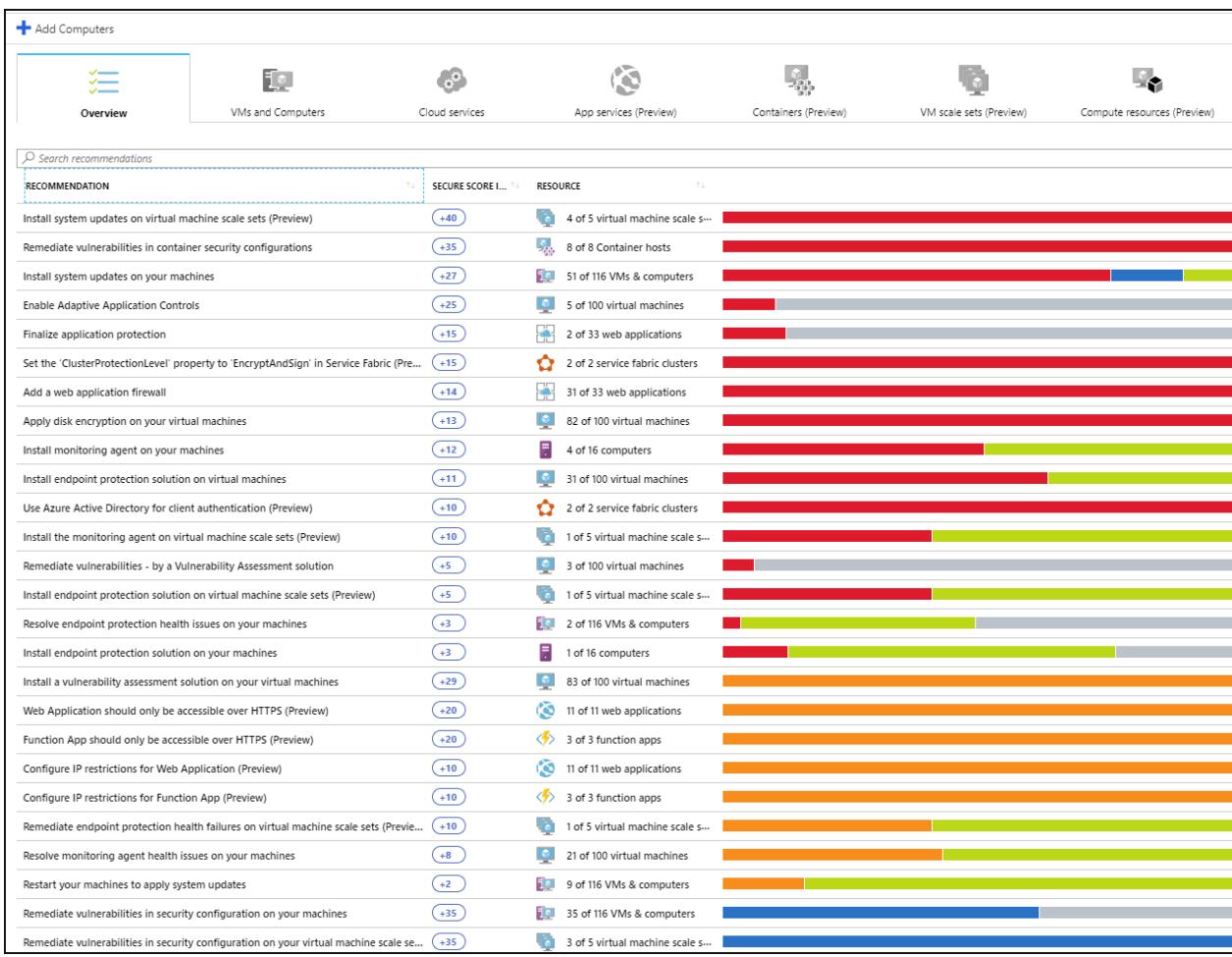
Search SQL servers

NAME	SUBSCRIPTION
sqlserver1ascdemo	ASC DEMO
sqlserver2ascdemo	ASC DEMO
contososerver23	Contoso IT - demo
contosoemps	Contoso IT - demo
soesqldb	Contoso IT - demo
csids	Microsoft Azure Internal Consumption
sqlinjectionalertgenerator	Rome ILDC - Detection Prod Test 1

Protect against threats

Security Center's threat protection enables you to detect and prevent threats at the Infrastructure as a Service (IaaS) layer, non-Azure servers as well as for Platforms as a Service (PaaS) in Azure.

Security Center's threat protection includes fusion kill-chain analysis, which automatically correlates alerts in your environment based on cyber kill-chain analysis, to help you better understand the full story of an attack campaign, where it started and what kind of impact it had on your resources.



Advanced threat protection

With Security Center, you get native integration with Windows Defender Advanced Threat Protection out of the box. This means that without any configuration, your Windows virtual machines and servers are fully integrated with Security Center's recommendations and assessments. Advanced threat detection is also offered out of the box for Linux virtual machines and servers.

In addition, Security Center lets you automate application control policies on server environments. The adaptive application controls in Security Center enable end-to-end app whitelisting across your Windows servers. You don't need to create the rules and check violations, it's all done automatically for you.

Protect PaaS

Security Center helps you detect threats across Azure PaaS services. You can detect threats targeting Azure services including Azure App Service, Azure SQL, Azure Storage Account, and more data services. You can also take advantage of the native integration with Microsoft Cloud App Security's User and Entity Behavioral Analytics (UEBA) to perform anomaly detection on your Azure activity logs.

Block brute force attacks

Security Center helps you limit exposure to brute force attacks. By reducing access to virtual machine ports, using the just-in-time VM access, you can harden your network by preventing unnecessary access. You can set secure access policies on selected ports, for only authorized users, allowed source IP address ranges or IP addresses, and for a limited amount of time.

Security incident detected

Incident Detected

DESCRIPTION	The incident which started on 2018-11-06 01:02:00Z and most recently detected on 2018-11-07 10:02:00Z indicate that an attacker has attacked other resources from your virtual machine vm1
ACTIVITY TIME	Wednesday, November 7, 2018, 12:02:00 PM
SEVERITY	! High
STATE	Active
ATTACKED RESOURCE	vm1
SUBSCRIPTION	ASC DEMO (212f9889-769e-45ae-ab43-6da33674bd26)
DETECTED BY	Microsoft
ACTION TAKEN	Detected
ENVIRONMENT	Azure
REMEDIATION STEPS	<ol style="list-style-type: none"> 1. Escalate the alert to the information security team. 2. Review the remediation steps of each one of the alerts

Alerts included in this incident

DESCRIPTION	COUNT	ACTIVITY TIME	ATTACKED RESOURCE	SEVERITY
SQL injection blocked	1	11/06/18, 3:02 AM	vm1	! Low
Failed RDP Brute Force Attack	1	11/06/18, 4:02 AM	vm1	! Low
Successful RDP brute force attack	1	11/07/18, 4:02 AM	vm1	! High
Suspicious SVCHOST process executed	1	11/07/18, 5:02 AM	vm1	! Low
Multiple Domain Accounts Queried	1	11/07/18, 6:02 AM	vm1	! Low
Network communication with a malicious m...	1	11/07/18, 7:02 AM	vm1	! Medium

Protect data services

Security Center includes capabilities that help you perform automatic classification of your data in Azure SQL. You can also get assessments for potential vulnerabilities across Azure SQL and Storage services, and recommendations for how to mitigate them.

Get secure faster

Native Azure integration (including Azure Policy and Log Analytics) combined with seamless integration with other Microsoft security solutions, such as Microsoft Cloud App Security and Windows Defender Advanced Threat Protection help make sure your security solution is comprehensive as well as simple to onboard and roll out.

In addition, you can extend the full solution beyond Azure to workloads running on other clouds and in on-premises data centers.

Automatically discover and onboard Azure resources

Security Center provides seamless, native integration with Azure and Azure resources. That means that you can pull together a complete security story involving Azure Policy and built-in Security Center policies across all your Azure resources, and make sure that the whole thing is automatically applied to newly discovered resources as you create them in Azure.

Extensive log collection - logs from Windows and Linux are all leveraged in the security analytics engine and used to create recommendations and alerts.

Next steps

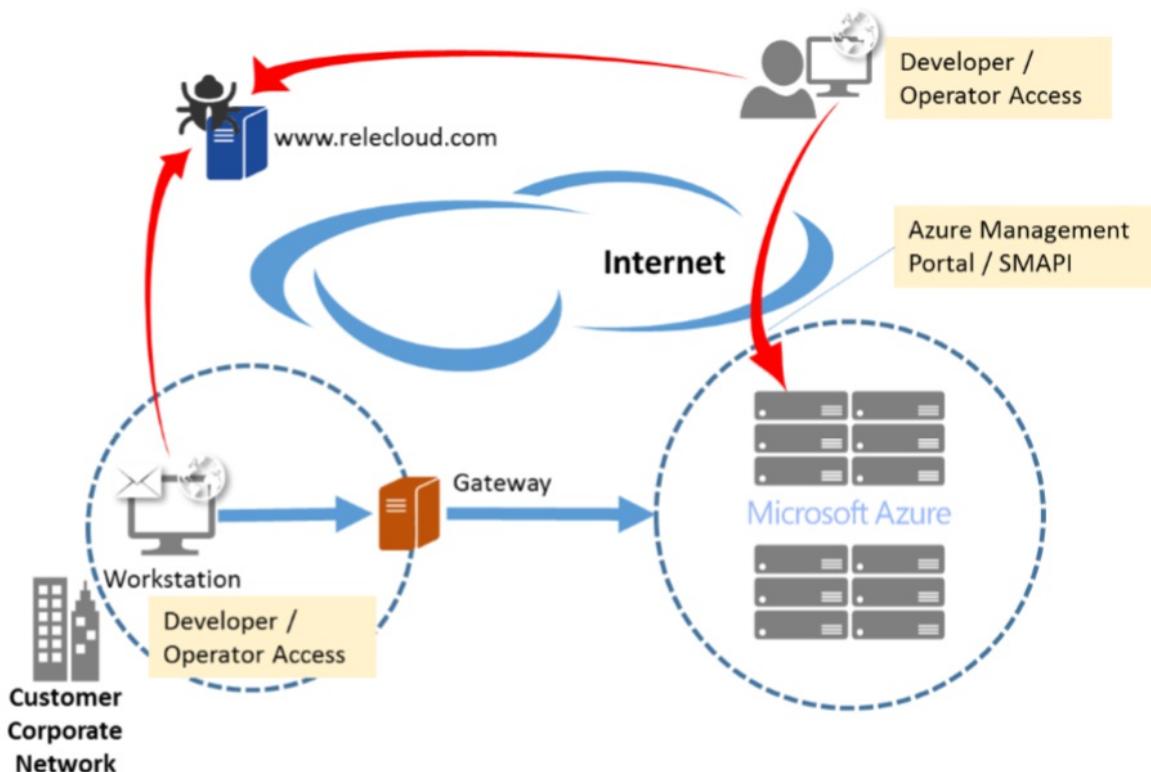
- To get started with Security Center, you need a subscription to Microsoft Azure. If you do not have a subscription, you can sign up for a [free trial](#).
- Security Center's Free pricing tier is enabled with your Azure subscription. To take advantage of advanced security management and threat detection capabilities, you must upgrade to the Standard pricing tier. The Standard tier can be tried for free. See the [Security Center pricing page](#) for more information.
- If you're ready to enable Security Center Standard now, the [Quickstart: Onboard your Azure subscription to Security Center Standard](#) walks you through the steps.

Security management in Azure

11/7/2018 • 20 minutes to read • [Edit Online](#)

Azure subscribers may manage their cloud environments from multiple devices, including management workstations, developer PCs, and even privileged end-user devices that have task-specific permissions. In some cases, administrative functions are performed through web-based consoles such as the [Azure portal](#). In other cases, there may be direct connections to Azure from on-premises systems over Virtual Private Networks (VPNs), Terminal Services, client application protocols, or (programmatically) the Azure Service Management API (SMAPI). Additionally, client endpoints can be either domain joined or isolated and unmanaged, such as tablets or smartphones.

Although multiple access and management capabilities provide a rich set of options, this variability can add significant risk to a cloud deployment. It can be difficult to manage, track, and audit administrative actions. This variability may also introduce security threats through unregulated access to client endpoints that are used for managing cloud services. Using general or personal workstations for developing and managing infrastructure opens unpredictable threat vectors such as web browsing (for example, watering hole attacks) or email (for example, social engineering and phishing).



The potential for attacks increases in this type of environment because it is challenging to construct security policies and mechanisms to appropriately manage access to Azure interfaces (such as SMAPI) from widely varied endpoints.

Remote management threats

Attackers often attempt to gain privileged access by compromising account credentials (for example, through password brute forcing, phishing, and credential harvesting), or by tricking users into running harmful code (for example, from harmful websites with drive-by downloads or from harmful email attachments). In a remotely managed cloud environment, account breaches can lead to an increased risk due to anywhere, anytime access.

Even with tight controls on primary administrator accounts, lower-level user accounts can be used to exploit weaknesses in one's security strategy. Lack of appropriate security training can also lead to breaches through accidental disclosure or exposure of account information.

When a user workstation is also used for administrative tasks, it can be compromised at many different points. Whether a user is browsing the web, using 3rd-party and open-source tools, or opening a harmful document file that contains a trojan.

In general, most targeted attacks that result in data breaches can be traced to browser exploits, plug-ins (such as Flash, PDF, Java), and spear phishing (email) on desktop machines. These machines may have administrative-level or service-level permissions to access live servers or network devices for operations when used for development or management of other assets.

Operational security fundamentals

For more secure management and operations, you can minimize a client's attack surface by reducing the number of possible entry points. This can be done through security principles: "separation of duties" and "segregation of environments."

Isolate sensitive functions from one another to decrease the likelihood that a mistake at one level leads to a breach in another. Examples:

- Administrative tasks should not be combined with activities that might lead to a compromise (for example, malware in an administrator's email that then infects an infrastructure server).
- A workstation used for high-sensitivity operations should not be the same system used for high-risk purposes such as browsing the Internet.

Reduce the system's attack surface by removing unnecessary software. Example:

- Standard administrative, support, or development workstation should not require installation of an email client or other productivity applications if the device's main purpose is to manage cloud services.

Client systems that have administrator access to infrastructure components should be subjected to the strictest possible policy to reduce security risks. Examples:

- Security policies can include Group Policy settings that deny open Internet access from the device and use of a restrictive firewall configuration.
- Use Internet Protocol security (IPsec) VPNs if direct access is needed.
- Configure separate management and development Active Directory domains.
- Isolate and filter management workstation network traffic.
- Use antimalware software.
- Implement multi-factor authentication to reduce the risk of stolen credentials.

Consolidating access resources and eliminating unmanaged endpoints also simplifies management tasks.

Providing security for Azure remote management

Azure provides security mechanisms to aid administrators who manage Azure cloud services and virtual machines. These mechanisms include:

- Authentication and [role-based access control](#).
- Monitoring, logging, and auditing.
- Certificates and encrypted communications.
- A web management portal.
- Network packet filtering.

With client-side security configuration and datacenter deployment of a management gateway, it is possible to restrict and monitor administrator access to cloud applications and data.

NOTE

Certain recommendations in this article may result in increased data, network, or compute resource usage, and may increase your license or subscription costs.

Hardened workstation for management

The goal of hardening a workstation is to eliminate all but the most critical functions required for it to operate, making the potential attack surface as small as possible. System hardening includes minimizing the number of installed services and applications, limiting application execution, restricting network access to only what is needed, and always keeping the system up to date. Furthermore, using a hardened workstation for management segregates administrative tools and activities from other end-user tasks.

Within an on-premises enterprise environment, you can limit the attack surface of your physical infrastructure through dedicated management networks, server rooms that have card access, and workstations that run on protected areas of the network. In a cloud or hybrid IT model, being diligent about secure management services can be more complex because of the lack of physical access to IT resources. Implementing protection solutions requires careful software configuration, security-focused processes, and comprehensive policies.

Using a least-privilege minimized software footprint in a locked-down workstation for cloud management—and for application development—can reduce the risk of security incidents by standardizing the remote management and development environments. A hardened workstation configuration can help prevent the compromise of accounts that are used to manage critical cloud resources by closing many common avenues used by malware and exploits. Specifically, you can use [Windows AppLocker](#) and Hyper-V technology to control and isolate client system behavior and mitigate threats, including email or Internet browsing.

On a hardened workstation, the administrator runs a standard user account (which blocks administrative-level execution) and associated applications are controlled by an allow list. The basic elements of a hardened workstation are as follows:

- Active scanning and patching. Deploy antimalware software, perform regular vulnerability scans, and update all workstations by using the latest security update in a timely fashion.
- Limited functionality. Uninstall any applications that are not needed and disable unnecessary (startup) services.
- Network hardening. Use Windows Firewall rules to allow only valid IP addresses, ports, and URLs related to Azure management. Ensure that inbound remote connections to the workstation are also blocked.
- Execution restriction. Allow only a set of predefined executable files that are needed for management to run (referred to as “default-deny”). By default, users should be denied permission to run any program unless it is explicitly defined in the allow list.
- Least privilege. Management workstation users should not have any administrative privileges on the local machine itself. This way, they cannot change the system configuration or the system files, either intentionally or unintentionally.

You can enforce all this by using [Group Policy Objects](#) (GPOs) in Active Directory Domain Services (AD DS) and applying them through your (local) management domain to all management accounts.

Managing services, applications, and data

Azure cloud services configuration is performed through either the Azure portal or SAPI, via the Windows PowerShell command-line interface or a custom-built application that takes advantage of these RESTful interfaces. Services using these mechanisms include Azure Active Directory (Azure AD), Azure Storage, Azure Websites, and Azure Virtual Network, and others.

Virtual Machine-deployed applications provide their own client tools and interfaces as needed, such as the Microsoft Management Console (MMC), an enterprise management console (such as Microsoft System Center or Windows Intune), or another management application—Microsoft SQL Server Management Studio, for example.

These tools typically reside in an enterprise environment or client network. They may depend on specific network protocols, such as Remote Desktop Protocol (RDP), that require direct, stateful connections. Some may have web-enabled interfaces that should not be openly published or accessible via the Internet.

You can restrict access to infrastructure and platform services management in Azure by using [multi-factor authentication](#), [X.509 management certificates](#), and firewall rules. The Azure portal and SAPI require Transport Layer Security (TLS). However, services and applications that you deploy into Azure require you to take protection measures that are appropriate based on your application. These mechanisms can frequently be enabled more easily through a standardized hardened workstation configuration.

Management gateway

To centralize all administrative access and simplify monitoring and logging, you can deploy a dedicated [Remote Desktop Gateway](#) (RD Gateway) server in your on-premises network, connected to your Azure environment.

A Remote Desktop Gateway is a policy-based RDP proxy service that enforces security requirements. Implementing RD Gateway together with Windows Server Network Access Protection (NAP) helps ensure that only clients that meet specific security health criteria established by Active Directory Domain Services (AD DS) Group Policy objects (GPOs) can connect. In addition:

- Provision an [Azure management certificate](#) on the RD Gateway so that it is the only host allowed to access the Azure portal.
- Join the RD Gateway to the same [management domain](#) as the administrator workstations. This is necessary when you are using a site-to-site IPsec VPN or ExpressRoute within a domain that has a one-way trust to Azure AD, or if you are federating credentials between your on-premises AD DS instance and Azure AD.
- Configure a [client connection authorization policy](#) to let the RD Gateway verify that the client machine name is valid (domain joined) and allowed to access the Azure portal.
- Use IPsec for [Azure VPN](#) to further protect management traffic from eavesdropping and token theft, or consider an isolated Internet link via [Azure ExpressRoute](#).
- Enable multi-factor authentication (via [Azure Multi-Factor Authentication](#)) or smart-card authentication for administrators who log on through RD Gateway.
- Configure source [IP address restrictions](#) or [Network Security Groups](#) in Azure to minimize the number of permitted management endpoints.

Security guidelines

In general, helping to secure administrator workstations for use with the cloud is similar to the practices used for any workstation on-premises—for example, minimized build and restrictive permissions. Some unique aspects of cloud management are more akin to remote or out-of-band enterprise management. These include the use and auditing of credentials, security-enhanced remote access, and threat detection and response.

Authentication

You can use Azure logon restrictions to constrain source IP addresses for accessing administrative tools and audit access requests. To help Azure identify management clients (workstations and/or applications), you can configure both SAPI (via customer-developed tools such as Windows PowerShell cmdlets) and the Azure portal to require client-side management certificates to be installed, in addition to SSL certificates. We also recommend that administrator access require multi-factor authentication.

Some applications or services that you deploy into Azure may have their own authentication mechanisms for both end-user and administrator access, whereas others take full advantage of Azure AD. Depending on whether you are federating credentials via Active Directory Federation Services (AD FS), using directory synchronization or maintaining user accounts solely in the cloud, using [Microsoft Identity Manager](#) (part of Azure AD Premium) helps you manage identity lifecycles between the resources.

Connectivity

Several mechanisms are available to help secure client connections to your Azure virtual networks. Two of these mechanisms, [site-to-site VPN](#) (S2S) and [point-to-site VPN](#) (P2S), enable the use of industry standard IPsec (S2S) or the [Secure Socket Tunneling Protocol](#) (SSTP) (P2S) for encryption and tunneling. When Azure is connecting to public-facing Azure services management such as the Azure portal, Azure requires Hypertext Transfer Protocol Secure (HTTPS).

A stand-alone hardened workstation that does not connect to Azure through an RD Gateway should use the SSTP-based point-to-site VPN to create the initial connection to the Azure Virtual Network, and then establish RDP connection to individual virtual machines from with the VPN tunnel.

Management auditing vs. policy enforcement

Typically, there are two approaches for helping to secure management processes: auditing and policy enforcement. Doing both provides comprehensive controls, but may not be possible in all situations. In addition, each approach has different levels of risk, cost, and effort associated with managing security, particularly as it relates to the level of trust placed in both individuals and system architectures.

Monitoring, logging, and auditing provide a basis for tracking and understanding administrative activities, but it may not always be feasible to audit all actions in complete detail due to the amount of data generated. Auditing the effectiveness of the management policies is a best practice, however.

Policy enforcement that includes strict access controls puts programmatic mechanisms in place that can govern administrator actions, and it helps ensure that all possible protection measures are being used. Logging provides proof of enforcement, in addition to a record of who did what, from where, and when. Logging also enables you to audit and crosscheck information about how administrators follow policies, and it provides evidence of activities

Client configuration

We recommend three primary configurations for a hardened workstation. The biggest differentiators between them are cost, usability, and accessibility, while maintaining a similar security profile across all options. The following table provides a short analysis of the benefits and risks to each. (Note that "corporate PC" refers to a standard desktop PC configuration that would be deployed for all domain users, regardless of roles.)

CONFIGURATION	BENEFITS	CONS
Stand-alone hardened workstation	Tightly controlled workstation	higher cost for dedicated desktops
-	Reduced risk of application exploits	Increased management effort
-	Clear separation of duties	-
Corporate PC as virtual machine	Reduced hardware costs	-
-	Segregation of role and applications	-
Windows to go with BitLocker drive encryption	Compatibility with most PCs	Asset tracking
-	Cost-effectiveness and portability	-
-	Isolated management environment	-

It is important that the hardened workstation is the host and not the guest, with nothing between the host operating system and the hardware. Following the "clean source principle" (also known as "secure origin") means that the host should be the most hardened. Otherwise, the hardened workstation (guest) is subject to attacks on

the system on which it is hosted.

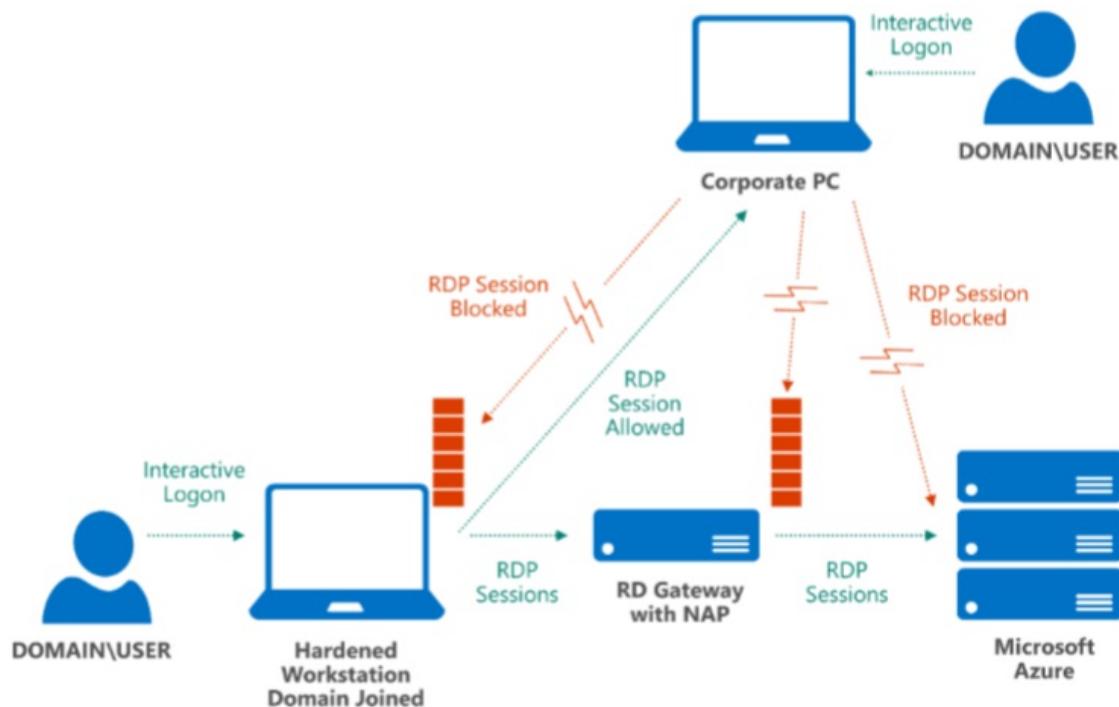
You can further segregate administrative functions through dedicated system images for each hardened workstation that have only the tools and permissions needed for managing select Azure and cloud applications, with specific local AD DS GPOs for the necessary tasks.

For IT environments that have no on-premises infrastructure (for example, no access to a local AD DS instance for GPOs because all servers are in the cloud), a service such as [Microsoft Intune](#) can simplify deploying and maintaining workstation configurations.

Stand-alone hardened workstation for management

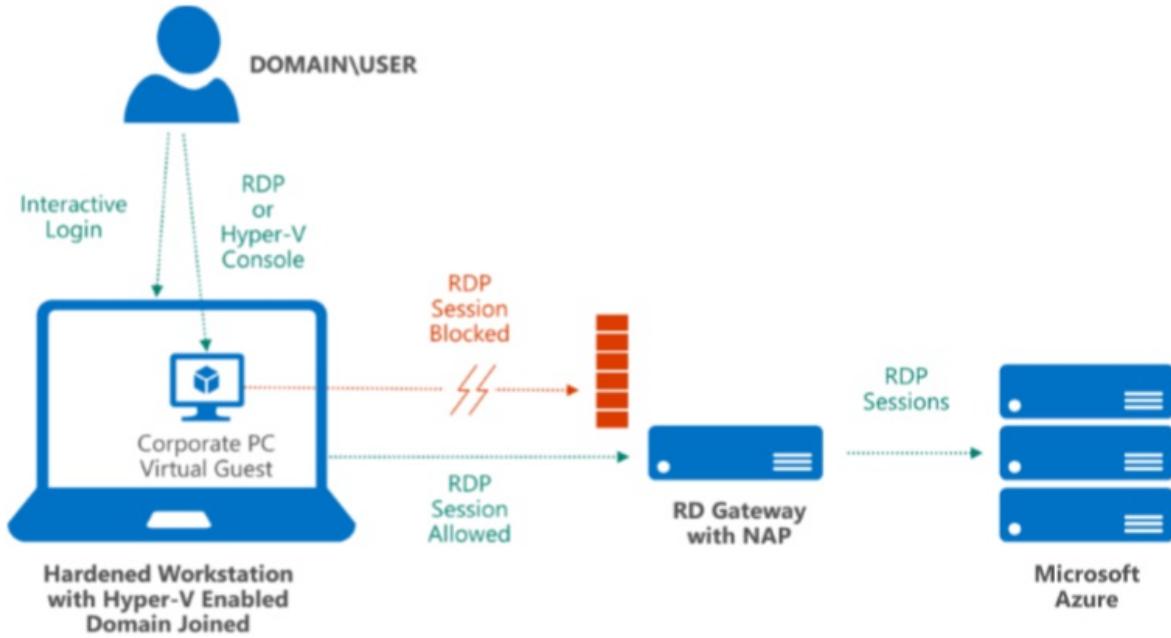
With a stand-alone hardened workstation, administrators have a PC or laptop that they use for administrative tasks and another, separate PC or laptop for non-administrative tasks. A workstation dedicated to managing your Azure services does not need other applications installed. Additionally, using workstations that support a [Trusted Platform Module](#) (TPM) or similar hardware-level cryptography technology aids in device authentication and prevention of certain attacks. TPM can also support full volume protection of the system drive by using [BitLocker Drive Encryption](#).

In the stand-alone hardened workstation scenario (shown below), the local instance of Windows Firewall (or a non-Microsoft client firewall) is configured to block inbound connections, such as RDP. The administrator can log on to the hardened workstation and start an RDP session that connects to Azure after establishing a VPN connect with an Azure Virtual Network, but cannot log on to a corporate PC and use RDP to connect to the hardened workstation itself.



Corporate PC as virtual machine

In cases where a separate stand-alone hardened workstation is cost prohibitive or inconvenient, the hardened workstation can host a virtual machine to perform non-administrative tasks.



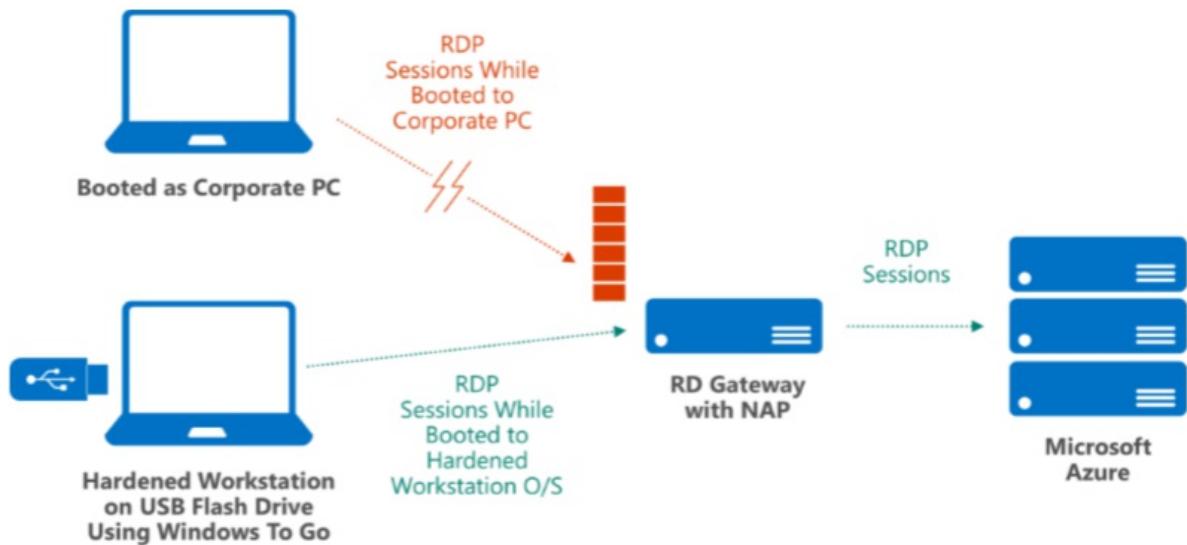
To avoid several security risks that can arise from using one workstation for systems management and other daily work tasks, you can deploy a Windows Hyper-V virtual machine to the hardened workstation. This virtual machine can be used as the corporate PC. The corporate PC environment can remain isolated from the Host, which reduces its attack surface and removes the user's daily activities (such as email) from coexisting with sensitive administrative tasks.

The corporate PC virtual machine runs in a protected space and provides user applications. The host remains a "clean source" and enforces strict network policies in the root operating system (for example, blocking RDP access from the virtual machine).

Windows To Go

Another alternative to requiring a stand-alone hardened workstation is to use a [Windows To Go](#) drive, a feature that supports a client-side USB-boot capability. Windows To Go enables users to boot a compatible PC to an isolated system image running from an encrypted USB flash drive. It provides additional controls for remote-administration endpoints because the image can be fully managed by a corporate IT group, with strict security policies, a minimal OS build, and TPM support.

In the figure below, the portable image is a domain-joined system that is preconfigured to connect only to Azure, requires multi-factor authentication, and blocks all non-management traffic. If a user boots the same PC to the standard corporate image and tries accessing RD Gateway for Azure management tools, the session is blocked. Windows To Go becomes the root-level operating system, and no additional layers are required (host operating system, hypervisor, virtual machine) that may be more vulnerable to outside attacks.



It is important to note that USB flash drives are more easily lost than an average desktop PC. Use of BitLocker to encrypt the entire volume, together with a strong password, makes it less likely that an attacker can use the drive image for harmful purposes. Additionally, if the USB flash drive is lost, revoking and [issuing a new management certificate](#) along with a quick password reset can reduce exposure. Administrative audit logs reside within Azure, not on the client, further reducing potential data loss.

Best practices

Consider the following additional guidelines when you are managing applications and data in Azure.

Dos and don'ts

Don't assume that because a workstation has been locked down that other common security requirements do not need to be met. The potential risk is higher because of elevated access levels that administrator accounts generally possess. Examples of risks and their alternate safe practices are shown in the table below.

DON'T	DO
Don't email credentials for administrator access or other secrets (for example, SSL or management certificates)	Maintain confidentiality by delivering account names and passwords by voice (but not storing them in voice mail), perform a remote installation of client/server certificates (via an encrypted session), download from a protected network share, or distribute by hand via removable media.
-	Proactively manage your management certificate life cycles.
Don't store account passwords unencrypted or un-hashed in application storage (such as in spreadsheets, SharePoint sites, or file shares).	Establish security management principles and system hardening policies, and apply them to your development environment.
-	Use Enhanced Mitigation Experience Toolkit 5.5 certificate pinning rules to ensure proper access to Azure SSL/TLS sites.
Don't share accounts and passwords between administrators, or reuse passwords across multiple user accounts or services, particularly those for social media or other nonadministrative activities.	Create a dedicated Microsoft account to manage your Azure subscription—an account that is not used for personal email.

DON'T	DO
Don't email configuration files.	Configuration files and profiles should be installed from a trusted source (for example, an encrypted USB flash drive), not from a mechanism that can be easily compromised, such as email.
Don't use weak or simple logon passwords.	Enforce strong password policies, expiration cycles (change-on-first-use), console timeouts, and automatic account lockouts. Use a client password management system with multi-factor authentication for password vault access.
Don't expose management ports to the Internet.	Lock down Azure ports and IP addresses to restrict management access. For more information, see the Azure Network Security white paper.
-	Use firewalls, VPNs, and NAP for all management connections.

Azure operations

Within Microsoft's operation of Azure, operations engineers and support personnel who access Azure's production systems use [hardened workstation PCs with VMs](#) provisioned on them for internal corporate network access and applications (such as e-mail, intranet, etc.). All management workstation computers have TPMs, the host boot drive is encrypted with BitLocker, and they are joined to a special organizational unit (OU) in Microsoft's primary corporate domain.

System hardening is enforced through Group Policy, with centralized software updating. For auditing and analysis, event logs (such as security and AppLocker) are collected from management workstations and saved to a central location.

In addition, dedicated jump-boxes on Microsoft's network that require two-factor authentication are used to connect to Azure's production network.

Azure security checklist

Minimizing the number of tasks that administrators can perform on a hardened workstation helps minimize the attack surface in your development and management environment. Use the following technologies to help protect your hardened workstation:

- IE hardening. The Internet Explorer browser (or any web browser, for that matter) is a key entry point for harmful code due to its extensive interactions with external servers. Review your client policies and enforce running in protected mode, disabling add-ons, disabling file downloads, and using [Microsoft SmartScreen](#) filtering. Ensure that security warnings are displayed. Take advantage of Internet zones and create a list of trusted sites for which you have configured reasonable hardening. Block all other sites and in-browser code, such as ActiveX and Java.
- Standard user. Running as a standard user brings a number of benefits, the biggest of which is that stealing administrator credentials via malware becomes more difficult. In addition, a standard user account does not have elevated privileges on the root operating system, and many configuration options and APIs are locked out by default.
- AppLocker. You can use [AppLocker](#) to restrict the programs and scripts that users can run. You can run AppLocker in audit or enforcement mode. By default, AppLocker has an allow rule that enables users who have an admin token to run all code on the client. This rule exists to prevent administrators from locking themselves out, and it applies only to elevated tokens. See also Code Integrity as part of Windows Server [core security](#).
- Code signing. Code signing all tools and scripts used by administrators provides a manageable mechanism for

deploying application lockdown policies. Hashes do not scale with rapid changes to the code, and file paths do not provide a high level of security. You should combine AppLocker rules with a PowerShell [execution policy](#) that only allows specific signed code and scripts to be [executed](#).

- Group Policy. Create a global administrative policy that is applied to any domain workstation that is used for management (and block access from all others), and to user accounts authenticated on those workstations.
- Security-enhanced provisioning. Safeguard your baseline hardened workstation image to help protect against tampering. Use security measures like encryption and isolation to store images, virtual machines, and scripts, and restrict access (perhaps use an auditable check-in/check-out process).
- Patching. Maintain a consistent build (or have separate images for development, operations, and other administrative tasks), scan for changes and malware routinely, keep the build up to date, and only activate machines when they are needed.
- Encryption. Make sure that management workstations have a TPM to more securely enable [Encrypting File System](#) (EFS) and BitLocker. If you are using Windows To Go, use only encrypted USB keys together with BitLocker.
- Governance. Use AD DS GPOs to control all the administrators' Windows interfaces, such as file sharing. Include management workstations in auditing, monitoring, and logging processes. Track all administrator and developer access and usage.

Summary

Using a hardened workstation configuration for administering your Azure cloud services, Virtual Machines, and applications can help you avoid numerous risks and threats that can come from remotely managing critical IT infrastructure. Both Azure and Windows provide mechanisms that you can employ to help protect and control communications, authentication, and client behavior.

Next steps

The following resources are available to provide more general information about Azure and related Microsoft services, in addition to specific items referenced in this paper:

- [Securing Privileged Access](#) – get the technical details for designing and building a secure administrative workstation for Azure management
- [Microsoft Trust Center](#) - learn about Azure platform capabilities that protect the Azure fabric and the workloads that run on Azure
- [Microsoft Security Response Center](#) -- where Microsoft security vulnerabilities, including issues with Azure, can be reported or via email to secure@microsoft.com
- [Azure Security Blog](#) – keep up to date on the latest in Azure Security

Introduction to Azure Log Integration

6/8/2018 • 3 minutes to read • [Edit Online](#)

IMPORTANT

The Azure Log integration feature will be deprecated by 06/01/2019. AzLog downloads will be disabled by Jun 27, 2018. For guidance on what to do moving forward review the post [Use Azure monitor to integrate with SIEM tools](#)

Azure Log Integration was made available to simplify the task of integrating Azure logs with your on-premises Security Information and Event Management (SIEM) system.

The recommended method for integrating Azure logs is to use your SIEM vendor's connectors. Azure Monitor provides the ability to stream the logs into event hubs, and SIEM vendors can write connectors to further integrate logs from the event hub into the SIEM. For a description of how this works, follow the instructions in [Monitor stream monitoring for data event hubs](#). The article also lists the SIEMs for which direct Azure connectors are already available.

IMPORTANT

If your primary interest is collecting virtual machine logs, most SIEM vendors include this option in their solution. Using the SIEM vendor's connector is always the preferred alternative.

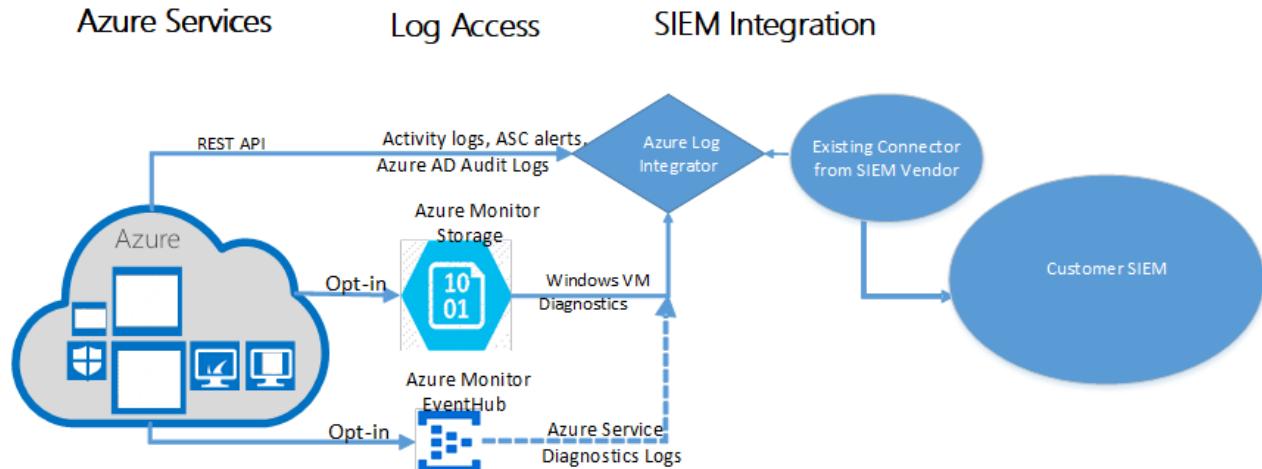
The documentation on the Azure Log Integration feature is still being maintained until the feature is deprecated.

Read further to learn more about the Azure Log Integration feature:

Azure Log Integration collects Windows events from Windows Event Viewer logs, [Azure activity logs](#), [Azure Security Center alerts](#), and [Azure Diagnostics logs](#) from Azure resources. Integration helps your SIEM solution provide a unified dashboard for all your assets, whether on-premises or in the cloud. You can use a dashboard to receive, aggregate, correlate, and analyze alerts for security events.

NOTE

Currently, Azure Log Integration supports only Azure commercial and Azure Government clouds. Other clouds are not supported.



What logs can I integrate?

Azure produces extensive logging for each Azure service. The logs represent three log types:

- **Control/management logs:** Provide visibility into the [Azure Resource Manager](#) CREATE, UPDATE, and DELETE operations. An Azure activity log is an example of this type of log.
- **Data plane logs:** Provide visibility into events that are raised when you use an Azure resource. An example of this type of log is the Windows Event Viewer's **System**, **Security**, and **Application** channels in a Windows virtual machine. Another example is Azure Diagnostics logging, which you configure through Azure Monitor.
- **Processed events:** Provide analyzed event and alert information that are processed for you. An example of this type of event is Azure Security Center alerts. Azure Security Center processes and analyzes your subscription to provide alerts that are relevant to your current security posture.

Azure Log Integration supports ArcSight, QRadar, and Splunk. Check with your SIEM vendor to assess whether the vendor has a native connector. Don't use Azure Log Integration if a native connector is available.

If no other options are available, consider using Azure Log Integration. The following table includes our recommendations:

SIEM	CUSTOMER ALREADY USES THE AZURE LOG INTEGRATOR	CUSTOMER IS INVESTIGATING SIEM INTEGRATION OPTIONS
Splunk	Begin migrating to the Azure Monitor add-on for Splunk .	Use the Splunk connector .
QRadar	Migrate to or begin using the QRadar connector that's documented in the last section of Stream Azure monitoring data to an event hub for consumption by an external tool .	Use the QRadar connector that's documented in the last section of Stream Azure monitoring data to an event hub for consumption by an external tool .
ArcSight	Continue to use the Azure log integrator until a connector is available, and then migrate to the connector-based solution.	Consider using Azure Log Analytics as an alternative. Don't onboard to Azure Log Integration unless you are willing to go through the migration process when the connector becomes available.

NOTE

Although Azure Log Integration is a free solution, there are Azure storage costs associated with log file information storage.

If you need assistance, you can create a [support request](#). For the service, select **Log Integration**.

Next steps

This article introduced you to Azure Log Integration. To learn more about Azure Log Integration and the types of logs that are supported, see the following articles:

- [Get started with Azure Log Integration](#). This tutorial walks you through the installation of Azure Log Integration. It also describes how to integrate logs from Windows Azure Diagnostics (WAD) storage, Azure activity logs, Azure Security Center alerts, and Azure Active Directory audit logs.
- [Azure Log Integration frequently asked questions \(FAQ\)](#). This FAQ answers common questions about Azure Log Integration.
- Learn more about how to [stream Azure monitoring data to an event hub for consumption by an external tool](#).

Azure Log Integration with Azure Diagnostics logging and Windows event forwarding

7/13/2018 • 11 minutes to read • [Edit Online](#)

IMPORTANT

The Azure Log integration feature will be deprecated by 06/01/2019. AzLog downloads will be disabled by Jun 27, 2018. For guidance on what to do moving forward review the post [Use Azure monitor to integrate with SIEM tools](#)

You should only use Azure log integration if an [Azure Monitor](#) connector isn't available from your Security Incident and Event Management (SIEM) vendor.

Azure Log Integration makes Azure logs available to your SIEM so you can create a unified security dashboard for all your assets. For more information about the status of an Azure Monitor connector, contact your SIEM vendor.

IMPORTANT

If your primary interest is collecting virtual machine logs, most SIEM vendors include this option in their solution. Using the SIEM vendor's connector is always the preferred alternative.

This article helps you get started with Azure Log Integration. It focuses on installing the Azure Log Integration service and integrating the service with Azure Diagnostics. The Azure Log Integration service then collects Windows Event Log information from the Windows Security Event channel from virtual machines deployed in an Azure infrastructure as a service. This is similar to *event forwarding* that you might use in an on-premises system.

NOTE

Integrating the output of Azure Log Integration with an SIEM is done by the SIEM itself. For more information, see [Integrate Azure Log Integration with your on-premises SIEM](#).

The Azure Log Integration service runs on either a physical or a virtual computer running Windows Server 2008 R2 or later (Windows Server 2016 or Windows Server 2012 R2 is preferred).

A physical computer can run on-premises or on a hosting site. If you choose to run the Azure Log Integration service on a virtual machine, the virtual machine can be located on-premises or in a public cloud, such as in Microsoft Azure.

The physical or virtual machine running the Azure Log Integration service requires network connectivity to the Azure public cloud. This article provides details about the required configuration.

Prerequisites

At a minimum, installing Azure Log Integration requires the following items:

- An **Azure subscription**. If you don't have one, you can sign up for a [free account](#).
- A **storage account** that can be used for Windows Azure Diagnostics (WAD) logging. You can use a preconfigured storage account or create a new storage account. Later in this article, we describe how to configure the storage account.

NOTE

Depending on your scenario, a storage account might not be required. For the Azure Diagnostics scenario covered in this article, a storage account is required.

- **Two systems:**

- A machine that runs the Azure Log Integration service. This machine collects all the log information that later is imported into your SIEM. This system:
 - Can be on-premises or hosted in Microsoft Azure.
 - Must be running an x64 version of Windows Server 2008 R2 SP1 or later, and have Microsoft .NET 4.5.1 installed. To determine the .NET version installed, see [Determine which .NET Framework versions are installed](#).
 - Must have connectivity to the Azure Storage account that's used for Azure Diagnostics logging. Later in this article, we describe how to confirm connectivity.
- A machine that you want to monitor. This is a VM running as an [Azure virtual machine](#). The logging information from this machine is sent to the Azure Log Integration service machine.

For a quick demonstration of how to create a virtual machine by using the Azure portal, take a look at the following video:

Deployment considerations

During testing, you can use any system that meets the minimum operating system requirements. For a production environment, the load might require you to plan for scaling up or scaling out.

You can run multiple instances of the Azure Log Integration service. However, you can run only one instance of the service per physical or virtual machine. In addition, you can load-balance Azure Diagnostics storage accounts for WAD. The number of subscriptions to provide to the instances is based on your capacity.

NOTE

Currently, we don't have specific recommendations about when to scale out instances of Azure Log Integration machines (that is, machines running the Azure Log Integration service), or for storage accounts or subscriptions. Make scaling decisions based on your performance observations in each of these areas.

To help improve performance, you also have the option to scale up the Azure Log Integration service. The following performance metrics can help you size the machines that you choose to run the Azure Log Integration service:

- On an 8-processor (core) machine, a single instance of Azure Log Integration can process about 24 million events per day (approximately 1 million events per hour).
- On a 4-processor (core) machine, a single instance of Azure Log Integration can process about 1.5 million events per day (approximately 62,500 events per hour).

Install Azure Log Integration

To install Azure Log Integration, download the [Azure Log Integration](#) installation file. Complete the setup process. Choose whether to provide telemetry information to Microsoft.

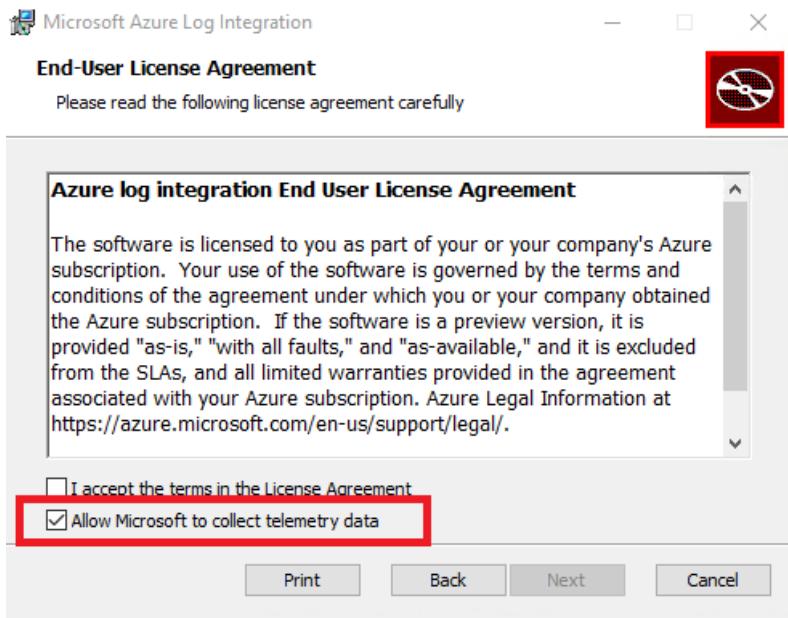
The Azure Log Integration service collects telemetry data from the machine on which it's installed.

Telemetry data that's collected includes the following:

- Exceptions that occur during execution of Azure Log Integration.
- Metrics about the number of queries and events processed.
- Statistics about which Azlog.exe command-line options are used.

NOTE

We recommend that you allow Microsoft to collect telemetry data. You can turn off the collection of telemetry data by clearing the **Allow Microsoft to collect telemetry data** check box.



The installation process is covered in the following video:

Post-installation and validation steps

After you complete basic setup, you're ready to perform post-installation and validation steps:

- Open PowerShell as an administrator. Then, go to C:\Program Files\Microsoft Azure Log Integration.
- Import the Azure Log Integration cmdlets. To import the cmdlets, run the script `LoadAzlogModule.ps1`. Enter `.\LoadAzlogModule.ps1`, and then press Enter (note the use of `\` in this command). You should see something like what appears in the following figure:

```
PS C:\Program Files\Microsoft Azure Log Integration> .\LoadAzLogModule.ps1
List of AzLog Cmdlets. To View again run "Get-Command -Module AzLogD1l"

```

CommandType	Name	Version	Source
Cmdlet	Add-AzLogEventDestination	1.0.6269.5008	AzLogD1l
Cmdlet	Add-AzLogEventRoute	1.0.6269.5008	AzLogD1l
Cmdlet	Add-AzLogEventSource	1.0.6269.5008	AzLogD1l
Cmdlet	Get-AzLogEventDestination	1.0.6269.5008	AzLogD1l
Cmdlet	Get-AzLogEventRoute	1.0.6269.5008	AzLogD1l
Cmdlet	Get-AzLogEventSource	1.0.6269.5008	AzLogD1l
Cmdlet	Remove-AzLogEventDestination	1.0.6269.5008	AzLogD1l
Cmdlet	Remove-AzLogEventRoute	1.0.6269.5008	AzLogD1l
Cmdlet	Remove-AzLogEventSource	1.0.6269.5008	AzLogD1l
Cmdlet	Set-AzLogAzureEnvironment	1.0.6269.5008	AzLogD1l

3. Next, configure Azure Log Integration to use a specific Azure environment. An *Azure environment* is the type of Azure cloud datacenter that you want to work with. Although there are several Azure environments, currently, the relevant options are either **AzureCloud** or **AzureUSGovernment**. Running PowerShell as an administrator, make sure that you are in C:\Program Files\Microsoft Azure Log Integration. Then, run this command:

```
Set-AzlogAzureEnvironment -Name AzureCloud (for AzureCloud)
```

If you want to use the US Government Azure cloud, use **AzureUSGovernment** for the **-Name** variable. Currently, other Azure clouds aren't supported.

NOTE

You don't receive feedback when the command succeeds.

4. Before you can monitor a system, you need the name of the storage account that's used for Azure Diagnostics. In the Azure portal, go to **Virtual machines**. Look for a Windows virtual machine that you will monitor. In the **Properties** section, select **Diagnostic Settings**. Then, select **Agent**. Make note of the storage account name that's specified. You need this account name for a later step.

Overview Performance counters Logs Crash dumps Sinks Agent Boot diagnostics

Configure additional options for the Azure Diagnostics agent.

* Storage account ⓘ

vrm01diag967



Disk quota (MB): ⓘ

5120



Diagnostic infrastructure logs: ⓘ

Disabled Enabled

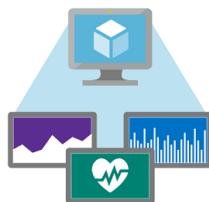
Log level: ⓘ

Error



 Save  Discard

Overview Performance counters Logs Crash dumps Sinks Agent Boot diagnostics



Azure Monitoring collects host-level metrics – like CPU utilization, disk and network usage – for all virtual machines without any additional software. For more insight into this virtual machine, you can collect guest-level metrics, logs, and other diagnostic data using the Azure Diagnostics agent. You can also send diagnostic data to other services like Application Insights. [Learn more](#)

To get started now, click the button below:

[Enable guest-level monitoring](#)

Already know what you're doing? You can customize the diagnostic data you want to collect by visiting each of the tabs above. You can add or remove data types to collect at any time.

NOTE

If monitoring wasn't enabled when the virtual machine was created, you can enable it as shown in the preceding image.

5. Now, go back to the Azure Log Integration machine. Verify that you have connectivity to the storage account from the system where you installed Azure Log Integration. The computer running the Azure Log Integration service needs access to the storage account to retrieve information that's logged by Azure Diagnostics on each of the monitored systems. To verify connectivity:
 - a. [Download Azure Storage Explorer](#).
 - b. Complete setup.
 - c. When installation is finished, select **Next**. Leave the **Launch Microsoft Azure Storage Explorer** check box selected.
 - d. Sign in to Azure.
 - e. Verify that you can see the storage account that you configured for Azure Diagnostics:

The screenshot shows the Microsoft Azure Storage Explorer interface. At the top, there's a toolbar with 'Edit', 'View', and 'Help' options. Below that is a search bar labeled 'Search for resources'. The main area is titled 'Microsoft Azure' and shows a navigation bar with icons for file operations. Underneath, there's a 'Collapse All' button. A tree view displays storage accounts. Under '(Local and Attached)', there's one entry for 'Storage Accounts'. Under 'Free Trial (barclay.neira@gmail.com)', there are two entries for 'Storage Accounts'. One of these accounts, 'azlogtest', is expanded, showing blob containers 'Blob Containers', file shares 'File Shares', queues 'Queues', and tables 'Tables'. Other entries under 'azlogtest' include 'testlogsdia572', 'testlogsdia862', 'testlogsdisks508', and 'testlogsdisks802'. A magnifying glass icon is visible next to the table section of the expanded 'azlogtest' node.

- a. A few options appear under storage accounts. Under **Tables**, you should see a table called **WADWindowsEventLogsTable**.

If monitoring wasn't enabled when the virtual machine was created, you can enable it, as described earlier.

Integrate Windows VM Logs

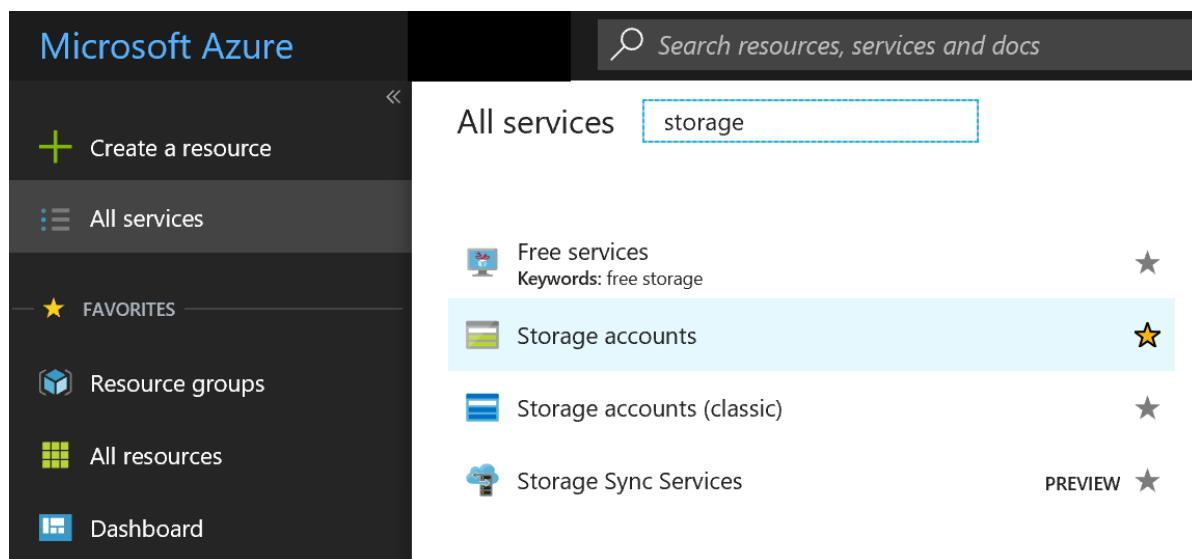
In this step, you configure the machine running the Azure Log Integration service to connect to the storage account that contains the log files.

To complete this step, you need a few things:

- **FriendlyNameForSource**: A friendly name that you can apply to the storage account that you've configured for the virtual machine to store information from Azure Diagnostics.
- **StorageAccountName**: The name of the storage account that you specified when you configured Azure Diagnostics.
- **StorageKey**: The storage key for the storage account where the Azure Diagnostics information is stored for this virtual machine.

To obtain the storage key, complete the following steps:

1. Go to the [Azure portal](#).
2. In the navigation pane, select **All services**.
3. In the **Filter** box, enter **Storage**. Then, select **Storage accounts**.



4. A list of storage accounts appears. Double-click the account that you assigned to log storage.

Storage accounts

Microsoft



Add

Columns



Refresh

Subscriptions: 1 of 6 selected – Don't see a subscription? [Switch directories](#)

Filter by name...

Microsoft Azure Internal C

5 items

NAME ▾



ad2041centralus



ad2041southcentralus



alirgdiag523



alirgdisks684



ascrgjuly20164543

5. Under **Settings**, select **Access keys**.

The screenshot shows the Azure Storage account overview page for the account 'alirgdiag523'. The top navigation bar includes a search bar labeled 'Search (Ctrl+I)'. The left sidebar contains several navigation items: 'Overview' (selected), 'Activity log', 'Access control (IAM)', 'Tags', and 'Diagnose and solve problems'. Below this, under 'SETTINGS', is a list of options: 'Access keys' (highlighted with a red box), 'Configuration', 'Shared access signature', 'Properties', 'Locks', and 'Automation script'. Further down, under 'BLOB SERVICE', are 'Containers' and 'CORS'.

6. Copy **key1**, and then save it in a secure location that you can access for the following step.
7. On the server where you installed Azure Log Integration, open a Command Prompt window as an administrator. (Be sure to open a Command Prompt window as an administrator, and not PowerShell).
8. Go to C:\Program Files\Microsoft Azure Log Integration.
9. Run this command: `Azlog source add <FriendlyNameForTheSource> WAD <StorageAccountName> <StorageKey>`.

Example:

```
Azlog source add Azlogtest WAD Azlog9414  
fxxxxFxxxxxxxxwoEJK2xxxxxxxxxxxxJ+xVJx6m/X5SQDYc4Wpjpli9S9Mm+vXS2RVYtp1mes0t9H5cuqXEw==
```

If you want the subscription ID to show up in the event XML, append the subscription ID to the friendly name:

```
Azlog source add <FriendlyNameForTheSource>.<SubscriptionID> WAD <StorageAccountName> <StorageKey>
```

Example:

```
Azlog source add Azlogtest.YourSubscriptionID WAD Azlog9414  
fxxxxFxxxxxxxxwoEJK2xxxxxxxxxxxxJ+xVJx6m/X5SQDYc4Wpjpli9S9Mm+vXS2RVYtp1mes0t9H5cuqXEw==
```

NOTE

Wait up to 60 minutes, and then view the events that are pulled from the storage account. To view the events, in Azure Log Integration, select **Event Viewer > Windows Logs > Forwarded Events**.

The following video covers the preceding steps:

If data isn't showing up in the Forwarded Events folder

If data isn't showing up in the Forwarded Events folder after an hour, complete these steps:

1. Check the machine that's running the Azure Log Integration service. Confirm that it can access Azure. To test connectivity, in a browser, try to go to the [Azure portal](#).
2. Make sure that the user account Azlog has write permission for the folder users\Azlog.
 - a. Open File Explorer.
 - b. Go to C:\users.
 - c. Right-click C:\users\Azlog.
 - d. Select **Security**.
 - e. Select **NT Service\Azlog**. Check the permissions for the account. If the account is missing from this tab, or if the appropriate permissions aren't showing, you can grant the account permissions on this tab.
3. When you run the command `Azlog source list`, make sure that the storage account that was added in the command `Azlog source add` is listed in the output.
4. To see if any errors are reported from the Azure Log Integration service, go to **Event Viewer > Windows Logs > Application**.

If you run into any issues during installation and configuration, you can create a [support request](#). For the service, select **Log Integration**.

Another support option is the [Azure Log Integration MSDN forum](#). In the MSDN forum, the community can provide support by answering questions and sharing tips and tricks about how to get the most out of Azure Log Integration. The Azure Log Integration team also monitors this forum. They help whenever they can.

Integrate Azure activity logs

The Azure Activity Log is a subscription log that provides insight into subscription-level events that have occurred in Azure. This includes a range of data, from Azure Resource Manager operational data to updates on Service Health events. The Azure Security Center Alerts are also included in this Log.

NOTE

Before you attempt the steps in this article, you must review the [Get started](#) article and complete the steps there.

Steps to integrate Azure Activity logs

1. Open the command prompt and run this command: `cd c:\Program Files\Microsoft Azure Log Integration`
2. Run this command: `azlog createazureid`

This command prompts you for your Azure login. The command then creates an Azure Active Directory service principal in the Azure AD tenants that host the Azure subscriptions in which the logged-in user is an

administrator, a co-administrator, or an owner. The command will fail if the logged-in user is only a guest user in the Azure AD tenant. Authentication to Azure is done through Azure AD. Creating a service principal for Azure Log Integration creates the Azure AD identity that is given access to read from Azure subscriptions.

- Run the following command to authorize the Azure Log Integration service principal created in the previous step access to the read the Activity Log for the subscription. You need to be an Owner on the subscription to run the command.

Azlog.exe authorize subscriptionId Example:

```
AZLOG.exe authorize ba2c2367-d24b-4a32-17b5-4443234859
```

- Check the following folders to confirm that the Azure Active Directory audit log JSON files are created in them:

- C:\Users\azlog\AzureResourceManagerJson
- C:\Users\azlog\AzureResourceManagerJsonLD

NOTE

For specific instructions on bringing the information in the JSON files into your security information and event management (SIEM) system, contact your SIEM vendor.

Community assistance is available through the [Azure Log Integration MSDN Forum](#). This forum enables people in the Azure Log Integration community to support each other with questions, answers, tips, and tricks. In addition, the Azure Log Integration team monitors this forum and helps whenever it can.

You can also open a [support request](#). Select Log Integration as the service for which you are requesting support.

Next steps

To learn more about Azure Log Integration, see the following articles: Before you attempt the steps in this article, you must review the Get started article and complete the steps there.

- [Azure Log Integration for Azure logs](#). The Download Center includes details, system requirements, and installation instructions for Azure Log Integration.
- [Introduction to Azure Log Integration](#). This article introduces you to Azure Log Integration, its key capabilities, and how it works.
- [Partner configuration steps](#). This blog post shows you how to configure Azure Log Integration to work with partner solutions Splunk, HP ArcSight, and IBM QRadar. It describes our current guidance about how to configure the SIEM components. Check with your SIEM vendor for additional details.
- [Azure Log Integration frequently asked questions \(FAQ\)](#). This FAQ answers common questions about Azure Log Integration.
- [Integrating Azure Security Center alerts with Azure Log Integration](#). This article shows you how to sync Security Center alerts and virtual machine security events that are collected by Azure Diagnostics and Azure activity logs. You sync the logs by using your Azure Log Analytics or SIEM solution.
- [New features for Azure Diagnostics and Azure audit logs](#). This blog post introduces you to Azure audit logs and other features that can help you gain insight into the operations of your Azure resources.

Integrate Azure Active Directory audit logs

11/26/2018 • 2 minutes to read • [Edit Online](#)

Azure Active Directory (Azure AD) audit events help you identify privileged actions that occurred in Azure Active Directory. You can see the types of events that you can track by reviewing [Azure Active Directory audit report events](#).

IMPORTANT

The Azure Log integration feature will be deprecated by 06/01/2019. AzLog downloads will be disabled by Jun 27, 2018. For guidance on what to do moving forward review the post [Use Azure monitor to integrate with SIEM tools](#)

Steps to integrate Azure Active Directory audit logs

NOTE

Before you attempt the steps in this article, you must review the [Get started](#) article and complete the relevant steps there.

1. Open the command prompt and run this command:

```
cd c:\Program Files\Microsoft Azure Log Integration
```

2. Run this command:

```
azlog createazureid
```

This command prompts you for your Azure login. The command then creates an Azure Active Directory service principal in the Azure AD tenants that host the Azure subscriptions in which the logged-in user is an administrator, a co-administrator, or an owner. The command will fail if the logged-in user is only a guest user in the Azure AD tenant. Authentication to Azure is done through Azure AD. Creating a service principal for Azure Log Integration creates the Azure AD identity that is given access to read from Azure subscriptions.

3. Run the following command to provide your tenant ID. You need to be member of the tenant admin role to run the command.

```
Azlog.exe authorizedirectoryreader tenantId
```

Example:

```
AZLOG.exe authorizedirectoryreader ba2c0000-d24b-4f4e-92b1-48c4469999
```

4. Check the following folders to confirm that the Azure Active Directory audit log JSON files are created in them:

- **C:\Users\azlog\AzureActiveDirectoryJson**
- **C:\Users\azlog\AzureActiveDirectoryJsonLD**

The following video demonstrates the steps covered in this article:

NOTE

For specific instructions on bringing the information in the JSON files into your security information and event management (SIEM) system, contact your SIEM vendor.

Community assistance is available through the [Azure Log Integration MSDN Forum](#). This forum enables people in the Azure Log Integration community to support each other with questions, answers, tips, and tricks. In addition, the Azure Log Integration team monitors this forum and helps whenever it can.

You can also open a [support request](#). Select **Log Integration** as the service for which you are requesting support.

Next steps

To learn more about Azure Log Integration, see:

- [Microsoft Azure Log Integration for Azure logs](#): This Download Center page gives details, system requirements, and installation instructions for Azure Log Integration.
- [Introduction to Azure Log Integration](#): This article introduces you to Azure Log Integration, its key capabilities, and how it works.
- [Azure Log Integration FAQ](#): This article answers questions about Azure Log Integration.
- [New features for Azure Diagnostics and Azure audit logs](#): This blog post introduces you to Azure audit logs and other features that help you gain insights into the operations of your Azure resources.

Azure Security data export to SIEM-Pipeline Configuration

11/26/2018 • 2 minutes to read • [Edit Online](#)

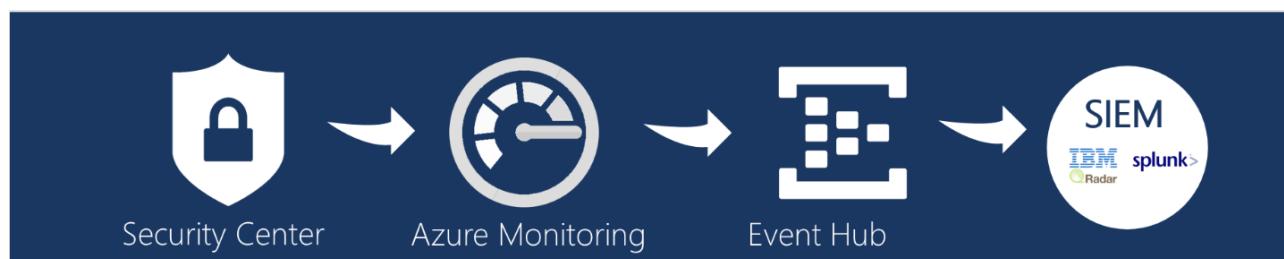
This document details the procedure to export Azure Security Center security data to a SIEM.

Processed events produced by Azure Security Center are published to the Azure [Activity log](#), one of the log types available through Azure Monitor. Azure Monitor offers a consolidated pipeline for routing any of your monitoring data into a SIEM tool. This is done by streaming that data to an Event Hub where it can then be pulled into a partner tool.

This pipe uses the [Azure Monitoring single pipeline](#) for getting access to the monitoring data from your Azure environment. This enables you to easily set up SIEMs and monitoring tools to consume the data.

The next sections describe how you can configure data to be streamed to an event hub. The steps assume that you already have Azure Security Center configured in your Azure subscription.

High-level overview



What is the Azure security data exposed to SIEM?

In this version we expose the [security alerts](#). In upcoming releases, we will enrich the data set with security recommendations.

How to setup the pipeline

Create an Event Hub

Before you begin, you need to [create an Event Hubs namespace](#). This namespace and Event Hub is the destination for all your monitoring data.

Stream the Azure Activity Log to Event Hubs

Please refer to the following article [stream activity log to Event Hubs](#)

Install a partner SIEM connector

Routing your monitoring data to an Event Hub with Azure Monitor enables you to easily integrate with partner SIEM and monitoring tools.

Refer to the following link to see the list of [supported SIEMs](#)

Example for Querying data

Here is a couple of Splunk queries that you can use to pull alert data:

DESCRIPTION OF QUERY	QUERY
All Alerts	index=main Microsoft.Security/locations/alerts
Summarize count of operations by their name	index=main sourcetype="am:security" table operationName stats count by operationName
Get Alerts info: Time, Name, State, ID, and Subscription	index=main Microsoft.Security/locations/alerts table _time, properties.eventName, State, properties.operationId, am_subscriptionId

Next steps

- [Supported SIEMs](#)
- [Stream activity log to Event Hubs](#)
- [Security alerts.](#)

Azure Log Integration tutorial: Process Azure Key Vault events by using Event Hubs

8/2/2018 • 7 minutes to read • [Edit Online](#)

IMPORTANT

The Azure Log integration feature will be deprecated by 06/01/2019. AzLog downloads will be disabled by Jun 27, 2018. For guidance on what to do moving forward review the post [Use Azure monitor to integrate with SIEM tools](#)

You can use Azure Log Integration to retrieve logged events and make them available to your security information and event management (SIEM) system. This tutorial shows an example of how Azure Log Integration can be used to process logs that are acquired through Azure Event Hubs.

The preferred method for integrating Azure logs is by using your SIEM vendor's Azure Monitor connector and following these [instructions](#). However, if your SIEM vendor doesn't provide a connector to Azure Monitor, you may be able to use Azure Log Integration as a temporary solution (if your SIEM is supported by Azure Log Integration) until such a connector is available.

Use this tutorial to get acquainted with how Azure Log Integration and Event Hubs work together by following the example steps and understanding how each step supports the solution. Then you can take what you've learned here to create your own steps to support your company's unique requirements.

WARNING

The steps and commands in this tutorial are not intended to be copied and pasted. They're examples only. Do not use the PowerShell commands "as is" in your live environment. You must customize them based on your unique environment.

This tutorial walks you through the process of taking Azure Key Vault activity logged to an event hub and making it available as JSON files to your SIEM system. You can then configure your SIEM system to process the JSON files.

NOTE

Most of the steps in this tutorial involve configuring key vaults, storage accounts, and event hubs. The specific Azure Log Integration steps are at the end of this tutorial. Do not perform these steps in a production environment. They are intended for a lab environment only. You must customize the steps before using them in production.

Information provided along the way helps you understand the reasons behind each step. Links to other articles give you more detail on certain topics.

For more information about the services that this tutorial mentions, see:

- [Azure Key Vault](#)
- [Azure Event Hubs](#)
- [Azure Log Integration](#)

Initial setup

Before you can complete the steps in this article, you need the following:

1. An Azure subscription and account on that subscription with administrator rights. If you don't have a subscription, you can create a [free account](#).
2. A system with access to the internet that meets the requirements for installing Azure Log Integration. The system can be on a cloud service or hosted on-premises.
3. [Azure Log Integration](#) installed. To install it:
 - a. Use Remote Desktop to connect to the system mentioned in step 2.
 - b. Copy the Azure Log Integration installer to the system. You can [download the installation files](#).
 - c. Start the installer and accept the Microsoft Software License Terms.
 - d. If you will provide telemetry information, leave the check box selected. If you'd rather not send usage information to Microsoft, clear the check box.

For more information about Azure Log Integration and how to install it, see [Azure Log Integration with Azure Diagnostics logging and Windows Event Forwarding](#).

4. The latest PowerShell version.

If you have Windows Server 2016 installed, then you have at least PowerShell 5.0. If you're using any other version of Windows Server, you might have an earlier version of PowerShell installed. You can check the version by entering `get-host` in a PowerShell window. If you don't have PowerShell 5.0 installed, you can [download it](#).

After you have at least PowerShell 5.0, you can proceed to install the latest version:

- a. In a PowerShell window, enter the `Install-Module Azure` command. Complete the installation steps.
- b. Enter the `Install-Module AzureRM` command. Complete the installation steps.

For more information, see [Install Azure PowerShell](#).

Create supporting infrastructure elements

1. Open an elevated PowerShell window and go to **C:\Program Files\Microsoft Azure Log Integration**.
2. Import the AzLog cmdlets by running the script LoadAzLogModule.ps1. Enter the `.\\LoadAzLogModule.ps1` command. (Notice the "\\" in that command.) You should see something like this:

```
PS C:\Program Files\Microsoft Azure Log Integration> .\LoadAzLogModule.ps1
List of AzLog Cmdlets. To View again run "Get-Command -Module AzLogD1l"

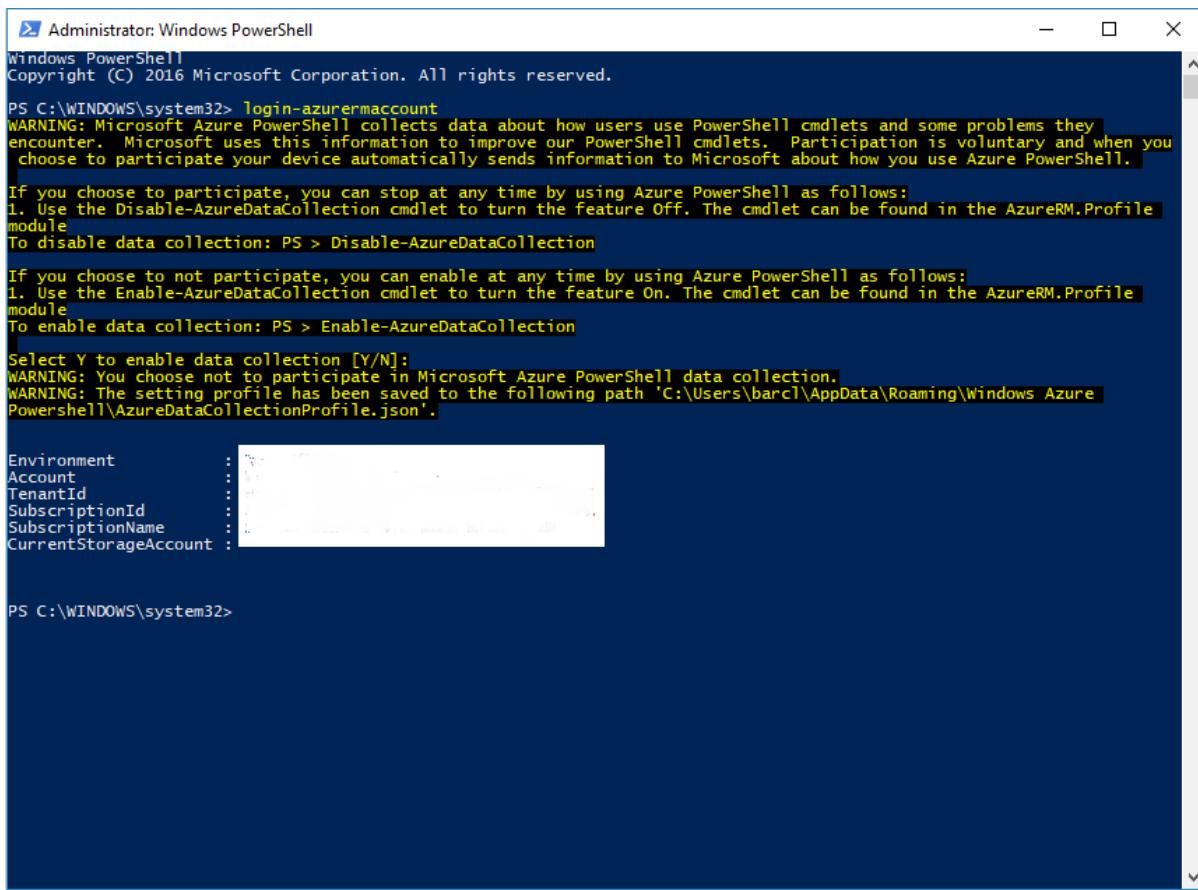
CommandType      Name          Version      Source
----           --          --          --
Cmdlet        Add-AzLogEventDestination 1.0.6269.5008 AzLogD1l
Cmdlet        Add-AzLogEventRoute     1.0.6269.5008 AzLogD1l
Cmdlet        Add-AzLogEventSource   1.0.6269.5008 AzLogD1l
Cmdlet        Get-AzLogEventDestination 1.0.6269.5008 AzLogD1l
Cmdlet        Get-AzLogEventRoute     1.0.6269.5008 AzLogD1l
Cmdlet        Get-AzLogEventSource   1.0.6269.5008 AzLogD1l
Cmdlet        Remove-AzLogEventDestination 1.0.6269.5008 AzLogD1l
Cmdlet        Remove-AzLogEventRoute     1.0.6269.5008 AzLogD1l
Cmdlet        Remove-AzLogEventSource   1.0.6269.5008 AzLogD1l
Cmdlet        Set-AzLogAzureEnvironment 1.0.6269.5008 AzLogD1l
```

3. Enter the `Connect-AzureRmAccount` command. In the login window, enter the credential information for the subscription that you will use for this tutorial.

NOTE

If this is the first time that you're logging in to Azure from this machine, you will see a message about allowing Microsoft to collect PowerShell usage data. We recommend that you enable this data collection because it will be used to improve Azure PowerShell.

4. After successful authentication, you're logged in and you see the information in the following screenshot. Take note of the subscription ID and subscription name, because you'll need them to complete later steps.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> Login-AzureRmAccount
WARNING: Microsoft Azure PowerShell collects data about how users use PowerShell cmdlets and some problems they encounter. Microsoft uses this information to improve our PowerShell cmdlets. Participation is voluntary and when you choose to participate your device automatically sends information to Microsoft about how you use Azure PowerShell.

If you choose to participate, you can stop at any time by using Azure PowerShell as follows:
1. Use the Disable-AzureDataCollection cmdlet to turn the feature Off. The cmdlet can be found in the AzureRM.Profile module
To disable data collection: PS > Disable-AzureDataCollection

If you choose to not participate, you can enable at any time by using Azure PowerShell as follows:
1. Use the Enable-AzureDataCollection cmdlet to turn the feature On. The cmdlet can be found in the AzureRM.Profile module
To enable data collection: PS > Enable-AzureDataCollection

Select Y to enable data collection [Y/N]:
WARNING: You choose not to participate in Microsoft Azure PowerShell data collection.
WARNING: The setting profile has been saved to the following path 'C:\Users\barcl\AppData\Roaming\Windows Azure Powershell\AzureDataCollectionProfile.json'.

Environment      : AzureCloud
Account         : barcl
TenantId        : 52f8e3d4-9c3a-4a0d-9a0a-0a0a0a0a0a0a
SubscriptionId  : 00000000-0000-0000-0000-000000000000
SubscriptionName : barcl
CurrentStorageAccount : barcl

PS C:\WINDOWS\system32>
```

5. Create variables to store values that will be used later. Enter each of the following PowerShell lines. You might need to adjust the values to match your environment.

- `$subscriptionName = 'Visual Studio Ultimate with MSDN'` (Your subscription name might be different. You can see it as part of the output of the previous command.)
- `$location = 'West US'` (This variable will be used to pass the location where resources should be created. You can change this variable to be any location of your choosing.)
- `$random = Get-Random`
- `$name = 'azlogtest' + $random` (The name can be anything, but it should include only lowercase letters and numbers.)
- `$storageName = $name` (This variable will be used for the storage account name.)
- `$rgname = $name` (This variable will be used for the resource group name.)
- `$eventHubNameSpaceName = $name` (This is the name of the event hub namespace.)

6. Specify the subscription that you will be working with:

```
Select-AzureRmSubscription -SubscriptionName $subscriptionName
```

7. Create a resource group:

```
$rg = New-AzureRmResourceGroup -Name $rgname -Location $location
```

If you enter `$rg` at this point, you should see output similar to this screenshot:



```
ResourceGroupName : azlogtest76
Location        : westus
ProvisioningState : Succeeded
Tags            :
ResourceId      : /subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/azlogtest76
```

8. Create a storage account that will be used to keep track of state information:

```
$storage = New-AzureRmStorageAccount -ResourceGroupName $rgname -Name $storagename -Location $location -  
SkuName Standard_LRS
```

9. Create the event hub namespace. This is required to create an event hub.

```
$eventHubNameSpace = New-AzureRmEventHubNamespace -ResourceGroupName $rgname -NamespaceName  
$eventHubnamespaceName -Location $location
```

10. Get the rule ID that will be used with the insights provider:

```
$sbruleid = $eventHubNameSpace.Id + '/authorizationrules/RootManageSharedAccessKey'
```

11. Get all possible Azure locations and add the names to a variable that can be used in a later step:

```
a. $locationObjects = Get-AzureRMLocation  
b. $locations = @('global') + $locationObjects.location
```

If you enter `$locations` at this point, you see the location names without the additional information returned by `Get-AzureRmLocation`.

12. Create an Azure Resource Manager log profile:

```
Add-AzureRmLogProfile -Name $name -ServiceBusRuleId $sbruleid -Locations $locations
```

For more information about the Azure log profile, see [Overview of the Azure Activity Log](#).

NOTE

You might get an error message when you try to create a log profile. You can then review the documentation for `Get-AzureRmLogProfile` and `Remove-AzureRmLogProfile`. If you run `Get-AzureRmLogProfile`, you see information about the log profile. You can delete the existing log profile by entering the `Remove-AzureRmLogProfile -name 'Log Profile Name'` command.

```
Add-AzureRmLogProfile : Exception type: CloudException, Message: The limit of 1 log profiles was reached. To create  
new log profile 'azlogtest2102253579', delete an existing one., Code: Conflict, Status code:Conflict, Reason phrase:  
Conflict  
At line:1 char:1  
+ Add-AzureRmLogProfile -Name $name -ServiceBusRuleId $sbruleid -Locati ...  
+ ~~~~~  
+ CategoryInfo          : CloseError: (:) [Add-AzureRmLogProfile], PSInvalidOperationException  
+ FullyQualifiedErrorId : Microsoft.Azure.Commands.Insights.LogProfiles.AddAzureRmLogProfileCommand
```

Create a key vault

1. Create the key vault:

```
$kv = New-AzureRmKeyVault -VaultName $name -ResourceGroupName $rgname -Location $location
```

2. Configure logging for the key vault:

```
Set-AzureRmDiagnosticSetting -ResourceId $kv.ResourceId -ServiceBusRuleId $sbruleid -Enabled $true
```

Generate log activity

Requests need to be sent to Key Vault to generate log activity. Actions like key generation, storing secrets, or reading secrets from Key Vault will create log entries.

1. Display the current storage keys:

```
Get-AzureRmStorageAccountKey -Name $storagename -ResourceGroupName $rgname | ft -a
```

2. Generate a new **key2**:

```
New-AzureRmStorageAccountKey -Name $storagename -ResourceGroupName $rgname -KeyName key2
```

3. Display the keys again and see that **key2** holds a different value:

```
Get-AzureRmStorageAccountKey -Name $storagename -ResourceGroupName $rgname | ft -a
```

4. Set and read a secret to generate additional log entries:

a.

```
Set-AzureKeyVaultSecret -VaultName $name -Name TestSecret -SecretValue (ConvertTo-SecureString -String 'Hi There!' -AsPlainText -Force)
```

b. (Get-AzureKeyVaultSecret -VaultName \$name -Name TestSecret).SecretValueText

```
PS C:\Program Files\Microsoft Azure Log Integration> Set-AzureKeyVaultSecret -VaultName $name -Name TestSecret -SecretValue (ConvertTo-SecureString -String 'Hi There!' -AsPlainText -Force)

Vault Name   : azlogtest76
Name         : TestSecret
Version      : c43159516c2445c485361638fb6d6c16
Id           : https://azlogtest76.vault.azure.net:443/secrets/TestSecret/c43159516c2445c485361638fb6d6c16
Enabled       : True
Expires      :
Not Before   :
Created       : 5/22/2017 8:36:35 PM
Updated       : 5/22/2017 8:36:35 PM
Content Type  :
Tags          :
```

Configure Azure Log Integration

Now that you have configured all the required elements to have Key Vault logging to an event hub, you need to configure Azure Log Integration:

1. \$storage = Get-AzureRmStorageAccount -ResourceGroupName \$rgname -Name \$storagename
2. \$eventHubKey = Get-AzureRmEventHubNamespaceKey -ResourceGroupName \$rgname -NamespaceName \$eventHubNamespace.name -AuthorizationRuleName RootManageSharedAccessKey
3. \$storagekeys = Get-AzureRmStorageAccountKey -ResourceGroupName \$rgname -Name \$storagename
4. \$storagekey = \$storagekeys[0].Value

Run the AzLog command for each event hub:

1. \$eventhubs = Get-AzureRmEventHub -ResourceGroupName \$rgname -NamespaceName \$eventHubNamespaceName \$eventhubs.Name | %{\$sub = Add-AzLogEventSource -Name \$sub -StorageAccount \$storage.StorageAccountName -StorageKey \$storagekey -EventHubConnectionString \$eventHubKey.PrimaryConnectionString -EventHubName \$_}
2. \$eventhubs

After a minute or so of running the last two commands, you should see JSON files being generated. You can confirm that by monitoring the directory **C:\users\AzLog\EventHubJson**.

Next steps

- [Azure Log Integration FAQ](#)
- [Get started with Azure Log Integration](#)
- [Integrate logs from Azure resources into your SIEM systems](#)

Azure Log Integration FAQ

6/8/2018 • 4 minutes to read • [Edit Online](#)

This article answers frequently asked questions (FAQ) about Azure Log Integration.

IMPORTANT

The Azure Log integration feature will be deprecated by 06/01/2019. AzLog downloads will be disabled by Jun 27, 2018. For guidance on what to do moving forward review the post [Use Azure monitor to integrate with SIEM tools](#)

Azure Log Integration is a Windows operating system service that you can use to integrate raw logs from your Azure resources into your on-premises security information and event management (SIEM) systems. This integration provides a unified dashboard for all your assets, on-premises or in the cloud. You can then aggregate, correlate, analyze, and alert for security events associated with your applications.

The preferred method for integrating Azure logs is by using your SIEM vendor's Azure Monitor connector and following these [instructions](#). However, if your SIEM vendor doesn't provide a connector to Azure Monitor, you may be able to use Azure Log Integration as a temporary solution (if your SIEM is supported by Azure Log Integration) until such a connector is available.

Is the Azure Log Integration software free?

Yes. There is no charge for the Azure Log Integration software.

Where is Azure Log Integration available?

It is currently available in Azure Commercial and Azure Government and is not available in China or Germany.

How can I see the storage accounts from which Azure Log Integration is pulling Azure VM logs?

Run the command **AzLog source list**.

How can I tell which subscription the Azure Log Integration logs are from?

In the case of audit logs that are placed in the **AzureResourcemanagerJson** directories, the subscription ID is in the log file name. This is also true for logs in the **AzureSecurityCenterJson** folder. For example:

20170407T070805_2768037.0000000023.**1111e5ee-1111-111b-a11e-1e111e1111dc.json**

Azure Active Directory audit logs include the tenant ID as part of the name.

Diagnostic logs that are read from an event hub do not include the subscription ID as part of the name. Instead, they include the friendly name specified as part of the creation of the event hub source.

How can I update the proxy configuration?

If your proxy setting does not allow Azure storage access directly, open the **AZLOG.EXE.CONFIG** file in **c:\Program Files\Microsoft Azure Log Integration**. Update the file to include the **defaultProxy** section with

the proxy address of your organization. After the update is done, stop and start the service by using the commands **net stop AzLog** and **net start AzLog**.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.net>
    <connectionManagement>
      <add address="*" maxconnection="400" />
    </connectionManagement>
    <defaultProxy>
      <proxy usesystemdefault="true"
        proxyaddress="http://127.0.0.1:8888"
        bypassonlocal="true" />
    </defaultProxy>
  </system.net>
  <system.diagnostics>
    <performanceCounters filemappingsize="20971520" />
  </system.diagnostics>
```

How can I see the subscription information in Windows events?

Append the subscription ID to the friendly name while adding the source:

```
Azlog source add <sourcefriendlyname>.<subscription id> <StorageName> <StorageKey>
```

The event XML has the following metadata, including the subscription ID:

```

SubjectDomainName WORKGROUP
SubjectLogonId 0x3e7
TargetUserId S-1-5-18
TargetUserName SYSTEM
TargetDomainName NT AUTHORITY
TargetLogonId 0x3e7
LogonType 5
LogonProcessName Advapi
AuthenticationPackageName Negotiate
WorkstationName
LogonGuid {00000000-0000-0000-0000-000000000000}
TransmittedServices -
LmPackageName -
KeyLength 0
ProcessId 0x234
ProcessName C:\Windows\System32\services.exe
IpAddress -
IpPort -
ImpersonationLevel %%%1833
- UserData
  - AzureSielIntegration
    SubscriptionId 00000000-0000-0000-0000-000000000000
    RoleName IaaS
    RoleInstanceId _azsiemdemo
    SourceStorageAccount azsiem9414
    SourceFriendlyName azsiem9414.SLAMDataAnalysis

```

Error messages

When I run the command `AzLog createazureid`, why do I get the following error?

Error:

Failed to create AAD Application - Tenant 72f988bf-86f1-41af-91ab-2d7cd011db37 - Reason = 'Forbidden' - Message = 'Insufficient privileges to complete the operation.'

The **azlog createazureid** command attempts to create a service principal in all the Azure AD tenants for the subscriptions that the Azure login has access to. If your Azure login is only a guest user in that Azure AD tenant, the command fails with "Insufficient privileges to complete the operation." Ask the tenant admin to add your account as a user in the tenant.

When I run the command `azlog authorize`, why do I get the following error?

Error:

Warning creating Role Assignment - AuthorizationFailed: The client janedo@microsoft.com' with object id 'fe9e03e4-4dad-4328-910f-fd24a9660bd2' does not have authorization to perform action 'Microsoft.Authorization/roleAssignments/write' over scope '/subscriptions/70d95299-d689-4c97-b971-0d8ff0000000'.

The **azlog authorize** command assigns the role of reader to the Azure AD service principal (created with **azlog**)

createazureid) to the provided subscriptions. If the Azure login is not a co-administrator or an owner of the subscription, it fails with an "Authorization Failed" error message. Azure Role-Based Access Control (RBAC) of co-administrator or owner is needed to complete this action.

Where can I find the definition of the properties in the audit log?

See:

- [Audit operations with Azure Resource Manager](#)
- [List the management events in a subscription in the Azure Monitor REST API](#)

Where can I find details on Azure Security Center alerts?

See [Managing and responding to security alerts in Azure Security Center](#).

How can I modify what is collected with VM diagnostics?

For details on how to get, modify, and set the Azure Diagnostics configuration, see [Use PowerShell to enable Azure Diagnostics in a virtual machine running Windows](#).

The following example gets the Azure Diagnostics configuration:

```
-AzureRmVMDiagnosticsExtension -ResourceGroupName AzLog-Integration -VMName AzlogClient
$publicsettings = (Get-AzureRmVMDiagnosticsExtension -ResourceGroupName AzLog-Integration -VMName
AzlogClient).PublicSettings
$encodedconfig = (ConvertFrom-Json -InputObject $publicsettings).xmlCfg
$xmlconfig = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($encodedconfig))
Write-Host $xmlconfig

$xmlconfig | Out-File -Encoding utf8 -FilePath "d:\WADConfig.xml"
```

The following example modifies the Azure Diagnostics configuration. In this configuration, only event ID 4624 and event ID 4625 are collected from the security event log. Microsoft Antimalware for Azure events are collected from the system event log. For details on the use of XPath expressions, see [Consuming Events](#).

```
<WindowsEventLog scheduledTransferPeriod="PT1M">
  <DataSource name="Security!*[System[(EventID=4624 or EventID=4625)]]" />
  <DataSource name="System!*[System[Provider[@Name='Microsoft Antimalware']]]" />
</WindowsEventLog>
```

The following example sets the Azure Diagnostics configuration:

```
$diagnosticsconfig_path = "d:\WADConfig.xml"
Set-AzureRmVMDiagnosticsExtension -ResourceGroupName AzLog-Integration -VMName AzlogClient -
DiagnosticsConfigurationPath $diagnosticsconfig_path -StorageAccountName log3121 -StorageAccountKey <storage
key>
```

After you make changes, check the storage account to ensure that the correct events are collected.

If you have any issues during the installation and configuration, please open a [support request](#). Select **Log Integration** as the service for which you are requesting support.

Can I use Azure Log Integration to integrate Network Watcher logs into my SIEM?

Azure Network Watcher generates large quantities of logging information. These logs are not meant to be sent to

a SIEM. The only supported destination for Network Watcher logs is a storage account. Azure Log Integration does not support reading these logs and making them available to a SIEM.

Azure operational security overview

11/7/2018 • 11 minutes to read • [Edit Online](#)

[Azure operational security](#) refers to the services, controls, and features available to users for protecting their data, applications, and other assets in Microsoft Azure. It's a framework that incorporates the knowledge gained through a variety of capabilities that are unique to Microsoft. These capabilities include the Microsoft Security Development Lifecycle (SDL), the Microsoft Security Response Center program, and deep awareness of the cybersecurity threat landscape.

Azure management services

An IT operations team is responsible for managing datacenter infrastructure, applications, and data, including the stability and security of these systems. However, gaining security insights across increasing complex IT environments often requires organizations to cobble together data from multiple security and management systems.

[Microsoft Log Analytics](#) is a cloud-based IT management solution that helps you manage and protect your on-premises and cloud infrastructure. Its core functionality is provided by the following services that run in Azure. Azure includes multiple services that help you manage and protect your on-premises and cloud infrastructure. Each service provides a specific management function. You can combine services to achieve different management scenarios.

Azure Monitor

[Azure Monitor](#) collects data from managed sources into central data stores. This data can include events, performance data, or custom data provided through the API. After the data is collected, it's available for alerting, analysis, and export.

You can consolidate data from a variety of sources and combine data from your Azure services with your existing on-premises environment. Log Analytics also clearly separates the collection of the data from the action taken on that data, so that all actions are available to all kinds of data.

Automation

[Azure Automation](#) provides a way for you to automate the manual, long-running, error-prone, and frequently repeated tasks that are commonly performed in a cloud and enterprise environment. It saves time and increases the reliability of administrative tasks. It even schedules these tasks to be automatically performed at regular intervals. You can automate processes by using runbooks or automate configuration management by using Desired State Configuration.

Backup

[Azure Backup](#) is the Azure-based service that you can use to back up (or protect) and restore your data in the Microsoft Cloud. Azure Backup replaces your existing on-premises or off-site backup solution with a cloud-based solution that's reliable, secure, and cost-competitive.

Azure Backup offers components that you download and deploy on the appropriate computer or server, or in the cloud. The component, or agent, that you deploy depends on what you want to protect. All Azure Backup components (whether you're protecting data on-premises or in the cloud) can be used to back up data to an Azure Recovery Services vault in Azure.

For more information, see the [Azure Backup components table](#).

Site Recovery

[Azure Site Recovery](#) provides business continuity by orchestrating the replication of on-premises virtual and physical machines to Azure, or to a secondary site. If your primary site is unavailable, you fail over to the secondary location so that users can keep working. You fail back when systems return to working order. Use Azure Security Center to perform more intelligent and effective threat detection.

Azure Active Directory

[Azure Active Directory \(Azure AD\)](#) is a comprehensive identity service that:

- Enables identity and access management (IAM) as a cloud service.
- Provides central access management, single sign-on (SSO), and reporting.
- Supports integrated access management for [thousands of applications](#) in the Azure Marketplace, including Salesforce, Google Apps, Box, and Concur.

Azure AD also includes a full suite of [identity management capabilities](#), including these:

- [Multi-factor authentication](#)
- [Self-service password management](#)
- [Self-service group management](#)
- [Privileged account management](#)
- [Role-based access control](#)
- [Application usage monitoring](#)
- [Rich auditing](#)
- [Security monitoring and alerting](#)

With Azure Active Directory, all applications that you publish for your partners and customers (business or consumer) have the same identity and access management capabilities. This enables you to significantly reduce your operational costs.

Azure Security Center

[Azure Security Center](#) helps you prevent, detect, and respond to threats with increased visibility into (and control over) the security of your Azure resources. It provides integrated security monitoring and policy management across your subscriptions. It helps detect threats that might otherwise go unnoticed, and it works with a broad ecosystem of security solutions.

Safeguard virtual machine (VM) data in Azure by providing visibility into your virtual machine's security settings and monitoring for threats. Security Center can monitor your virtual machines for:

- Operating system security settings with the recommended configuration rules.
- System security and critical updates that are missing.
- Endpoint protection recommendations.
- Disk encryption validation.
- Network-based attacks.

Security Center uses [Role-Based Access Control \(RBAC\)](#). RBAC provides [built-in roles](#) that can be assigned to users, groups, and services in Azure.

Security Center assesses the configuration of your resources to identify security issues and vulnerabilities. In Security Center, you see information related to a resource only when you're assigned the role of owner, contributor, or reader for the subscription or resource group that a resource belongs to.

NOTE

To learn more about roles and allowed actions in Security Center, see [Permissions in Azure Security Center](#).

Security Center uses the Microsoft Monitoring Agent. This is the same agent that the Log Analytics service uses. Data collected from this agent is stored in either an existing Log Analytics [workspace](#) associated with your Azure subscription or a new workspace, taking into account the geolocation of the VM.

Azure Monitor

Performance issues in your cloud app can affect your business. With multiple interconnected components and frequent releases, degradations can happen at any time. And if you're developing an app, your users usually discover issues that you didn't find in testing. You should know about these issues immediately, and you should have tools for diagnosing and fixing the problems.

[Azure Monitor](#) is basic tool for monitoring services running on Azure. It gives you infrastructure-level data about the throughput of a service and the surrounding environment. If you're managing your apps all in Azure and deciding whether to scale up or down resources, Azure Monitor is the place to start.

You can also use monitoring data to gain deep insights about your application. That knowledge can help you to improve application performance or maintainability, or automate actions that would otherwise require manual intervention.

Azure Monitor includes the following components.

Azure Activity Log

The [Azure Activity Log](#) provides insight into the operations that were performed on resources in your subscription. It was previously known as "Audit Log" or "Operational Log," because it reports control-plane events for your subscriptions.

Azure diagnostic logs

[Azure diagnostic logs](#) are emitted by a resource and provide rich, frequent data about the operation of that resource. The content of these logs varies by resource type.

Windows event system logs are one category of diagnostic logs for VMs. Blob, table, and queue logs are categories of diagnostic logs for storage accounts.

Diagnostic logs differ from the [Activity Log](#). The Activity log provides insight into the operations that were performed on resources in your subscription. Diagnostic logs provide insight into operations that your resource performed itself.

Metrics

Azure Monitor provides telemetry that gives you visibility into the performance and health of your workloads on Azure. The most important type of Azure telemetry data is the [metrics](#) (also called performance counters) emitted by most Azure resources. Azure Monitor provides several ways to configure and consume these metrics for monitoring and troubleshooting.

Azure Diagnostics

Azure Diagnostics enables the collection of diagnostic data on a deployed application. You can use the Diagnostics extension from various sources. Currently supported are [Azure cloud service roles](#), [Azure virtual machines](#) running Microsoft Windows, and [Azure Service Fabric](#).

Azure Network Watcher

Customers build an end-to-end network in Azure by orchestrating and composing individual network resources

such as virtual networks, Azure ExpressRoute, Azure Application Gateway, and load balancers. Monitoring is available on each of the network resources.

The end-to-end network can have complex configurations and interactions between resources. The result is complex scenarios that need scenario-based monitoring through [Azure Network Watcher](#).

Network Watcher simplifies monitoring and diagnosing of your Azure network. You can use the diagnostic and visualization tools in Network Watcher to:

- Take remote packet captures on an Azure virtual machine.
- Gain insights into your network traffic by using flow logs.
- Diagnose Azure VPN Gateway and connections.

Network Watcher currently has the following capabilities:

- [Topology](#): Provides a view of the various interconnections and associations between network resources in a resource group.
- [Variable packet capture](#): Captures packet data in and out of a virtual machine. Advanced filtering options and fine-tuned controls, such as the ability to set time and size limitations, provide versatility. The packet data can be stored in a blob store or on the local disk in .cap format.
- [IP flow verify](#): Checks if a packet is allowed or denied based on 5-tuple packet parameters for flow information (destination IP, source IP, destination port, source port, and protocol). If a security group denies the packet, the rule and group that denied the packet are returned.
- [Next hop](#): Determines the next hop for packets being routed in the Azure network fabric, so you can diagnose any misconfigured user-defined routes.
- [Security group view](#): Gets the effective and applied security rules that are applied on a VM.
- [NSG flow logs for network security groups](#): Enable you to capture logs related to traffic that is allowed or denied by the security rules in the group. The flow is defined by 5-tuple information: source IP, destination IP, source port, destination port, and protocol.
- [Virtual network gateway and connection troubleshooting](#): Provides the ability to troubleshoot virtual network gateways and connections.
- [Network subscription limits](#): Enables you to view network resource usage against limits.
- [Diagnostic logs](#): Provides a single pane to enable or disable diagnostic logs for network resources in a resource group.

For more information, see [Configure Network Watcher](#).

Cloud Service Provider Access Transparency

[Customer Lockbox for Microsoft Azure](#) is a service integrated into Azure portal that gives you explicit control in the rare instance when a Microsoft Support Engineer may need access to your data to resolve an issue. There are very few instances, such as a debugging remote access issue, where a Microsoft Support Engineer requires elevated permissions to resolve this issue. In such cases, Microsoft engineers use just-in-time access service that provides limited, time-bound authorization with access limited to the service.

While Microsoft has always obtained customer consent for access, Customer Lockbox now gives you the ability to review and approve or deny such requests from the Azure Portal. Microsoft support engineers will not be granted access until you approve the request.

Standardized and Compliant Deployments

[Azure Blueprints](#) enable cloud architects and central information technology groups to define a repeatable set of Azure resources that implement and adhere to an organization's standards, patterns, and requirements.

This makes it possible for DevOps teams to rapidly build and stand up new environments and trust that they're building them with infrastructure that maintains organizational compliance. Blueprints provide a declarative way to

orchestrate the deployment of various resource templates and other artifacts such as:

- Role Assignments
- Policy Assignments
- Azure Resource Manager templates
- Resource Groups

DevOps

Before [Developer Operations \(DevOps\)](#) application development, teams were in charge of gathering business requirements for a software program and writing code. Then a separate QA team tested the program in an isolated development environment. If requirements were met, the QA team released the code for operations to deploy. The deployment teams were further fragmented into groups like networking and database. Each time a software program was “thrown over the wall” to an independent team, it added bottlenecks.

DevOps enables teams to deliver more secure, higher-quality solutions faster and more cheaply. Customers expect a dynamic and reliable experience when consuming software and services. Teams must rapidly iterate on software updates and measure the impact of the updates. They must respond quickly with new development iterations to address issues or provide more value.

Cloud platforms such as Microsoft Azure have removed traditional bottlenecks and helped commoditize infrastructure. Software reigns in every business as the key differentiator and factor in business outcomes. No organization, developer, or IT worker can or should avoid the DevOps movement.

Mature DevOps practitioners adopt several of the following practices. These practices [involve people](#) to form strategies based on the business scenarios. Tooling can help automate the various practices.

- [Agile planning and project management](#) techniques are used to plan and isolate work into sprints, manage team capacity, and help teams quickly adapt to changing business needs.
- [Version control, usually with Git](#), enables teams located anywhere in the world to share source and integrate with software development tools to automate the release pipeline.
- [Continuous integration](#) drives the ongoing merging and testing of code, which leads to finding defects early. Other benefits include less time wasted on fighting merge issues and rapid feedback for development teams.
- [Continuous delivery](#) of software solutions to production and testing environments helps organizations quickly fix bugs and respond to ever-changing business requirements.
- [Monitoring](#) of running applications--including production environments for application health, as well as customer usage--helps organizations form a hypothesis and quickly validate or disprove strategies. Rich data is captured and stored in various logging formats.
- [Infrastructure as Code \(IaC\)](#) is a practice that enables the automation and validation of creation and teardown of networks and virtual machines to help with delivering secure, stable application hosting platforms.
- [Microservices](#) architecture is used to isolate business use cases into small reusable services. This architecture enables scalability and efficiency.

Next steps

To learn about the Security and Audit solution, see the following articles:

- [Security and compliance](#)
- [Azure Security Center](#)
- [Azure Monitor](#)

Azure Operational Security best practices

11/16/2018 • 9 minutes to read • [Edit Online](#)

Azure operational security refers to the services, controls, and features available to users for protecting their data, applications, and other assets in Azure. Azure operational security is built on a framework that incorporates the knowledge gained through capabilities that are unique to Microsoft, including the [Security Development Lifecycle \(SDL\)](#), the [Microsoft Security Response Center](#) program, and deep awareness of the cybersecurity threat landscape.

In this article, we discuss a collection of security best practices. These best practices are derived from our experience with Azure database security and the experiences of customers like yourself.

For each best practice, we explain:

- What the best practice is
- Why you want to enable that best practice
- What might be the result if you fail to enable the best practice
- How you can learn to enable the best practice

This Azure Operational Security Best Practices article is based on a consensus opinion, and Azure platform capabilities and feature sets, as they exist at the time this article was written. Opinions and technologies change over time and this article will be updated on a regular basis to reflect those changes.

Monitor storage services for unexpected changes in behavior

Diagnosing and troubleshooting issues in a distributed application hosted in a cloud environment can be more complex than it is in traditional environments. Applications can be deployed in a PaaS or IaaS infrastructure, on-premises, on a mobile device, or in some combination of these environments. Your application's network traffic might traverse public and private networks, and your application might use multiple storage technologies.

You should continuously monitor the storage services that your application uses for any unexpected changes in behavior (such as slower response times). Use logging to collect more detailed data and to analyze a problem in depth. The diagnostics information that you obtain from both monitoring and logging helps you to determine the root cause of the issue that your application encountered. Then you can troubleshoot the issue and determine the appropriate steps to remediate it.

[Azure Storage Analytics](#) performs logging and provides metrics data for an Azure storage account. We recommend that you use this data to trace requests, analyze usage trends, and diagnose issues with your storage account.

Prevent, detect, and respond to threats

[Azure Security Center](#) helps you prevent, detect, and respond to threats with increased visibility into (and control over) the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with various security solutions.

Security Center's Free tier offers limited security for your Azure resources only. The Standard tier extends these capabilities to on-premises and other clouds. Security Center Standard helps you find and fix security vulnerabilities, apply access and application controls to block malicious activity, detect threats using analytics and intelligence, and respond quickly when under attack. You can try Security Center Standard at no cost for the first 60 days. We recommend that you [onboard your Azure subscription to Security Center Standard](#).

Use Security Center to get a central view of the security state of all of your Azure resources. At a glance, verify that the appropriate security controls are in place and configured correctly, and quickly identify any resources that need attention.

Monitor end-to-end scenario-based network monitoring

Customers build an end-to-end network in Azure by combining network resources like a virtual network, ExpressRoute, Application Gateway, and load balancers. Monitoring is available on each of the network resources.

[Azure Network Watcher](#) is a regional service. Use its diagnostic and visualization tools to monitor and diagnose conditions at a network scenario level in, to, and from Azure.

The following are best practices for network monitoring and available tools.

Best practice: Automate remote network monitoring with packet capture.

Detail: Monitor and diagnose networking issues without logging in to your VMs by using Network Watcher.

Trigger [packet capture](#) by setting alerts and gain access to real-time performance information at the packet level.

When you see an issue, you can investigate in detail for better diagnoses.

Best practice: Gain insight into your network traffic by using flow logs.

Detail: Build a deeper understanding of your network traffic patterns by using [network security group flow logs](#).

Information in flow logs helps you gather data for compliance, auditing, and monitoring your network security profile.

Best practice: Diagnose VPN connectivity issues.

Detail: Use Network Watcher to [diagnose your most common VPN Gateway and connection issues](#). You can not only identify the issue but also use detailed logs to further investigate.

Secure deployment by using proven DevOps tools

Use the following DevOps best practices to ensure that your enterprise and teams are productive and efficient.

Best practice: Automate the build and deployment of services.

Detail: [Infrastructure as code](#) is a set of techniques and practices that help IT pros remove the burden of day-to-day build and management of modular infrastructure. It enables IT pros to build and maintain their modern server environment in a way that's like how software developers build and maintain application code.

You can use [Azure Resource Manager](#) to provision your applications by using a declarative template. In a single template, you can deploy multiple services along with their dependencies. You use the same template to repeatedly deploy your application in every stage of the application lifecycle.

Best practice: Automatically build and deploy to Azure web apps or cloud services.

Detail: You can use Azure Pipelines to [automatically build and deploy](#) to Azure web apps or cloud services. Azure Pipelines automatically deploys the binaries after doing a build to Azure after every code check-in. The package build process is equivalent to the Package command in Visual Studio, and the publishing steps are equivalent to the Publish command in Visual Studio.

Best practice: Use continuous deployment.

Detail: [Azure Pipelines](#) is a solution for automating multiple-stage deployment and managing the release process. Create managed continuous deployment pipelines to release quickly, easily, and often. With Azure Pipelines, you can automate your release process, and you can have predefined approval workflows. Deploy on-premises and to the cloud, extend, and customize as required.

Best practice: Check your app's performance before you launch it or deploy updates to production.

Detail: Run cloud-based [load tests](#) by using Azure Test Plans to:

- Find performance problems in your app.

- Improve deployment quality.
- Make sure that your app is always available.
- Make sure that your app can handle traffic for your next launch or marketing campaign.

Best practice: Monitor application performance.

Detail: [Azure Application Insights](#) is an extensible application performance management (APM) service for web developers on multiple platforms. Use Application Insights to monitor your live web application. It automatically detects performance anomalies. It includes analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability.

Mitigate and protect against DDoS

Distributed denial of service (DDoS) is a type of attack that tries to exhaust application resources. The goal is to affect the application's availability and its ability to handle legitimate requests. These attacks are becoming more sophisticated and larger in size and impact. They can be targeted at any endpoint that is publicly reachable through the internet.

Designing and building for DDoS resiliency requires planning and designing for a variety of failure modes.

Following are best practices for building DDoS-resilient services on Azure.

Best practice: Ensure that security is a priority throughout the entire lifecycle of an application, from design and implementation to deployment and operations. Applications can have bugs that allow a relatively low volume of requests to use a lot of resources, resulting in a service outage.

Detail: To help protect a service running on Microsoft Azure, you should have a good understanding of your application architecture and focus on the [five pillars of software quality](#). You should know typical traffic volumes, the connectivity model between the application and other applications, and the service endpoints that are exposed to the public internet.

Ensuring that an application is resilient enough to handle a denial of service that's targeted at the application itself is most important. Security and privacy are built into the Azure platform, beginning with the [Security Development Lifecycle \(SDL\)](#). The SDL addresses security at every development phase and ensures that Azure is continually updated to make it even more secure.

Best practice: Design your applications to [scale horizontally](#) to meet the demand of an amplified load, specifically in the event of a DDoS attack. If your application depends on a single instance of a service, it creates a single point of failure. Provisioning multiple instances makes your system more resilient and more scalable.

Detail: For [Azure App Service](#), select an [App Service plan](#) that offers multiple instances.

For Azure Cloud Services, configure each of your roles to use [multiple instances](#).

For [Azure Virtual Machines](#), ensure that your VM architecture includes more than one VM and that each VM is included in an [availability set](#). We recommend using virtual machine scale sets for autoscaling capabilities.

Best practice: Layering security defenses in an application reduces the chance of a successful attack. Implement secure designs for your applications by using the built-in capabilities of the Azure platform.

Detail: The risk of attack increases with the size (surface area) of the application. You can reduce the surface area by using whitelisting to close down the exposed IP address space and listening ports that are not needed on the load balancers ([Azure Load Balancer](#) and [Azure Application Gateway](#)).

[Network security groups](#) are another way to reduce the attack surface. You can use [service tags](#) and [application security groups](#) to minimize complexity for creating security rules and configuring network security, as a natural extension of an application's structure.

You should deploy Azure services in a [virtual network](#) whenever possible. This practice allows service resources to communicate through private IP addresses. Azure service traffic from a virtual network uses public IP addresses as source IP addresses by default.

Using [service endpoints](#) switches service traffic to use virtual network private addresses as the source IP addresses when they're accessing the Azure service from a virtual network.

We often see customers' on-premises resources getting attacked along with their resources in Azure. If you're connecting an on-premises environment to Azure, minimize exposure of on-premises resources to the public internet.

Azure has two DDoS [service offerings](#) that provide protection from network attacks:

- Basic protection is integrated into Azure by default at no additional cost. The scale and capacity of the globally deployed Azure network provides defense against common network-layer attacks through always-on traffic monitoring and real-time mitigation. Basic requires no user configuration or application changes and helps protect all Azure services, including PaaS services like Azure DNS.
- Standard protection provides advanced DDoS mitigation capabilities against network attacks. It's automatically tuned to protect your specific Azure resources. Protection is simple to enable during the creation of virtual networks. It can also be done after creation and requires no application or resource changes.

Next steps

See [Azure security best practices and patterns](#) for more security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.

The following resources are available to provide more general information about Azure security and related Microsoft services:

- [Azure Security Team Blog](#) - for up to date information on the latest in Azure Security
- [Microsoft Security Response Center](#) - where Microsoft security vulnerabilities, including issues with Azure, can be reported or via email to secure@microsoft.com

Azure operational security checklist

10/18/2018 • 4 minutes to read • [Edit Online](#)

Deploying an application on Azure is fast, easy, and cost-effective. Before deploying cloud application in production useful to have a checklist to assist in evaluating your application against a list of essential and recommended operational security actions for you to consider.

Introduction

Azure provides a suite of infrastructure services that you can use to deploy your applications. Azure Operational Security refers to the services, controls, and features available to users for protecting their data, applications, and other assets in Microsoft Azure.

- To get the maximum benefit out of the cloud platform, we recommend that you leverage Azure services and follow the checklist.
- Organizations that invest time and resources assessing the operational readiness of their applications before launch have a much higher rate of satisfaction than those who don't. When performing this work, checklists can be an invaluable mechanism to ensure that applications are evaluated consistently and holistically.
- The level of operational assessment will varies depending on the organization's cloud maturity level and the application's development phase, availability needs, and data sensitivity requirements.

Checklist

This checklist is intended to help enterprises think through various operational security considerations as they deploy sophisticated enterprise applications on Azure. It can also be used to help you build a secure cloud migration and operation strategy for your organization.

CHECKLIST CATEGORY	DESCRIPTION
Security Roles & Access Controls	<ul style="list-style-type: none">• Use Role based access control (RBAC) to provide user-specific that used to assign permissions to users, groups, and applications at a certain scope.

CHECKLIST CATEGORY	DESCRIPTION
Data Collection & Storage	<ul style="list-style-type: none"> • Use Management Plane Security to secure your Storage Account using Role-Based Access Control (RBAC). • Data Plane Security to Securing Access to your Data using Shared Access Signatures (SAS) and Stored Access Policies. • Use Transport-Level Encryption – Using HTTPS and the encryption used by SMB (Server message block protocols) 3.0 for Azure File Shares. • Use Client-side encryption to secure data that you send to storage accounts when you require sole control of encryption keys. • Use Storage Service Encryption (SSE) to automatically encrypt data in Azure Storage, and Azure Disk Encryption to encrypt virtual machine disk files for the OS and data disks. • Use Azure Storage Analytics to monitor authorization type; like with Blob Storage, you can see if users have used a Shared Access Signature or the storage account keys. • Use Cross-Origin Resource Sharing (CORS) to access storage resources from different domains.
Security Policies & Recommendations	<ul style="list-style-type: none"> • Use Azure Security Center to deploy endpoint solutions. • Add a web application firewall (WAF) to secure web applications. • Use a next generation firewall (NGFW) from a Microsoft partner to increase your security protections. • Apply security contact details for your Azure subscription; this the Microsoft Security Response Centre (MSRC) contacts you if it discovers that your customer data has been accessed by an unlawful or unauthorized party.
Identity & Access Management	<ul style="list-style-type: none"> • Synchronize your on-premises directory with your cloud directory using Azure AD. • Use Single Sign-On to enable users to access their SaaS applications based on their organizational account in Azure AD. • Use the Password Reset Registration Activity report to monitor the users that are registering. • Enable multi-factor authentication (MFA) for users. • Developers to use secure identity capabilities for apps like Microsoft Security Development Lifecycle (SDL). • Actively monitor for suspicious activities by using Azure AD Premium anomaly reports and Azure AD identity protection capability.

CHECKLIST CATEGORY	DESCRIPTION
Ongoing Security Monitoring	<ul style="list-style-type: none"> • Use Malware Assessment Solution Log Analytics to report on the status of antimalware protection in your infrastructure. • Use Update assessment to determine the overall exposure to potential security problems, and whether or how critical these updates are for your environment. • The Identity and Access provide you an overview of user <ul style="list-style-type: none"> • user identity state, • number of failed attempts to log on, • the user's account that were used during those attempts, accounts that were locked out • accounts with changed or reset password • Currently number of accounts that are logged in.
Azure Security Center detection capabilities	<ul style="list-style-type: none"> • Use detection capabilities, to identify active threats targeting your Microsoft Azure resources. • Use integrated threat intelligence that looks for known bad actors by leveraging global threat intelligence from Microsoft products and services, the Microsoft Digital Crimes Unit (DCU), the Microsoft Security Response Center (MSRC), and external feeds. • Use Behavioral analytics that applies known patterns to discover malicious behavior. • Use Anomaly detection that uses statistical profiling to build a historical baseline.
Developer Operations (DevOps)	<ul style="list-style-type: none"> • Infrastructure as Code (IaC) is a practice, which enables the automation and validation of creation and teardown of networks and virtual machines to help with delivering secure, stable application hosting platforms. • Continuous Integration and Deployment drive the ongoing merging and testing of code, which leads to finding defects early. • Release Management Manage automated deployments through each stage of your pipeline. • App Performance Monitoring of running applications including production environments for application health as well as customer usage help organizations form a hypothesis and quickly validate or disprove strategies. • Using Load Testing & Auto-Scale we can find performance problems in our app to improve deployment quality and to make sure our app is always up or available to cater to the business needs.

Conclusion

Many organizations have successfully deployed and operated their cloud applications on Azure. The checklists provided highlight several checklists that is essential and help you to increase the likelihood of successful deployments and frustration-free operations. We highly recommend these operational and strategic considerations for your existing and new application deployments on Azure.

Next steps

To learn more about Security, see the following articles:

- [Design and operational security.](#)
- [Azure Security Center planning and operations.](#)

Azure Security and Compliance Blueprint - IaaS Web Application for Australia Protected

10/15/2018 • 19 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of an infrastructure as a service (IaaS) environment suitable for the collection, storage, and retrieval of AU-PROTECTED government data that is compliant with the objectives of the Australian Government Information Security Manual (ISM) produced by the Australian Signals Directorate (ASD). This blueprint showcases a common reference architecture and helps demonstrate the proper handling of sensitive government data in a secure, compliant, multi-tier environment.

This reference architecture, implementation guide, and threat model provide a foundation for customers to undertake their own planning and system accreditation processes, helping customers deploy workloads to Azure in an ASD-compliant manner. Customers may choose to implement an Azure VPN Gateway or ExpressRoute to use federated services and to integrate on-premises resources with Azure resources. Customers must consider the security implications of using on-premises resources. Additional configuration is required to meet all the requirements, as they may vary based on the specifics of each customer's implementation.

Achieving ASD-compliance requires that an Information Security Registered Assessor audits the system. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution deploys a reference architecture for an IaaS web application with a SQL Server backend. The architecture includes a web tier, data tier, Active Directory infrastructure, Application Gateway, and Load Balancer. Virtual machines deployed to the web and data tiers are configured in an availability set, and SQL Server instances are configured in an Always On availability group for high availability. Virtual machines are domain-joined, and Active Directory group policies are used to enforce security and compliance configurations at the operating system level. A management bastion host provides a secure connection for administrators to access deployed resources.

The architecture can deliver a secure hybrid environment that extends an on-premises network to Azure, allowing web-based workloads to be accessed securely by corporate users of an organization's private local-area network or from the internet. For on-premises solutions, the customer is both accountable and responsible for all aspects of security, operations and compliance.

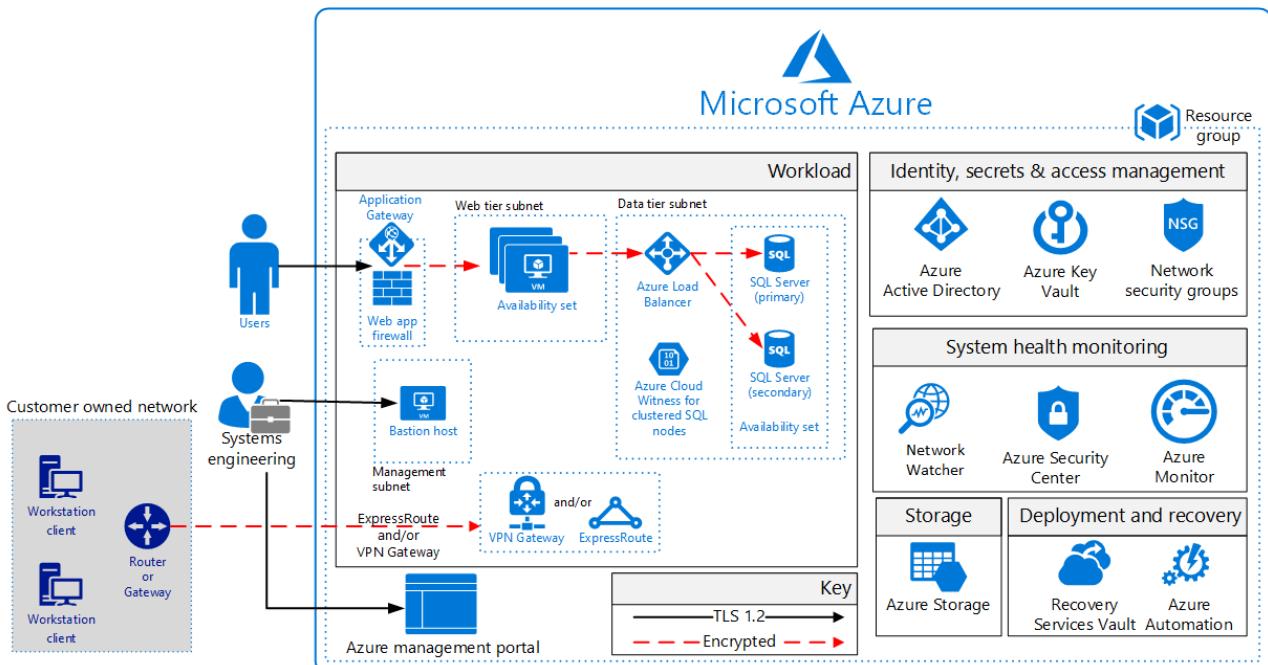
The Azure resources included in this solution can connect to an on-premises network or datacentre colocation facility (e.g. CDC in Canberra) through IPSec VPN using the Azure VPN Gateway or through ExpressRoute.. The decision to utilize a VPN should be done with the classification of the transmitted data and the network path in mind. Customers running large-scale, mission critical workloads with big data requirements should consider a hybrid network architecture using ExpressRoute for private network connectivity to Azure services. Refer to the guidance and recommendations section for further details on connection mechanisms to Azure.

Federation with Azure Active Directory should be used to enable users to authenticate using on-premises credentials and access all resources in the cloud by using an on-premises Active Directory Federation Services infrastructure. Active Directory Federation Services can provide simplified, secured identity federation and web single sign-on capabilities for this hybrid environment. Refer to the guidance and recommendations section for further details Azure Active Directory setup.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to

maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected region for resiliency. Azure regions are deployed in resilient region pairs, and geographic redundant storage ensures that data will be replicated to the second region with three copies as well. This prevents an adverse event at the customer's primary data location resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources and keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard. Azure Application Gateway is configured as a firewall in prevention mode and requires TLS traffic with a minimum version of 1.2.



This solution uses the following Azure services. Further details are in the [deployment architecture](#) section.

- Availability Sets
 - (1) SQL cluster nodes
 - (1) Web/IIS
- Azure Active Directory
- Azure Application Gateway
 - (1) Web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP 3.0
 - Listener port: 443
- Azure Cloud Witness
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure Log Analytics
- Azure Storage
- Azure Virtual Machines
 - (1) management/bastion (Windows Server 2016 Datacenter)

- (2) SQL Server cluster node (SQL Server 2017 on Windows Server 2016)
- (2) Web/IIS (Windows Server 2016 Datacenter)
- Azure Virtual Network
 - (4) Network security groups
 - Azure Network Watcher
- Recovery Services Vault

This Blueprint contains Azure Services that have not been certified for use at the Protected classification by the Australian Cyber Security Centre (ACSC). All services included in this reference architecture have been certified by ACSC at the Dissemination Limiting Markers (DLM) level. Microsoft recommends that customers review the published security and audit reports related to these Azure Services and use their risk management framework to determine whether the Azure Service is suitable for their internal accreditation and use at the Protected classification.

Deployment architecture

The following section details the deployment and implementation elements.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: This solution deploys resources in an architecture with a separate web subnet, database subnet, Active Directory subnet, and management subnet inside of a virtual network. Subnets are logically separated by network security group rules applied to the individual subnets to restrict traffic between subnets to only that necessary for system and management functionality.

See the configuration for [network security groups](#) deployed with this solution. Organizations can configure network security groups by editing the file above using [this documentation](#) as a guide.

Each of the subnets has a dedicated network security group:

- 1 network security group for Application Gateway (LBNSG)
- 1 network security group for bastion host (MGTNSG)
- 1 network security group for SQL Servers and Cloud Witness (SQLNSG)
- 1 network security group for web tier (WEBNSG)

Data in transit

Azure encrypts all communications to and from Azure datacentres by default.

For Protected data in transit from customer owned networks, the Architecture uses the Internet or ExpressRoute with a VPN Gateway configured with IPSEC.

Additionally, all transactions to Azure through the Azure management portal occur via HTTPS utilising TLS 1.2.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by the Australian Government ISM.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

SQL Server: The SQL Server instance uses the following database security measures:

- [SQL Server auditing](#) tracks database events and writes them to audit logs.
- [Transparent data encryption](#) performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps with reducing access such that sensitive data does not exit the database via unauthorized access. **Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.**

Identity management

Customers may utilize on-premises Active Directory Federated Services to federate with [Azure Active Directory](#), which is Microsoft's multi-tenant cloud-based directory and identity management service. [Azure Active Directory Connect](#) integrates on-premises directories with Azure Active Directory. All users in this solution require Azure Active Directory accounts. With federation sign-in, users can sign in to Azure Active Directory and authenticate to Azure resources using on-premises credentials.

Furthermore, the following Azure Active Directory capabilities help manage access to data in the Azure environment:

- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Azure Multi-Factor Authentication: To protect identities, multi-factor authentication should be implemented. [Azure Multi-Factor Authentication](#) is an easy to use, scalable, and reliable solution that provides a second method

of authentication to protect users. Azure Multi-Factor Authentication uses the power of the cloud and integrates with on-premises Active Directory and custom applications. This protection is extended to high-volume, mission-critical scenarios.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security module protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.
- The solution is integrated with Azure Key Vault to manage IaaS virtual machine disk-encryption keys and secrets.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Furthermore, this reference architecture utilizes the [vulnerability assessment](#) in Azure Security Center. Once configured, a partner agent (e.g. Qualys) reports vulnerability data to the partner's management platform. In turn, the partner's management platform provides vulnerability and health monitoring data back to Azure Security Center, allowing customers to quickly identify vulnerable virtual machines.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities using an Azure Application Gateway with a web application firewall configured, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)

- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Business continuity

High availability: The solution deploys all virtual machines in an [Availability Set](#). Availability sets ensure that the virtual machines are distributed across multiple isolated hardware clusters to improve availability. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS virtual machine without restoring the entire virtual machine, enabling faster restore times.

Cloud Witness: [Cloud Witness](#) is a type of Failover Cluster quorum witness in Windows Server 2016 that leverages Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations, but it uses the standard publicly available Azure Blob Storage. This eliminates the extra maintenance overhead of virtual machines hosted in a public cloud.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type, which allows all data to be analysed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- **Active Directory Assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **SQL Assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- **Activity Log Analytics:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

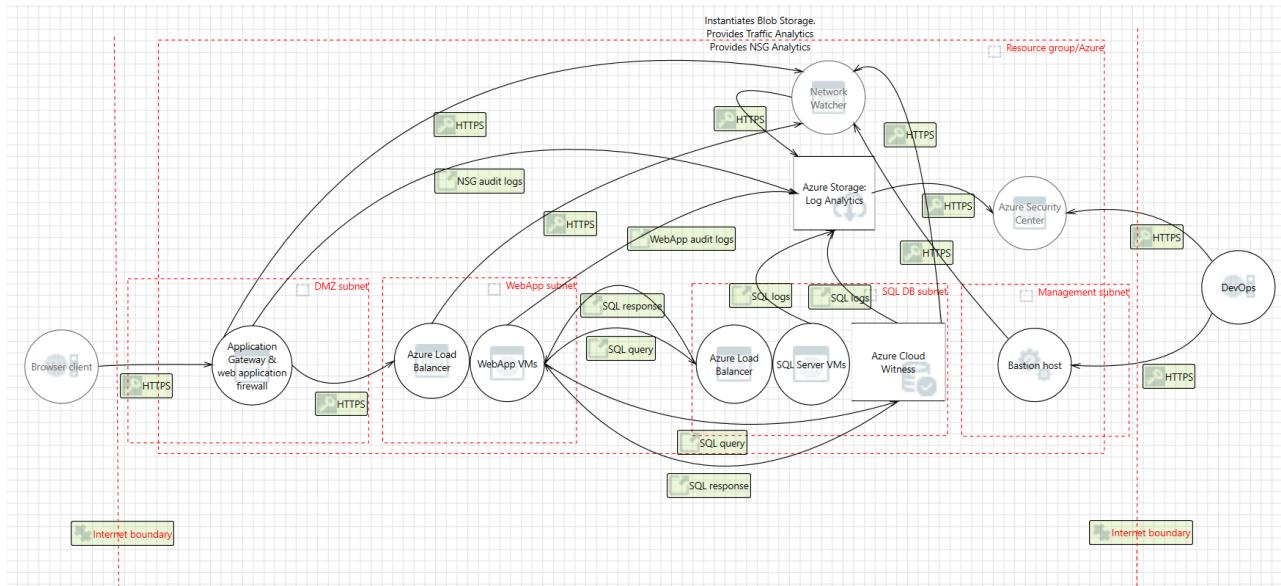
Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Server. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: Azure Monitor helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Azure Network Watcher: Azure Network Watcher provides tools to monitor, diagnose, view metrics, and enable or disable logs for resources in an Azure virtual network. Commonwealth entities should implement Network Watcher flow logs for NSGs and Virtual Machines. These logs should be stored in a dedicated storage account that only security logs are stored in and access to the storage account should be secured with Role Based Access Controls.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

This compliance documentation is produced by Microsoft based on platforms and services available from Microsoft. Due to the wide variety of customer deployments, this documentation provides a generalized approach for a solution only hosted in the Azure environment. Customers may identify and use alternative products and services based on their own operating environments and business outcomes. Customers choosing to use on-premises resources must address the security and operations for those on-premises resources. The documented solution can be customized by customers to address their specific on-premises and security requirements.

The [Azure Security and Compliance Blueprint – AU-PROTECTED Customer Responsibility Matrix](#) lists all security controls required by AU-PROTECTED. This matrix details whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – AU-PROTECTED IaaS Web Application Implementation Matrix](#) provides information on which AU-PROTECTED controls are addressed by the IaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

For classified information a secure IPSec VPN tunnel needs to be configured to securely establish a connection to the resources deployed as a part of this IaaS web application reference architecture. By appropriately setting up an IPSec VPN, customers can add a layer of protection for data in transit.

By implementing a secure IPSec VPN tunnel with Azure, a virtual private connection between an on-premises

network and an Azure virtual network can be created. This connection can take place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers a private connection option. Azure ExpressRoute is a dedicated link between Azure and an on-premises location or an Exchange hosting provider and is considered a private network. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds and lower latencies than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

To help protect classified data, whether using the Internet or Azure ExpressRoute, customers must configure their IPSec VPN by applying the following settings:

- The customer VPN initiator must be configured for a SA lifetime of no greater than 14400 seconds.
- The customer VPN initiator must disable IKEv1 aggressive mode.
- The customer VPN initiator must configure Perfect Forward Secrecy.
- The customer VPN Initiator must configure the use of HMAC-SHA256 or greater.

Configuration options for VPN devices and IPSec/ IKE parameters is [available](#) for review.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#).

Furthermore, [Azure Active Directory Connect](#) allows customers to configure federation with on-premises [Active Directory Federation Services](#) and Azure Active Directory. With federation sign-in, customers can enable users to sign in to Azure Active Directory-based services with their on-premises passwords and without having to enter their passwords again while on the corporate network. By using the federation option with Active Directory Federation Services, you can deploy a new installation of Active Directory Federation Services, or you can specify an existing installation in a Windows Server 2012 R2 farm.

To prevent classified data from synchronizing to Azure Active Directory, customers can restrict the attributes that are replicated to Azure Active Directory by applying the following settings in Azure Active Directory Connect:

- [Enable filtering](#)
- [Disable password hash synchronization](#)
- [Disable password writeback](#)
- [Disable device writeback](#)
- Leave the default settings for [prevent accidental deletes](#) and [automatic upgrade](#)

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.

- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - PaaS Web Application for Australia PROTECTED

10/17/2018 • 20 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of a platform as a service (PaaS) environment suitable for the collection, storage, and retrieval of AU-PROTECTED government data that is compliant with the objectives of the Australian Government Information Security Manual (ISM) produced by the Australian Signals Directorate (ASD). This blueprint showcases a common reference architecture and helps demonstrate the proper handling of sensitive government data in a secure, compliant, multi-tier environment.

This reference architecture, implementation guide, and threat model provide a foundation for customers to undertake their own planning and system accreditation processes, helping customers deploy workloads to Azure in an ASD-compliant manner. Customers may choose to implement an Azure VPN Gateway or ExpressRoute to use federated services and to integrate on-premises resources with Azure resources. Customers must consider the security implications of using on-premises resources. Additional configuration is required to meet all the requirements, as they may vary based on the specifics of each customer's implementation.

Achieving ASD-compliance requires that an Information Security Registered Assessor audits the system. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture for a PaaS web application with an Azure SQL Database backend. The web application is hosted in an isolated Azure App Service Environment, which is a private, dedicated environment in an Azure datacentre. The environment load balances traffic for the web application across virtual machines managed by Azure. All web application connections require TLS with a minimum version of 1.2. This architecture also includes network security groups, an Application Gateway, Azure DNS, and Load Balancer.

The architecture can deliver a secure hybrid environment that extends an on-premises network to Azure, allowing web-based workloads to be accessed securely by corporate users of an organization's private local-area network or from the internet. For on-premises solutions, the customer is both accountable and responsible for all aspects of security, operations, and compliance.

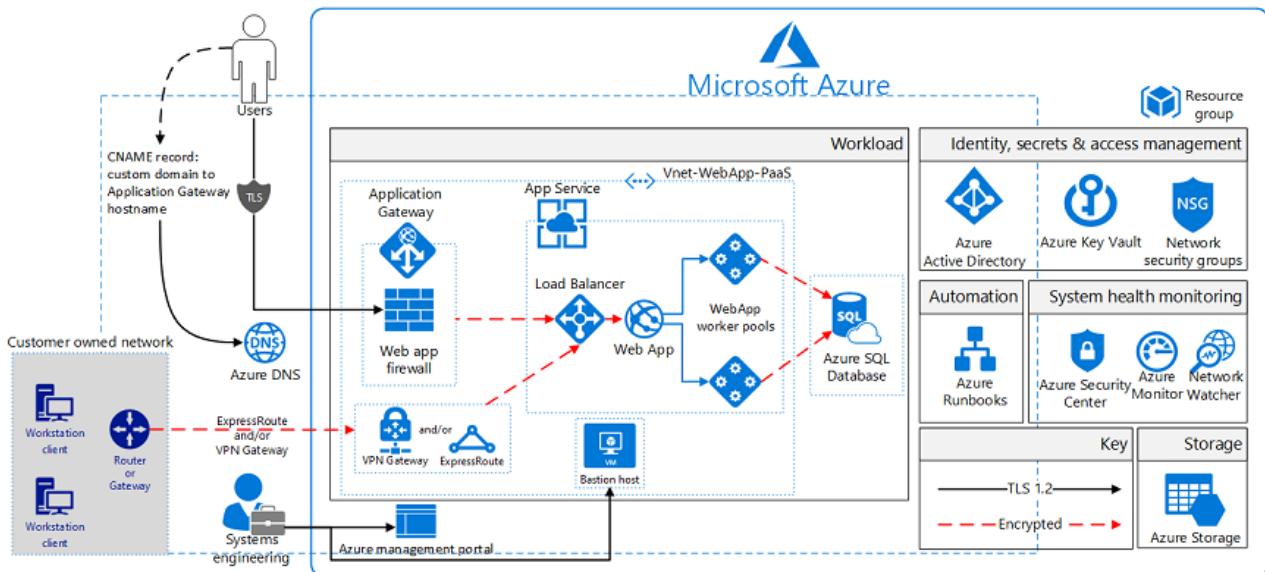
The Azure resources included in this solution can connect to an on-premises network or datacentre colocation facility (e.g. CDC in Canberra) through an IPSec VPN using a VPN Gateway and through ExpressRoute. The decision to utilise a VPN should be done with the classification of the transmitted data and the network path in mind. Customers running large-scale, mission critical workloads with big data requirements should consider a hybrid network architecture using ExpressRoute for private network connectivity to Azure services. Refer to the [guidance and recommendations](#) section for further details on connection mechanisms to Azure.

Federation with Azure Active Directory should be used to enable users to authenticate using on-premises credentials and access all resources in the cloud by using an on-premises Active Directory Federation Services infrastructure. Active Directory Federation Services can provide simplified, secured identity federation and web single sign-on capabilities for this hybrid environment. Refer to the [guidance and recommendations](#) section for further details Azure Active Directory setup.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected region for

resiliency. Azure regions are deployed in resilient region pairs, and geographic redundant storage ensures that data will be replicated to the second region with three copies as well. This prevents an adverse event at the customer's primary data location resulting in a loss of data.

For enhanced security, all Azure resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources and keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard. Azure Application Gateway is configured as a firewall in prevention mode and disallows traffic that is not TLS v1.2 or above. The solution utilizes Azure Application Service Environment v2 to isolate the web tier in a non-multi-tenant environment.



This solution uses the following Azure services. Further details are in the [deployment architecture](#) section.

- Application Gateway
 - Web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP
 - Listener port: 443
- Application Insights
- Azure Active Directory
- Azure Application Service Environment v2
- Azure Automation
- Azure DNS
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Log Analytics

- Azure Virtual Network
 - (1) /16 Network
 - (4) /24 Networks
 - Network security groups
- Network security groups
- Recovery Services Vault
- Azure Web App

This Blueprint contains Azure Services that have not been certified for use at the Protected classification by the Australian Cyber Security Centre (ACSC). Microsoft recommends that customers review the published security and audit reports related to these Azure Services and use their risk management framework to determine whether the Azure Service is suitable for their internal accreditation and use at the Protected classification.

Deployment architecture

The following section details the deployment and implementation elements.

Azure Resource Manager: [Azure Resource Manager](#) enables customers to work with the resources in the solution as a group. Customers can deploy, update, or delete all the resources for the solution in a single, coordinated operation. Customers use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help customers manage their resources after deployment.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use

App Service Environment v2: The [Azure App Service Environment](#) is an App Service feature that provides a fully isolated and dedicated environment for securely running App Service applications at a high scale.

App Service Environments are isolated to only run a single customer's applications and are always deployed into a virtual network. Customers have fine-grained control over both inbound and outbound application network traffic, and applications can establish high-speed secure connections over virtual networks to on-premises corporate resources.

Use of App Service Environments for this architecture allow for the following controls/configurations:

- App Service Environments must be configured to use the isolated service plan
 - Configure different App Service Environments for applications at different classifications
- Host inside a secured Azure virtual network and network security rules
- App Service Environments configured with a self-signed internal load balancer certificate for HTTPS communication. As a best practice, Microsoft recommends the use of a trusted certificate authority for enhanced security.
- [Internal load balancing mode](#) (mode 3)
- Disable [TLS v1.0 and v1.1](#)
- Change [TLS cipher](#)

- Control [inbound traffic N/W ports](#)
- [Web application firewall – restrict data](#)
- Allow [Azure SQL Database traffic](#)

Azure Web App: [Azure Web Apps](#) enables customers to build and host web applications in the programming language of their choice without managing infrastructure. It offers auto-scaling and high availability, supports both Windows and Linux, and enables automated deployments from GitHub, Azure DevOps Services, or any Git repo.

Virtual Network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual virtual machine level. The following network security groups exist:

- 1 network security group for Application Gateway
- 1 network security group for App Service Environment
- 1 network security group for Azure SQL Database
- 1 network security group for bastion host

Each of the network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Azure Log Analytics is connected to the [network security group's diagnostics](#)

Subnets: Each subnet is associated with its corresponding network security group.

Azure DNS: The Domain Name System, or DNS, is responsible for translating (or resolving) a website or service name to its IP address. [Azure DNS](#) is a hosting service for DNS domains that provides name resolution using Azure infrastructure. By hosting domains in Azure, users can manage DNS records using the same credentials, APIs, tools, and billing as other Azure services. Azure DNS also supports private DNS domains.

Azure Load Balancer: [Azure Load Balancer](#) allows customers to scale their applications and create high availability for services. Load Balancer supports inbound as well as outbound scenarios, and provides low latency, high throughput, and scales up to millions of flows for all TCP and UDP applications.

Data in transit

Azure encrypts all communications to and from Azure datacentres by default.

For Protected data in transit from customer owned networks, the Architecture uses Azure the Internet or ExpressRoute with a VPN Gateway configured with IPSEC.

Additionally, all transactions to Azure through the Azure management portal occur via HTTPS utilising TLS v1.2.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by Australian Government ISM.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns. SQL Threat Detection integrates alerts with [Azure Security Center](#), which includes details of suspicious activity and recommended action on how to investigate and mitigate the threat.
- [Always Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps with reducing access such that sensitive data does not exit the database via unauthorized access. Customers will need to adjust dynamic data masking settings to adhere to their database schema.

Identity management

Customers may utilize on-premises Active Directory Federated Services to federate with [Azure Active Directory](#), which is Microsoft's multi-tenant cloud-based directory and identity management service. [Azure Active Directory Connect](#) integrates on-premises directories with Azure Active Directory. All users in this solution require Azure Active Directory accounts, including users accessing the Azure SQL Database. With federation sign-in, users can sign in to Azure Active Directory and authenticate to Azure resources using on-premises credentials.

Furthermore, the following Azure Active Directory capabilities help manage access to data in the Azure environment:

- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see [how to protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Azure Multi-Factor Authentication: To protect identities, multi-factor authentication should be implemented. [Azure Multi-Factor Authentication](#) is an easy to use, scalable, and reliable solution that provides a second method of authentication to protect users. Azure Multi-Factor Authentication uses the power of the cloud and integrates with on-premises Active Directory and custom applications. This protection is extended to high-volume, mission-critical scenarios.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security module protected 2048-bit RSA key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Application Gateway: The architecture reduces the risk of security vulnerabilities using an Application Gateway with a web application firewall, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type, which allows all data to be analysed

together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

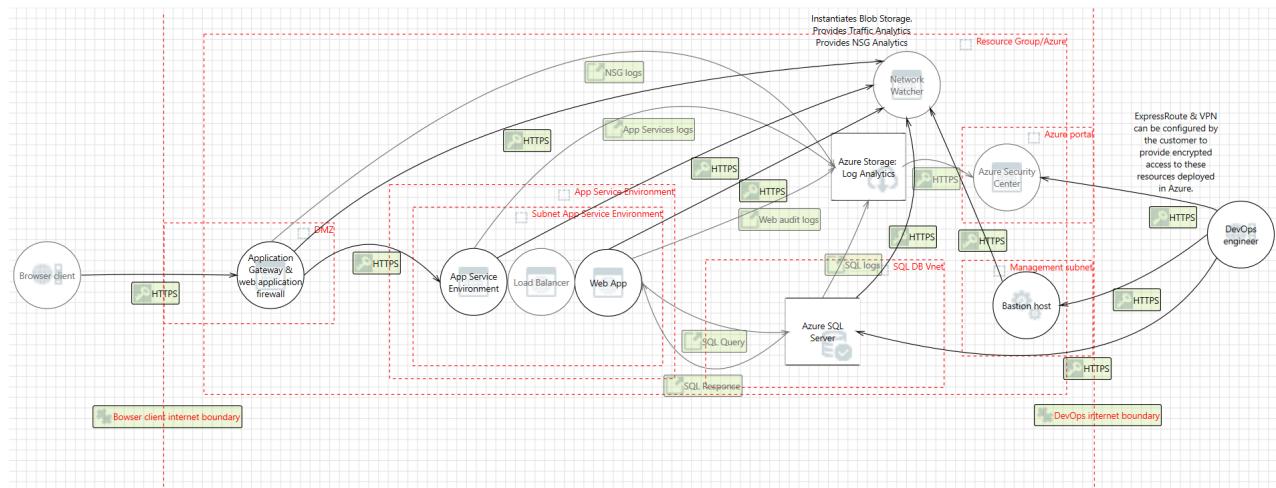
Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Azure Network Watcher: [Azure Network Watcher]9<https://docs.microsoft.com/azure/network-watcher/network-watcher-monitoring-overview>) provides tools to monitor, diagnose, view metrics, and enable or disable logs for resources in an Azure virtual network. Commonwealth entities should implement Network Watcher flow logs for NSGs and Virtual Machines. These logs should be stored in a dedicated storage account that only security logs are stored in and access to the storage account should be secured with Role Based Access Controls.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

This compliance documentation is produced by Microsoft based on platforms and services available from Microsoft. Due to the wide variety of customer deployments, this documentation provides a generalized approach for a solution only hosted in the Azure environment. Customers may identify and use alternative products and

services based on their own operating environments and business outcomes. Customers choosing to use on-premises resources must address the security and operations for those on-premises resources. The documented solution can be customized by customers to address their specific on-premises and security requirements.

The [Azure Security and Compliance Blueprint - AU-PROTECTED Customer Responsibility Matrix](#) lists all security controls required by AU-Prot. This matrix details whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - AU-PROTECTED PaaS Web Application Implementation Matrix](#) provides information on which AU-PROTECTED controls are addressed by the PaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

For classified information a secure IPSec VPN tunnel needs to be configured to securely establish a connection to the resources deployed as a part of this IaaS web application reference architecture. By appropriately setting up an IPSec VPN, customers can add a layer of protection for data in transit.

By implementing a secure IPSec VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection can take place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers a private connection option. Azure ExpressRoute is a dedicated link between Azure and an on-premises location or an Exchange hosting provider and is considered a private network. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds and lower latencies than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

To help protect classified data, whether using the Internet or Azure ExpressRoute, customers must configure their IPSec VPN by applying the following settings:

- The customer VPN initiator must be configured for a SA lifetime of no greater than 14400 seconds.
- The customer VPN initiator must disable IKEv1 aggressive mode.
- The customer VPN initiator must configure Perfect Forward Secrecy.
- The customer VPN Initiator must configure the use of HMAC-SHA256 or greater.

Configuration options for VPN devices and IPSec/ IKE parameters is [available](#) for review.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#).

Furthermore, [Azure Active Directory Connect](#) allows customers to configure federation with on-premises [Active Directory Federation Services](#) and Azure Active Directory. With federation sign-in, customers can enable users to sign in to Azure Active Directory-based services with their on-premises passwords and without having to enter their passwords again while on the corporate network. By using the federation option with Active Directory Federation Services, you can deploy a new installation of Active Directory Federation Services, or you can specify an existing installation in a Windows Server 2012 R2 farm.

To prevent classified data from synchronizing to Azure Active Directory, customers can restrict the attributes that are replicated to Azure Active Directory by applying the following settings in Azure Active Directory Connect:

- [Enable filtering](#)
- [Disable password hash synchronization](#)
- [Disable password writeback](#)
- [Disable device writeback](#)
- Leave the default settings for [prevent accidental deletes](#) and [automatic upgrade](#)

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Analytics for FedRAMP

10/18/2018 • 14 minutes to read • [Edit Online](#)

Overview

The [Federal Risk and Authorization Management Program \(FedRAMP\)](#) is a United States government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services. This Azure Security and Compliance Blueprint provides guidance for how to deliver a Microsoft Azure analytics architecture that helps implement a subset of FedRAMP High controls. This solution provides guidance on the deployment and configuration of Azure resources for a common reference architecture, demonstrating ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own analytics solutions in Azure.

This reference architecture, associated control implementation guides, and threat models are intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment. Deploying an application into this environment without modification is insufficient to completely meet the requirements of the FedRAMP High baseline. Please note the following:

- The architecture provides a baseline to help customers deploy workloads to Azure in a FedRAMP-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides an analytics platform upon which customers can build their own analytics tools. The reference architecture outlines a generic use case where customers input data either through bulk data imports by the SQL/Data Administrator or through operational data updates via an Operational User. Both work streams incorporate [Azure Functions](#) for importing data into the SQL Database. Azure Functions must be configured by the customer through the Azure Portal to handle the import tasks unique to each customer's own analytics requirements.

Microsoft Azure offers a variety of reporting and analytics services for the customer; however, this solution incorporates Azure Analysis Services in conjunction with Azure SQL Database to rapidly browse through data and deliver faster results through smarter modeling of customer data. Azure Analytics Services is a form of machine learning intended to increase query speeds by discovering new relationships between datasets. Once the data has been trained through several statistical functions, up to 7 additional query pools (8 total including the customer server) can be synchronized with the same tabular models to spread query workload and reduce response times.

For enhanced analytics and reporting, SQL Databases can be configured with columnstore indexes. Both Azure Analytics Services and SQL Databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

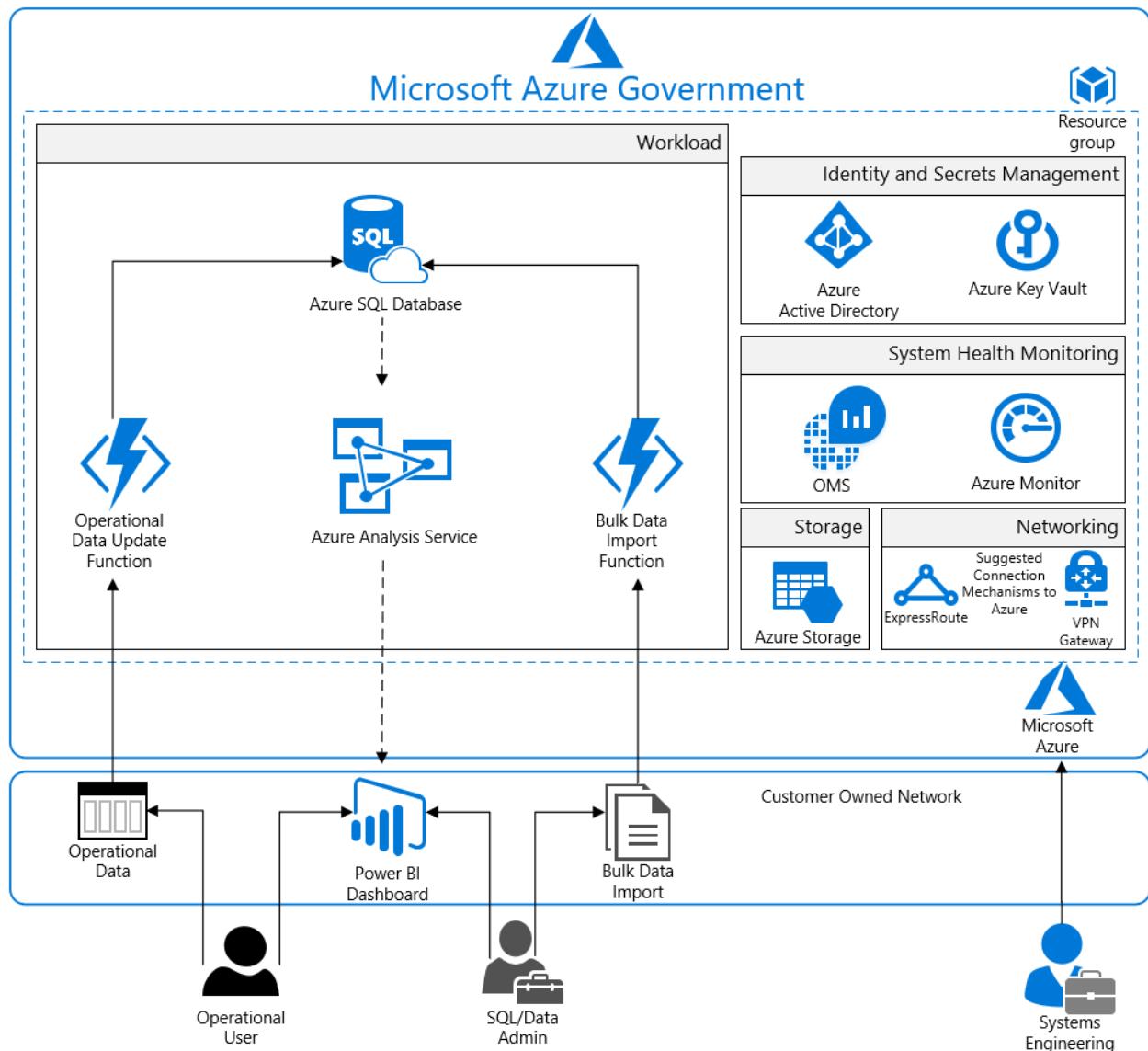
Once data is uploaded to the Azure SQL Database and trained by Azure Analysis Services, it is digested by both the Operational User and SQL/Data Admin with Power BI. Power BI displays data intuitively and pulls together information across multiple datasets to draw greater insight. Its high degree of adaptability and easy integration with Azure SQL Database ensures that customers can configure it to handle a wide array of scenarios as required

by their business needs.

The entire solution is built upon an Azure Storage which account customers configure from the Azure Portal. Azure Storage encrypts all data with Storage Service Encryption to maintain confidentiality of data at rest. Geographic Redundant Storage (GRS) ensures that an adverse event at the customer's primary data center will not result in a loss of data as a second copy will be stored in a separate location hundreds of miles away.

For enhanced security, this architecture manages resources with Azure Active Directory and Azure Key Vault. System health is monitored through Log Analytics and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

Azure SQL Database is commonly managed through SQL Server Management Studio (SSMS), which runs from a local machine configured to access the Azure SQL Database via a secure VPN or ExpressRoute connection. **Azure recommends configuring a VPN or Azure ExpressRoute connection for management and data import into the reference architecture resource group.**



Roles

The analytics blueprint outlines a scenario with three general user types: the Operational User, the SQL/Data Admin, and the Systems Engineer. Azure Role-based Access Control (RBAC) enables the implementation of precise access management through built-in custom roles. Resources are available for configuring [Role-based Access Control](#) and outlining and implementing [pre-defined roles](#).

Systems engineer

The Systems Engineer owns the customer subscription for Azure and configures the deployment of the solution

through the Azure Portal.

SQL/data administrator

The SQL/Data Administrator establishes the bulk data import function and the operational data update function for uploading to the Azure SQL database. The SQL/Data Administrator is not responsible for any operational data updates in the database but will be able to view the data through Power BI.

Operational user

The Operational User updates the data regularly and owns the day-to-day data generation. The Operational User will also interpret results through Power BI.

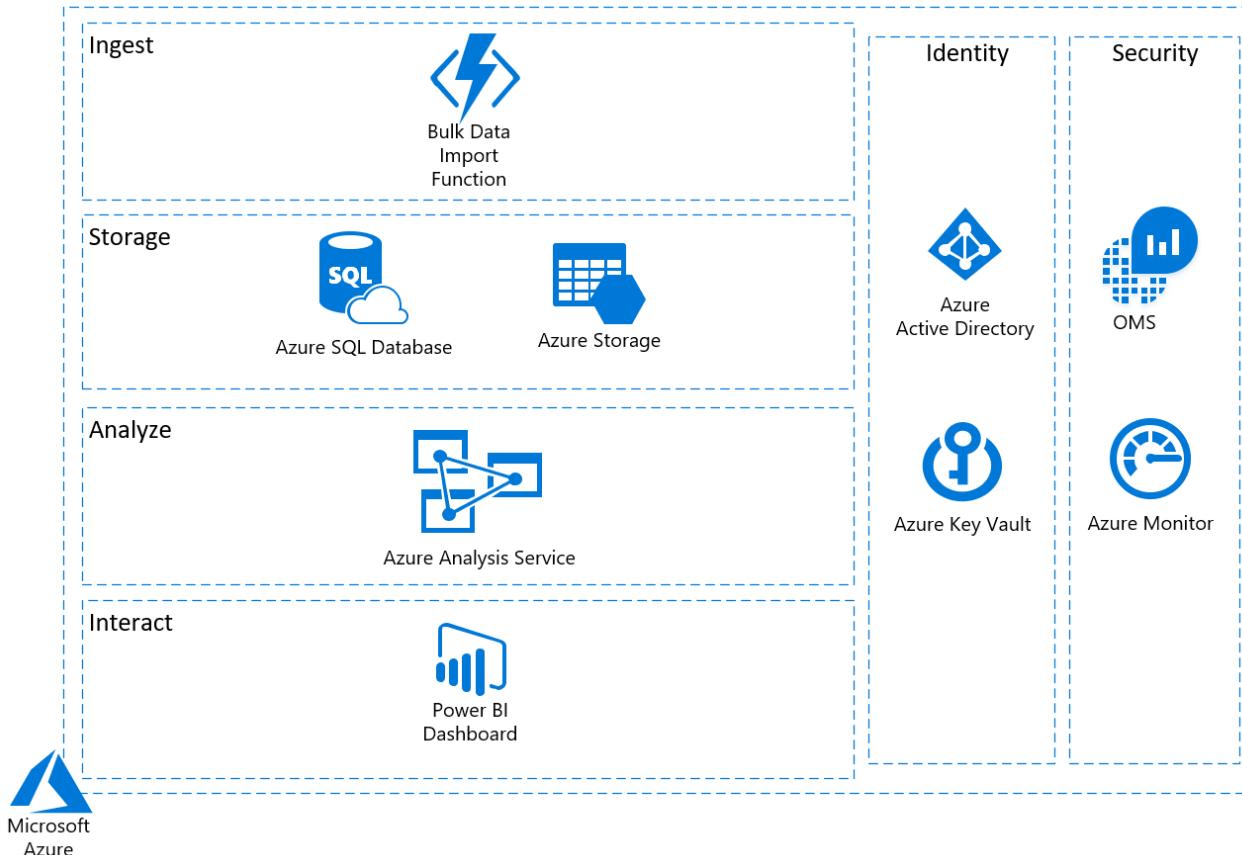
Azure services

This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- Azure Functions
- Azure SQL Database
- Azure Analysis Service
- Azure Active Directory
- Azure Key Vault
- Azure Log Analytics
- Azure Monitor
- Azure Storage
- ExpressRoute/VPN Gateway
- Power BI Dashboard

Deployment architecture

The following section details the development and implementation elements.



Azure Functions: Azure Functions are solutions for running small pieces of code in the cloud via most programming languages. Functions in this solution integrate with Azure Storage to automatically pull customer data into the cloud, facilitating integration with other Azure services. Functions are easily scalable and only incur a cost when they are running.

Azure Analysis Service: Azure Analysis Service provides enterprise data modeling and integration with Azure data platform services. Azure Analysis Service speeds up browsing through massive amounts of data by combining data from multiple sources into a single data model.

Power BI: Power BI provides analytics and reporting functionality for customers trying to pull greater insight out of their data processing efforts.

Networking

Network security groups: NSGs are set up to manage traffic directed at deployed resources and services. Network Security Groups are set to a deny-by-default scheme and only permit traffic that is contained in the pre-configured Access Control List (ACL).

Each of the NSGs have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- Diagnostic logs and events are enabled and stored in a storage account
- Log Analytics is connected to the NSG's diagnostic logs.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Data replication Azure Government has two options for data replication:

- The default setting for data replication is **Geo-Redundant Storage (GRS)**, which asynchronously stores customer data in a separate data center outside of the primary region. This ensures recovery of data in a total loss event for the primary data center.
- **Locally Redundant Storage (LRS)** can alternatively be configured via the Azure Storage Account. LRS replicates data within a storage scale unit, which is hosted in the same region in which the customer creates their account. All data is replicated concurrently, ensuring that no backup data is lost in a primary storage scale unit failure.

Azure Storage To meet encrypted data at rest requirements, all services deployed in this reference architecture leverage [Azure Storage](#), which stores data with [Storage Service Encryption](#).

Azure Disk Encryption Azure Disk Encryption leverages the BitLocker feature of Windows to provide volume encryption for OS and data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database The Azure SQL Database instance uses the following database security measures:

- AD authentication and authorization enables identity management of database users and other Microsoft services in one central location.
- SQL database auditing tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of data and log files to protect information at rest. TDE provides assurance that stored data has not been subject to unauthorized access.
- Firewall rules prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- SQL Threat Detection enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.

- [Always Encrypted columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or app servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) can be done after the reference architecture deploys. Customers will need to adjust dynamic data masking settings to adhere to their database schema.

Logging and audit

[Azure Monitor](#) generates an all-up display of monitoring data including activity logs, metrics, and diagnostic data, enabling customers to create a complete picture of system health.

[Log Analytics](#) provides extensive logging of system and user activity, as well as system health. It collects and analyzes data generated by resources in Azure and on-premises environments.

- **Activity Logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription.
- **Diagnostic Logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs and Azure Blob storage, tables, and queue logs.
- **Firewall Logs:** The Application Gateway provides full diagnostic and access logs. Firewall logs are available for WAF-enabled Application Gateway resources.
- **Log Archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival with a defined retention period of 2 days. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

Additionally, the following monitoring solutions are included as a part of this architecture:

- [Azure Automation](#): The Azure Automation solution stores, runs, and manages runbooks.
- [Security and Audit](#): The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Azure Activity Logs](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Identity management

- Authentication to the application is performed using Azure AD. For more information, see [Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure AD to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in SQL Database](#).
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.
- [Azure Role-based Access Control \(RBAC\)](#) enables focused access management for Azure. Subscription access is limited to the subscription administrator.

To learn more about using the security features of Azure SQL Database, see the [Contoso Clinic Demo Application](#) sample.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services.

Guidance and recommendations

ExpressRoute and VPN

ExpressRoute or a secure VPN tunnel needs to be configured to securely establish a connection to the resources deployed as a part of this data analytics reference architecture. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. By appropriately setting up ExpressRoute or a VPN, customers can add a layer of protection for data in transit.

Azure Active Directory setup

Azure Active Directory is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with AAD in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing AAD by making the deployed Active Directory infrastructure a subdomain of an AAD forest.

Additional services

IaaS - VM considerations

This PaaS solution does not incorporate any Azure IaaS VMs. A customer could create an Azure VM to run many of these PaaS services. In this case, specific features and services for business continuity and Log Analytics could be leveraged:

Business continuity

- **High availability:** Server workloads are grouped in an [Availability Set](#) to help ensure high availability of virtual machines in Azure. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.
- **Recovery Services Vault:** The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS VM without restoring the entire VM, enabling faster restore times.

Monitoring solutions

- [AD Assessment:](#) The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [Antimalware Assessment:](#) The Antimalware solution reports on malware, threats, and protection status.
- [Update Management:](#) The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- [Agent Health:](#) The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Change Tracking:](#) The Change Tracking solution allows customers to easily identify changes in the environment.

Security

- **Malware protection:** Microsoft Antimalware for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.
- **Patch management:** Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Azure Commercial

Although this data analytics architecture is not intended for deployment to the [Azure Commercial](#) environment, similar objectives can be achieved through the services described in this reference architecture, as well as additional services available only in the Azure Commercial environment. Please note that Azure Commercial maintains a FedRAMP JAB P-ATO at the Moderate Impact Level, allowing government agencies and partners to deploy moderately sensitive information to the cloud leveraging the Azure Commercial environment.

Azure Commercial offers a wide variety of analytics services for pulling insights out of large amounts of data:

- [Azure Machine Learning Studio](#), a component of the [Cortana Intelligence Suite](#), develops a predictive analysis model from one or more data sources. Statistical functions are used over several iterations to generate an effective model that applications such as Power BI can then consume.
- [Azure Data Lake Store](#) enables the capture of data of any size, type, and ingestion speed in a single place for operational and exploratory analytics. Azure Data Lake Store is compatible with most open source components in the Hadoop ecosystem and integrates nicely with other Azure services.

Threat model

The data flow diagram (DFD) for this reference architecture is available for [download](#) or can be found below:

Compliance documentation

The [Azure Security and Compliance Blueprint – FedRAMP High Customer Responsibility Matrix](#) lists all security controls required by the FedRAMP High baseline. Similarly, the [Azure Security and Compliance Blueprint – FedRAMP Moderate Customer Responsibility Matrix](#) lists all security controls required by the FedRAMP Moderate baseline. Both documents detail whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - FedRAMP High Control Implementation Matrix](#) and the [Azure Security and Compliance Blueprint - FedRAMP Moderate Control Implementation Matrix](#) provide information on which controls are covered by the analytics architecture for each FedRAMP baseline, including detailed descriptions of how the implementation meets the requirements of each covered control.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Data Warehouse for FedRAMP Automation

10/18/2018 • 13 minutes to read • [Edit Online](#)

Overview

The [Federal Risk and Authorization Management Program \(FedRAMP\)](#) is a United States government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services. This Azure Security and Compliance Blueprint provides guidance for how to deliver a Microsoft Azure data warehouse architecture that helps implement a subset of FedRAMP High controls. This solution provides guidance on the deployment and configuration of Azure resources for a common reference architecture, demonstrating ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own data warehouse solutions in Azure.

This reference architecture, associated control implementation guides, and threat models are intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment. Deploying an application into this environment without modification is insufficient to completely meet the requirements of the FedRAMP High baseline. Please note the following:

- The architecture provides a baseline to help customers deploy workloads to Azure in a FedRAMP-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a data warehouse reference architecture which implements a high-performance and secure cloud data warehouse. There are two separate data tiers in this architecture: one where data is imported, stored, and staged within a clustered SQL environment, and another for the Azure SQL Data Warehouse where the data is loaded using an ETL tool (e.g. [PolyBase](#) T-SQL queries) for processing. Once data is stored in Azure SQL Data Warehouse, analytics can run at a massive scale.

Microsoft Azure offers a variety of reporting and analytics services for the customer. This solution includes SQL Server Reporting Services (SSRS) for quick creation of reports from the Azure SQL Data Warehouse. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

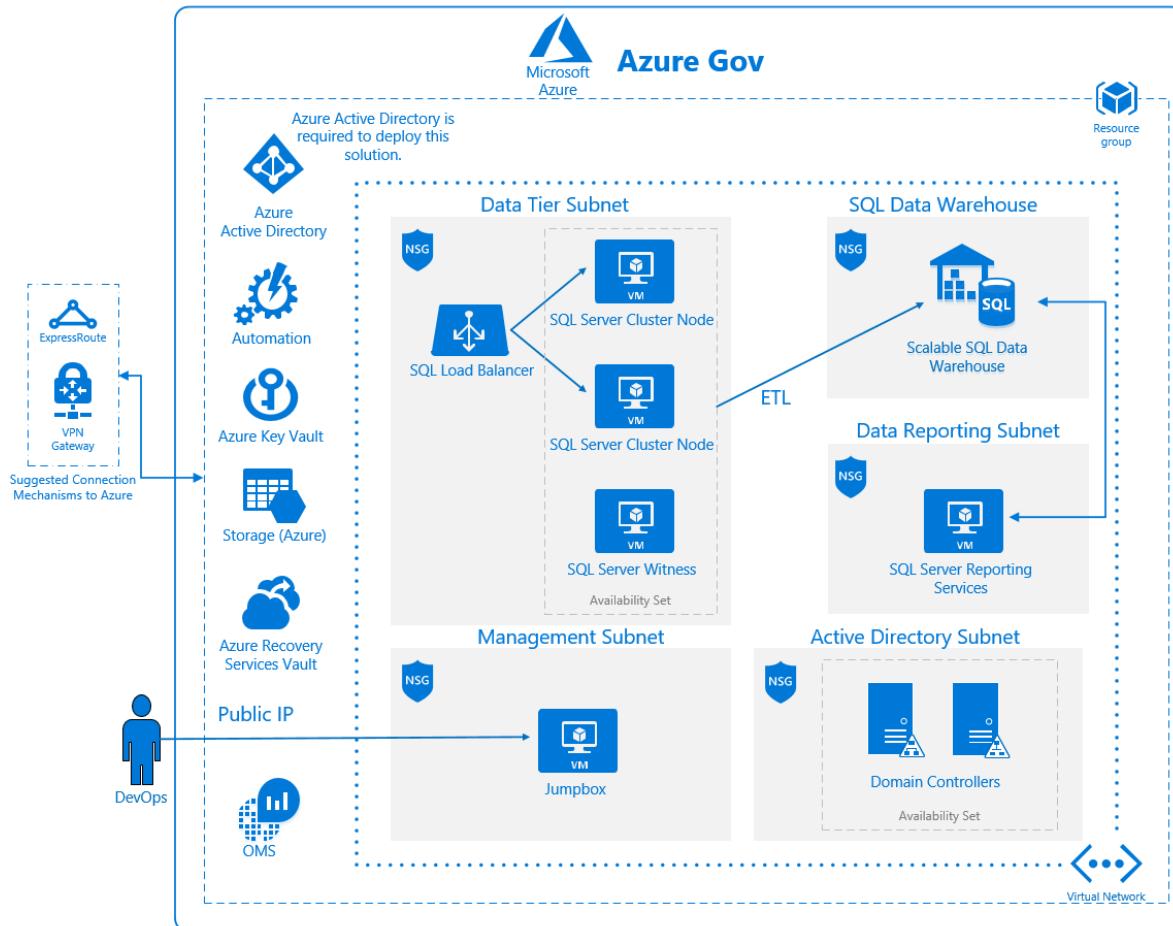
Data in the Azure SQL Data Warehouse is stored in relational tables with columnar storage, a format that significantly reduces the data storage costs while improving query performance. Depending on usage requirements, Azure SQL Data Warehouse compute resources can be scaled up or down or shut off completely if there are no active processes requiring compute resources.

A SQL load balancer manages SQL traffic, ensuring high performance. All virtual machines in this reference architecture deploy in an availability set with SQL Server instances configured in an AlwaysOn availability group for high-availability and disaster-recovery capabilities.

This data warehouse reference architecture also includes an Active Directory (AD) tier for management of resources within the architecture. The Active Directory subnet enables easy adoption under a larger AD forest structure, allowing for continuous operation of the environment even when access to the larger forest is

unavailable. All virtual machines are domain-joined to the Active Directory tier and use Active Directory group policies to enforce security and compliance configurations at the operating system level.

A virtual machine serves as a management bastion host, providing a secure connection for administrators to access deployed resources. The data loads into the staging area through this management bastion host. **Azure recommends configuring a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment architecture](#) section.

Azure Virtual Machines

- (1) Bastion Host
- (2) Active Directory domain controller
- (2) SQL Server Cluster Node
- (1) SQL Server Witness

Availability Sets

- (1) Active Directory domain controllers
- (1) SQL cluster nodes and witness

Virtual Network

- (4) Subnets
- (4) Network Security Groups

SQL Data Warehouse

SQL Server Reporting Services

Azure SQL Load Balancer

Azure Active Directory

Recovery Services Vault

Azure Key Vault

Log Analytics

Deployment architecture

The following section details the development and implementation elements.

SQL Data Warehouse: [SQL Data Warehouse](#) is an Enterprise Data Warehouse (EDW) that leverages Massively Parallel Processing (MPP) to quickly run complex queries across petabytes of data. Import big data into SQL Data Warehouse with simple PolyBase T-SQL queries and use the power of MPP to run high-performance analytics.

SQL Server Reporting Services: [SQL Server Reporting Services](#) enables quick creation of reports with tables, charts, maps, gauges, matrixes, and more for Azure SQL Data Warehouse.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the Network Security Group (NSG).

A virtual machine was created as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Log Analytics extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault (respects Azure Government, PCI DSS, HIPAA and other requirements)
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

Virtual network

This reference architecture defines a private virtual network with an address space of 10.0.0.0/16.

Network security groups: [NSGs](#) contain Access Control Lists (ACLs) that allow or deny traffic within a VNet. NSGs can be used to secure traffic at a subnet or individual VM level. The following NSGs exist:

- An NSG for the Data Tier (SQL Server Clusters, SQL Server Witness, and SQL Load Balancer)
- An NSG for the management bastion host
- An NSG for Active Directory
- An NSG for SQL Server Reporting Services

Each of the NSGs have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [NSG's diagnostics](#)

Subnets: Each subnet is associated with its corresponding NSG.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#).

Azure Disk Encryption [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for OS and data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database The Azure SQL Database instance uses the following database security measures:

- [AD Authentication and Authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of data and log files to protect information at rest. TDE provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or app servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) can be done after the reference architecture deploys. Customers will need to adjust dynamic data masking settings to adhere to their database schema.

Business continuity

High availability: Server workloads are grouped in an [Availability Set](#) to help ensure high availability of virtual machines in Azure. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS VM without restoring the entire VM, enabling faster restore times.

Logging and audit

[Log Analytics](#) provides extensive logging of system and user activity, as well as system health. The [Log Analytics](#) solution collects and analyzes data generated by resources in Azure and on-premises environments.

- **Activity Logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription.
- **Diagnostic Logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs and Azure Blob storage, tables, and queue logs.
- **Firewall Logs:** The Application Gateway provides full diagnostic and access logs. Firewall logs are available for WAF-enabled Application Gateway resources.
- **Log Archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival with a defined retention period of 2 days. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

Additionally, the following monitoring solutions are included as a part of this architecture:

- **AD Assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **Antimalware Assessment:** The Antimalware solution reports on malware, threats, and protection status.

- [Azure Automation](#): The Azure Automation solution stores, runs, and manages runbooks.
- [Security and Audit](#): The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Update Management](#): The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Azure Activity Logs](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.
- [Change Tracking](#): The Change Tracking solution allows customers to easily identify changes in the environment.

Identity management

The following technologies provide identity management capabilities in the Azure environment:

- [Active Directory \(AD\)](#) can be Microsoft's multi-tenant cloud-based directory and identity management service. All users for the solution were created in Azure Active Directory, including users accessing the SQL Database.
- Authentication to the application is performed using Azure AD. For more information, see [Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure AD to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in SQL Database](#).
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.
- [Azure Role-based Access Control \(RBAC\)](#) enables focused access management for Azure. Subscription access is limited to the subscription administrator.

To learn more about using the security features of Azure SQL Database, see the [Contoso Clinic Demo Application](#) sample.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Guidance and recommendations

ExpressRoute and VPN

[ExpressRoute](#) or a secure VPN tunnel needs to be configured to securely establish a connection to the resources deployed as a part of this data warehouse reference architecture. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. By appropriately setting up ExpressRoute or a VPN, customers can add a layer of

protection for data in transit.

Extract-Transform-Load (ETL) process

PolyBase can load data into Azure SQL Data Warehouse without the need for a separate ETL or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

Azure Active Directory is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with AAD in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing AAD by making the deployed Active Directory infrastructure a subdomain of an AAD forest.

Additional services

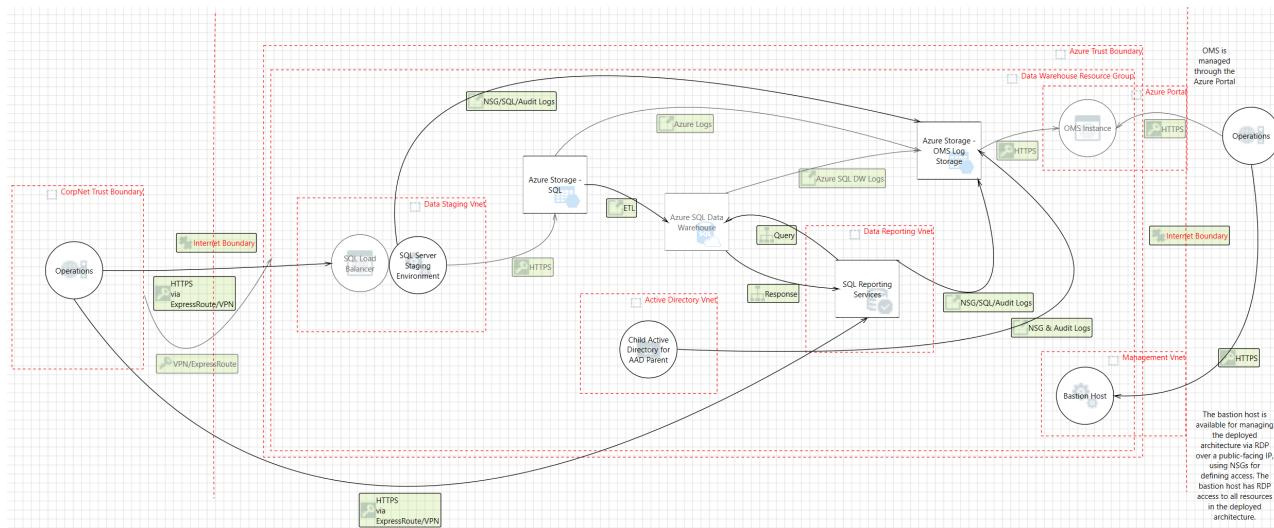
Although this data warehouse architecture is not intended for deployment to the [Azure Commercial](#) environment, similar objectives can be achieved through the services described in this reference architecture, as well as additional services available only in the Azure Commercial environment. Please note that Azure Commercial maintains a FedRAMP JAB P-ATO at the Moderate Impact Level, allowing government agencies and partners to deploy moderately sensitive information to the cloud leveraging the Azure Commercial environment.

Azure Commercial offers a wide variety of services that handle formatted and unformatted data storage and staging to be used in data warehousing, including:

- [Azure Data Factory](#) is a managed cloud service that is built for complex hybrid extract-transform-load (ETL), extract-load-transform (ELT), and data integration projects. Using Azure Data Factory, customers can create and schedule data-driven workflows called pipelines that ingest data from disparate data stores. Customers can then process and transform the data for output into data stores such as Azure SQL Data Warehouse.
- [Azure Data Lake Store](#) enables the capture of data of any size, type, and ingestion speed in a single place for operational and exploratory analytics. Azure Data Lake Store is compatible with most open source components in the Hadoop ecosystem and integrates nicely with other Azure services such as Azure SQL Data Warehouse.

Threat model

The data flow diagram (DFD) for this reference architecture is available for [download](#) or can be found below:



Compliance documentation

The [Azure Security and Compliance Blueprint – FedRAMP High Customer Responsibility Matrix](#) lists all security controls required by the FedRAMP High baseline. Similarly, the [Azure Security and Compliance Blueprint – FedRAMP Moderate Customer Responsibility Matrix](#) lists all security controls required by the FedRAMP Moderate

baseline. Both documents detail whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - FedRAMP High Control Implementation Matrix](#) and the [Azure Security and Compliance Blueprint - FedRAMP Moderate Control Implementation Matrix](#) provide information on which controls are covered by the data warehouse architecture for each FedRAMP baseline, including detailed descriptions of how the implementation meets the requirements of each covered control.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: IaaS Web Application for FedRAMP

10/18/2018 • 12 minutes to read • [Edit Online](#)

Overview

The [Federal Risk and Authorization Management Program \(FedRAMP\)](#) is a U.S. government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services. This Azure Security and Compliance Blueprint Automation provides guidance for the deployment of a FedRAMP-compliant infrastructure as a service (IaaS) environment suitable for a simple Internet-facing web application. This solution automates deployment and configuration of Azure resources for a common reference architecture, demonstrating ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own solutions on Azure. The solution implements a subset of controls from the FedRAMP High baseline, based on NIST SP 800-53. For more information about FedRAMP requirements and this solution, see the [compliance documentation](#).

NOTE

This solution deploys to Azure Government.

This Azure Security and Compliance Blueprint Automation automatically deploys an IaaS web application reference architecture with pre-configured security controls to help customers achieve compliance with FedRAMP requirements. The solution consists of Azure Resource Manager templates and PowerShell scripts that guide resource deployment and configuration.

This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment. Deploying an application into this environment without modification is not sufficient to completely meet the requirements of the FedRAMP High baseline. Please note the following:

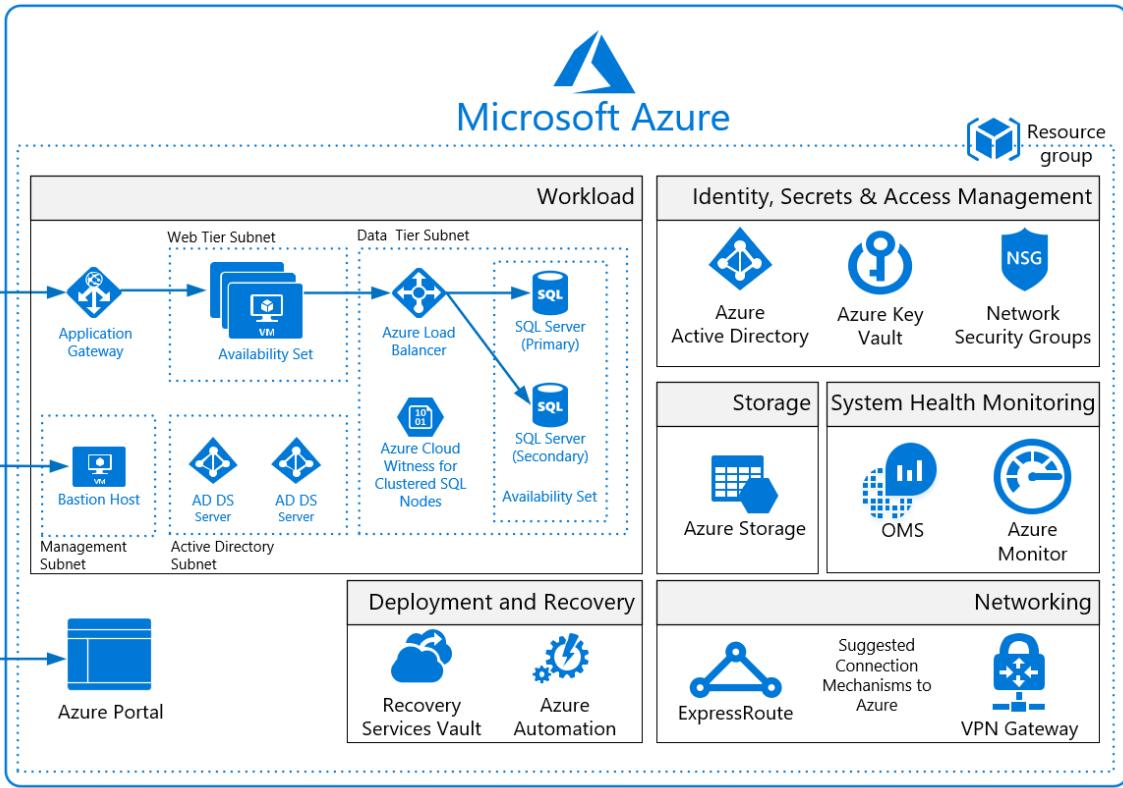
- This architecture provides a baseline to help customers use Azure in a FedRAMP-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessment of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

For a quick overview of how this solution works, watch this [video](#) explaining and demonstrating its deployment.

Click [here](#) for deployment instructions.

Architecture diagram and components

This solution deploys a reference architecture for an IaaS web application with a SQL Server backend. The architecture includes a web tier, data tier, Active Directory infrastructure, Application Gateway, and Load Balancer. Virtual machines deployed to the web and data tiers are configured in an Availability Set, and SQL Server instances are configured in an AlwaysOn availability group for high availability. Virtual machines are domain-joined, and Active Directory group policies are used to enforce security and compliance configurations at the operating system level. A bastion host provides a secure connection for administrators to access deployed resources. **Azure recommends configuring a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

- Azure Virtual Machines
 - (1) bastion host (Windows Server 2016 Datacenter)
 - (2) Active Directory domain controller (Windows Server 2016 Datacenter)
 - (2) SQL Server cluster node (SQL Server 2017 on Windows Server 2016)
 - (2) Web/IIS (Windows Server 2016 Datacenter)
- Availability Sets
 - (1) Active Directory domain controllers
 - (1) SQL cluster nodes
 - (1) Web/IIS
- Azure Virtual Network
 - (1) /16 virtual networks
 - (5) /24 subnets
 - DNS settings are set to both domain controllers
- Azure Load Balancer
- Azure Application Gateway
 - (1) WAF Application Gateway enabled
 - firewall mode: prevention
 - rule set: OWASP 3.0
 - listener: port 443
- Azure Storage
 - (7) Geo-redundant storage accounts
- Azure Cloud Witness
- Recovery Services vault
- Azure Key Vault
- Azure Active Directory (Azure AD)

- Azure Resource Manager
- Log Analytics
- Azure Monitor

Deployment architecture

The following section details the development and implementation elements.

Bastion host: The bastion host is the single point of entry that provides a secure connection for administrators to access deployed resources. The bastion host's NSG allows connections only on TCP port 3389 for RDP. Customers can further configure the bastion host to meet organization system hardening requirements.

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: This solution deploys resources in an architecture with a separate web subnet, database subnet, Active Directory subnet, and management subnet inside of a virtual network. Subnets are logically separated by network security group rules applied to the individual subnets to restrict traffic between subnets to only that necessary for system and management functionality.

Please see the configuration for [network security groups](#) deployed with this solution. Customers can configure network security groups by editing the file above using [this documentation](#) as a guide.

Each of the subnets has a dedicated network security group (NSG):

- 1 NSG for Application Gateway (LBNSG)
- 1 NSG for bastion host (MGTNSG)
- 1 NSG for Primary and Backup Domain Controllers (ADNSG)
- 1 NSG for SQL Servers (SQLNSG)
- 1 NSG for Web Tier (WEBNSG)

Subnets: Each subnet is associated with its corresponding NSG.

Data at rest

The architecture protects data at rest by using several encryption measures.

Azure Storage: To meet data-at-rest encryption requirements, all storage accounts use [Storage Service Encryption](#).

SQL Server: SQL Server is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of data and log files to protect information at rest. TDE provides assurance that stored data has not been subject to unauthorized access.

Customers may also configure the following SQL Server security measures:

- [AD authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or app servers with access to the keys can access plaintext data.

- [SQL Database dynamic data masking](#) can be done after the reference architecture deploys. Customers will need to adjust dynamic data masking settings to adhere to their database schema.

Azure Disk Encryption: Azure Disk Encryption is used to encrypted Windows IaaS virtual machine disks. [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for OS and data disks. The solution is integrated with Azure Key Vault to help control and manage the disk-encryption keys.

Identity management

The following technologies provide identity management capabilities in the Azure environment:

- [Azure Active Directory \(Azure AD\)](#) is Microsoft's multi-tenant cloud-based directory and identity management service.
- Authentication to a customer-deployed web application can be performed using Azure AD. For more information, see [Integrating applications with Azure Active Directory](#).
- [Azure Role-based Access Control \(RBAC\)](#) enables precisely focused access management for Azure. Subscription access is limited to the subscription administrator, and access to resources can be limited based on user role.
- A deployed IaaS Active Directory instance provides identity management at the OS-level for deployed IaaS virtual machines.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. Azure Key Vault helps manage IaaS virtual machine disk-encryption keys and secrets for this reference architecture.

Patch management: Windows virtual machines deployed by this Azure Security and Compliance Blueprint Automation are configured by default to receive automatic updates from Windows Update Service. This solution also deploys the Azure Automation solution through which Update Deployments can be created to deploy patches to Windows servers when needed.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Application Gateway: The architecture reduces the risk of security vulnerabilities using an Application Gateway with web application firewall (WAF), and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-End-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall \(WAF mode\)](#)
- [Prevention mode](#) with OWASP 3.0 ruleset

Business continuity

High availability: At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA. The solution deploys all web tier and data tier virtual machines in an [Availability Set](#). Availability sets ensure that the virtual machines are distributed across multiple isolated hardware clusters to improve availability. Furthermore, this solution deploys the SQL Server virtual machines in an Availability Set as an [AlwaysOn availability group](#). The Always On availability group feature provides for high-availability and disaster-recovery capabilities.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS VM without restoring the entire VM, enabling faster restore times.

Cloud Witness: [Cloud Witness](#) is a type of Failover Cluster quorum witness in Windows Server 2016 that

leverages Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations, but it uses the standard publicly available Azure Blob Storage. This eliminates the extra maintenance overhead of VMs hosted in a public cloud.

Logging and auditing

Log Analytics provides extensive logging of system and user activity, as well as system health. The [Log Analytics](#) solution collects and analyzes data generated by resources in Azure and on-premises environments.

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) are all logs emitted by every resource. These logs include Windows event system logs, Azure storage logs, Key Vault audit logs, and Application Gateway access and firewall logs.
- **Log archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

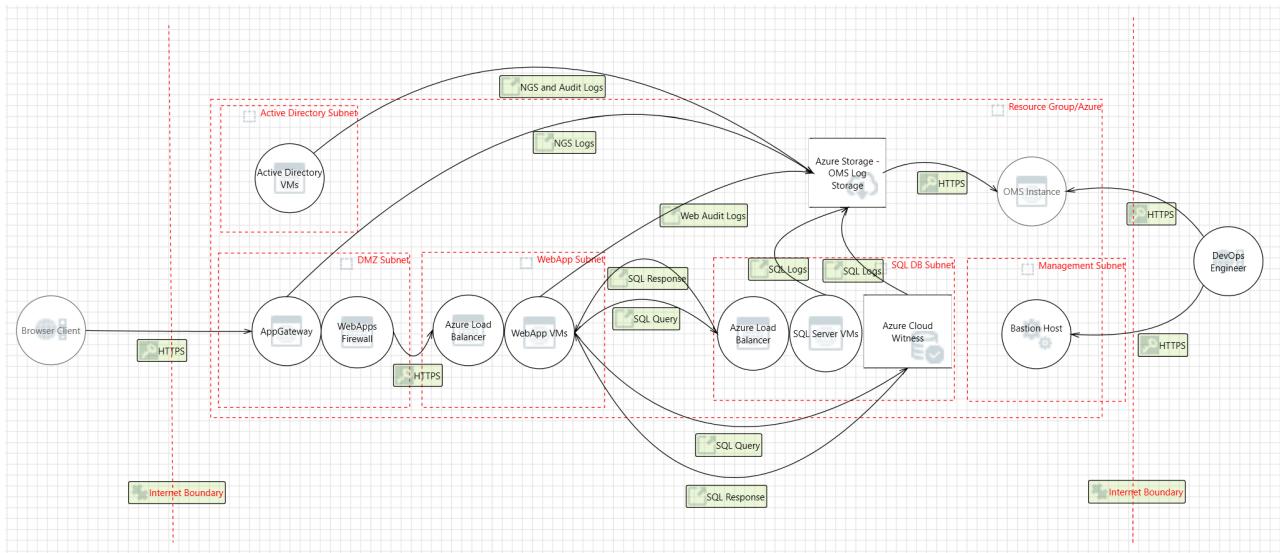
Additionally, the following monitoring solutions are installed as a part of this architecture. Note that it's the customer's responsibility to configure these solutions to align with FedRAMP security controls:

- [AD Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [Antimalware Assessment](#): The Antimalware solution reports on malware, threats, and protection status.
- [Azure Automation](#): The Azure Automation solution stores, runs, and manages runbooks.
- [Security and Audit](#): The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Update Management](#): The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Azure Activity Logs](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.
- [Change Tracking](#): The Change Tracking solution allows customers to easily identify changes in the environment.

Azure Monitor [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in customers' Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - FedRAMP High Customer Responsibility Matrix](#) lists all security controls required by the FedRAMP High baseline. The matrix denotes whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - FedRAMP IaaS Web Application High Control Implementation Matrix](#) lists all security controls required by the FedRAMP High baseline. The matrix provides information on which controls are covered by the IaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered control.

Deploy the solution

This Azure Security and Compliance Blueprint Automation is comprised of JSON configuration files and PowerShell scripts that are handled by Azure Resource Manager's API service to deploy resources within Azure. Detailed deployment instructions are available [here](#).

NOTE

This solution deploys to Azure Government.

Quickstart

1. Clone or download [this GitHub repository](#) to your local workstation.
2. Run the pre-deployment PowerShell script: `azure-blueprint/predeploy/Orchestration_InitialSetup.ps1`.
3. Click the button below, sign into the Azure portal, enter the required ARM template parameters, and click **Purchase**.

 Deploy to Azure Gov

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this IaaS Web Application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network

and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: PaaS Web Application for FedRAMP

10/18/2018 • 12 minutes to read • [Edit Online](#)

Overview

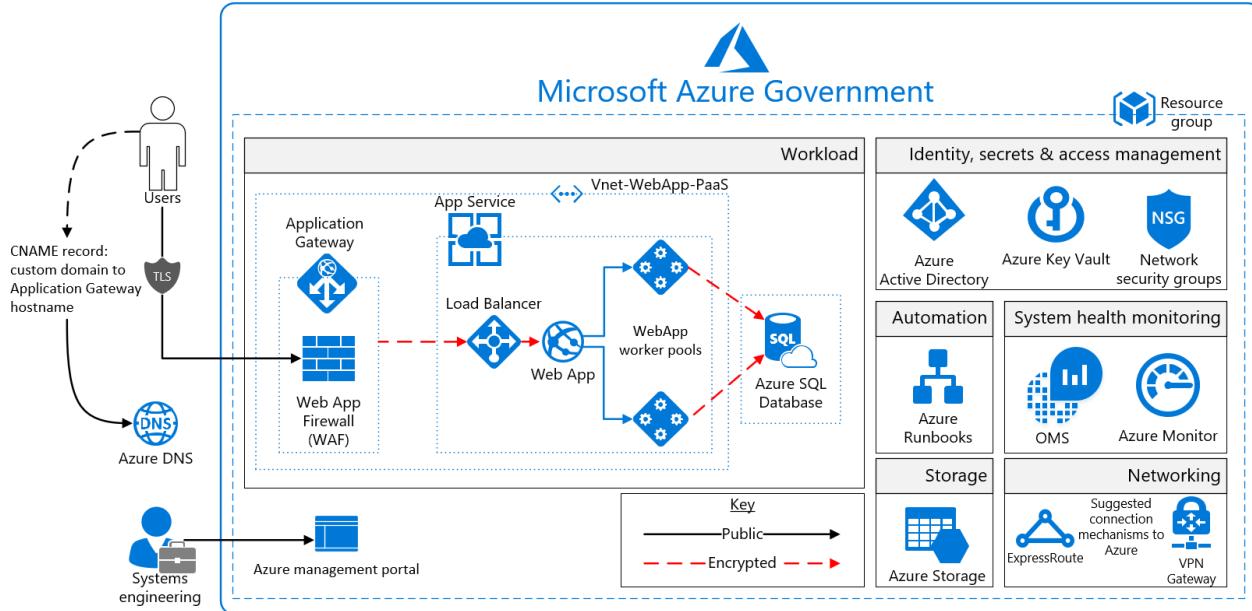
The [Federal Risk and Authorization Management Program \(FedRAMP\)](#) is a United States government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services. This Azure Security and Compliance Blueprint provides guidance for how to deliver a Microsoft Azure platform as a service (PaaS) architecture that helps implement a subset of FedRAMP High controls. This solution provides guidance on the deployment and configuration of Azure resources for a common reference architecture, demonstrating ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own solutions on Azure.

This reference architecture, associated control implementation guides, and threat models are intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment. Deploying an application into this environment without modification is insufficient to completely meet the requirements of the FedRAMP High baseline. Please note the following:

- The architecture provides a baseline to help customers deploy workloads to Azure in a FedRAMP-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture for a PaaS web application with an Azure SQL Database backend. The web application is hosted in an isolated Azure App Service Environment, which is a private, dedicated environment in an Azure datacenter. The environment load balances traffic for the web application across VMs managed by Azure. This architecture also includes network security groups, an Application Gateway, Azure DNS, and Load Balancer. Furthermore, Azure Monitor provides real-time analytics of system health. **Azure recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

- Azure Active Directory
- Azure Key Vault
- Azure SQL Database
- Application Gateway
 - (1) Web Application Firewall
 - Firewall mode: prevention
 - Rule set: OWASP 3.0
 - Listener: port 443
- Azure virtual network
- Network security groups
- Azure DNS
- Azure Storage
- Azure Monitor
- App Service Environment v2
- Azure Load Balancer
- Azure Web App
- Azure Resource Manager

Deployment architecture

The following section details the deployment and implementation elements.

Azure Resource Manager: [Azure Resource Manager](#) enables customers to work with the resources in the solution as a group. Customers can deploy, update, or delete all the resources for the solution in a single, coordinated operation. Customers use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help customers manage their resources after deployment.

App Service Environment v2: The [Azure App Service Environment \(ASE\)](#) is an App Service feature that provides a fully isolated and dedicated environment for securely running App Service applications at a high scale.

ASEs are isolated to only run a single customer's applications and are always deployed into a virtual network.

Customers have fine-grained control over both inbound and outbound application network traffic, and applications can establish high-speed secure connections over virtual networks to on-premises corporate resources.

Use of ASEs for this architecture are allowed for the following controls/configurations:

- Host inside a secured Azure Virtual Network and network security rules
- ASE configured with self-signed ILB certificate for HTTPS communication
- [Internal Load Balancing mode](#)
- Disable [TLS 1.0](#)
- Change [TLS Cipher](#)
- Control [inbound traffic N/W ports](#)
- [Web Application Firewall – Restrict Data](#)
- Allow [Azure SQL Database traffic](#)

The [Guidance and recommendations](#) section contains additional information about ASEs.

Azure Web App: [Azure Web Apps](#) enables customers to build and host web applications in the programming language of their choice without managing infrastructure. It offers auto-scaling and high availability, supports both Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repo.

Virtual Network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: [Network security groups \(NSGs\)](#) contain access control lists that allow or deny traffic within a virtual network. NSGs can be used to secure traffic at a subnet or individual VM level. The following NSGs exist:

- 1 NSG for Application Gateway
- 1 NSG for App Service Environment
- 1 NSG for Azure SQL Database

Each of the NSGs have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [NSG's diagnostics](#)

Subnets: Each subnet is associated with its corresponding NSG.

Azure DNS: The Domain Name System, or DNS, is responsible for translating (or resolving) a website or service name to its IP address. [Azure DNS](#) is a hosting service for DNS domains that provides name resolution using Azure infrastructure. By hosting domains in Azure, users can manage DNS records using the same credentials, APIs, tools, and billing as other Azure services. Azure DNS also supports private DNS domains.

Azure Load Balancer: [Azure Load Balancer](#) allows customers to scale their applications and create high availability for services. Load Balancer supports inbound as well as outbound scenarios, and provides low latency, high throughput, and scales up to millions of flows for all TCP and UDP applications.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#).

Azure Disk Encryption Azure Disk Encryption leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [AD authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database dynamic data masking](#) can be done after the reference architecture deploys. Customers will need to adjust dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide identity management capabilities in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in AAD, including users accessing the Azure SQL Database.
- Authentication to the application is performed using AAD. For more information, see [Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables precisely focused access management for Azure. Subscription access is limited to the subscription administrator, and access to resources can be limited based on user role.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use AAD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets Management The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect data and access to such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules (HSMs). The key type is an HSM

Protected 2048-bit RSA Key.

- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Application Gateway The architecture reduces the risk of security vulnerabilities using an Application Gateway with Web Application Firewall, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-End-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web Application Firewall](#)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Logging and auditing

Azure Monitor provides extensive logging of system and user activity, as well as system health. It collects and analyzes data generated by resources in Azure and on-premises environments.

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs.
- **Log archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

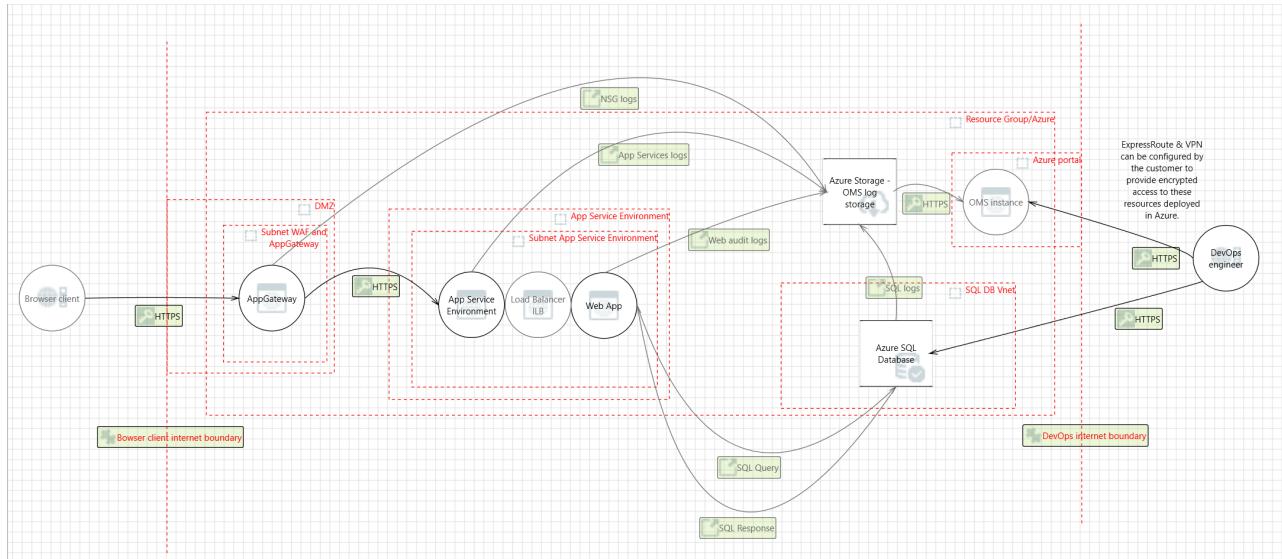
Additionally, the following monitoring solutions are included as a part of this architecture:

- [Active directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [Antimalware Assessment](#): The Antimalware solution reports on malware, threats, and protection status.
- [Azure Automation](#): The Azure Automation solution stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Application Insights and Azure SQL Database.
- [Security and Audit](#): The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Update Management](#): The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Azure Activity Logs](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.
- [Change Tracking](#): The Change Tracking solution allows customers to easily identify changes in the environment.

Azure Monitor Azure Monitor helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in customers' Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - FedRAMP High Customer Responsibility Matrix](#) lists all security controls required by the FedRAMP High baseline. The matrix denotes whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - FedRAMP PaaS WebApp High Control Implementation Matrix](#) lists all security controls required by the FedRAMP High baseline. The matrix provides information on which controls are covered by the PaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this PaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a VPN connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely “tunnel” information inside an encrypted link between the customer’s network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer’s telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are available.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Analytics for FFIEC Financial Services

12/6/2018 • 16 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of a data analytics architecture in Azure suitable for the collection, storage, and retrieval of financial data regulated by the Federal Financial Institution Examination Council (FFIEC).

This reference architecture, implementation guide, and threat model provide a foundation for customers to comply with FFIEC requirements. This solution provides a baseline to help customers deploy workloads to Azure in a FFIEC compliant manner; however, this solution should not be used as-is in a production environment because additional configuration is required.

Achieving FFIEC-compliance requires that qualified auditors certify a production customer solution. Audits are overseen by examiners from FFIEC's member agencies, including the Board of Governors of the Federal Reserve System (FRB), the Federal Deposit Insurance Corporation (FDIC), the National Credit Union Administration (NCUA), the Office of the Comptroller of the Currency (OCC), and the Consumer Financial Protection Bureau (CFPB). These examiners certify that audits are completed by assessors who maintain independence from the audited institution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This Azure Security and Compliance Blueprint provides an analytics platform upon which customers can build their own analytics tools. The reference architecture outlines a generic use case where customers input data either through bulk data imports by the SQL/Data Administrator or through operational data updates via an Operational User. Both work streams incorporate Azure Functions for importing data into Azure SQL Database. Azure Functions must be configured by the customer through the Azure portal to handle the import tasks unique to each customer's own analytics requirements.

Azure offers a variety of reporting and analytics services for the customers. This solution incorporates Azure Machine Learning services in conjunction with Azure SQL Database to rapidly browse through data and deliver faster results through smarter modeling. Azure Machine Learning increases query speeds by discovering new relationships between datasets. Once the data has been trained through several statistical functions, up to 7 additional query pools (8 total including the customer server) can be synchronized with the same tabular models to spread query workloads and reduce response times.

For enhanced analytics and reporting, Azure SQL databases can be configured with columnstore indexes. Both Azure Machine Learning and Azure SQL databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

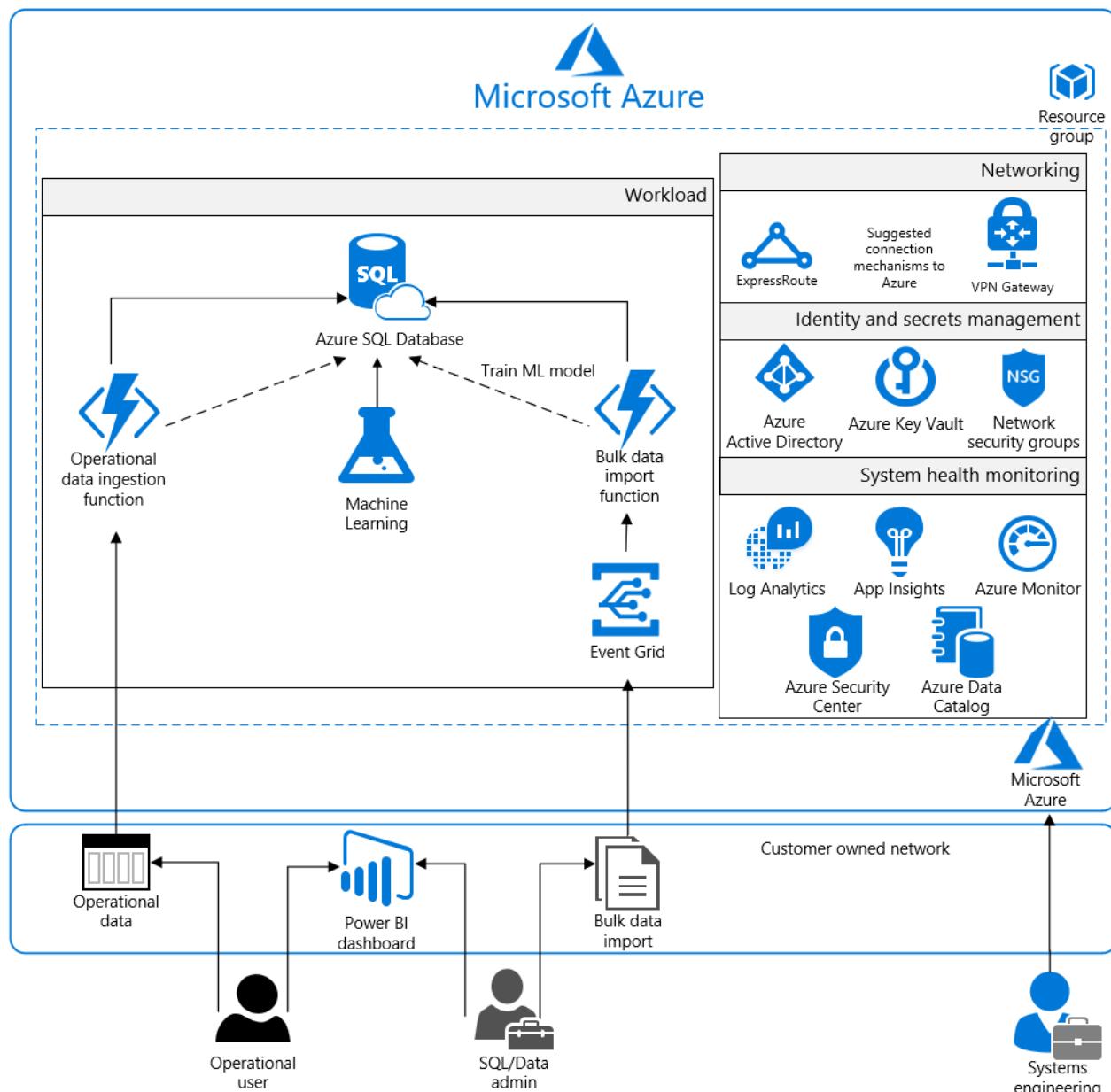
Once data is uploaded to the Azure SQL Database and trained by Azure Machine Learning, it is digested by both the Operational User and SQL/Data Admin with Power BI. Power BI displays data intuitively and pulls together information across multiple datasets to draw greater insight. Its high degree of adaptability and easy integration with Azure SQL Database ensures that customers can configure it to handle a wide array of scenarios as required by their business needs.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and again stored as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

Azure SQL Database is commonly managed through SQL Server Management Studio (SSMS), which runs from a local machine configured to access the Azure SQL Database via a secure VPN or ExpressRoute connection.

Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture resource group.



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- Application Insights
- Azure Active Directory

- Azure Data Catalog
- Azure Disk Encryption
- Azure Event Grid
- Azure Functions
- Azure Key Vault
- Azure Machine Learning
- Azure Monitor
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Log Analytics
- Azure Virtual Network
 - (1) /16 Network
 - (2) /24 Networks
 - (2) Network security groups
- Power BI Dashboard

Deployment architecture

The following section details the deployment and implementation elements.

Azure Event Grid: [Azure Event Grid](#) allows customers to easily build applications with event-based architectures. Users select the Azure resource they would like to subscribe to and give the event handler or webhook an endpoint to send the event to. Customers can secure webhook endpoints by adding query parameters to the webhook URL when creating an Event Subscription. Azure Event Grid only supports HTTPS webhook endpoints. Azure Event Grid allows customers to control the level of access given to different users to do various management operations such as list event subscriptions, create new ones, and generate keys. Event Grid utilizes Azure role-based access control.

Azure Functions: [Azure Functions](#) is a server-less compute service that enables users to run code on-demand without having to explicitly provision or manage infrastructure. Use Azure Functions to run a script or piece of code in response to a variety of events.

Azure Machine Learning service: [Azure Machine Learning](#) is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends.

Azure Data Catalog: [Data Catalog](#) makes data sources easily discoverable and understandable by the users who manage the data. Common data sources can be registered, tagged, and searched for financial data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual VM level. The following network security groups exist:

- A network security group for Active Directory
- A network security group for the workload

Each of the network security groups have specific ports and protocols open so that the solution can work securely

and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [network security group's diagnostic logs](#)

Subnets: Each subnet is associated with its corresponding network security group.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by FFIEC.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Extended Properties](#) can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated financial data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see

[Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).

- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

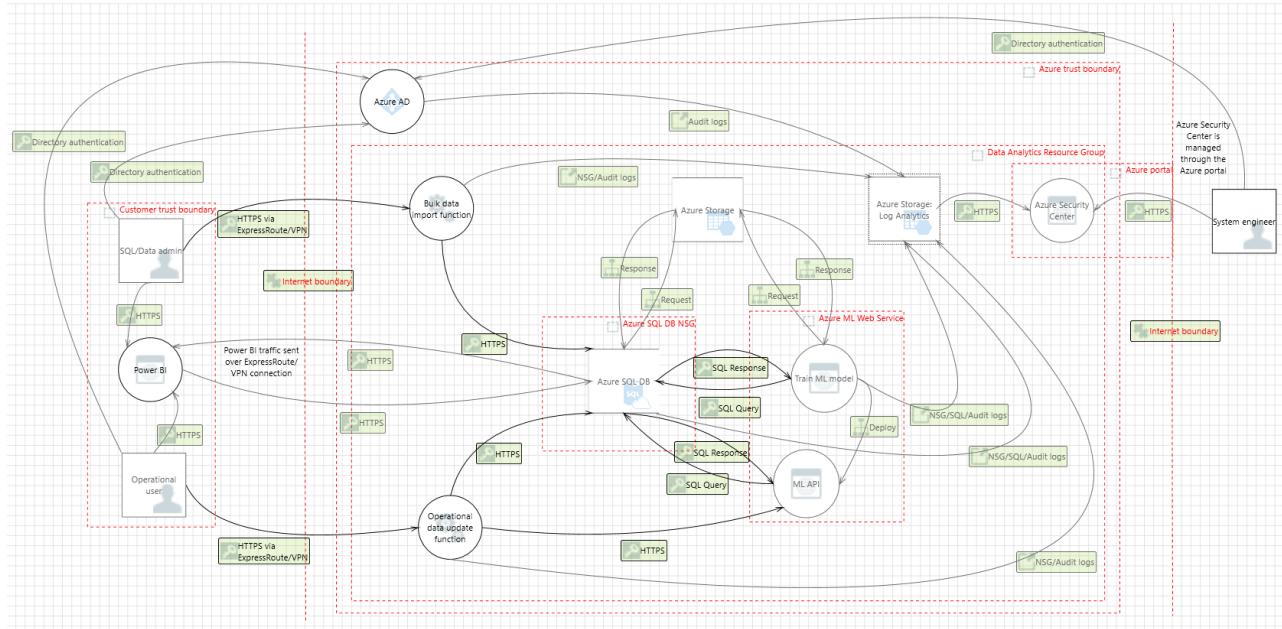
Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Application Insights: [Application Insights](#) is an extensible Application Performance Management (APM) service for web developers on multiple platforms. It detects performance anomalies and includes powerful analytics tools to help diagnose issues and to understand what users actually do with the app. It's designed to help users continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – FFIEC Customer Responsibility Matrix](#) lists all objectives required by FFIEC. This matrix details whether the implementation of each objective is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – FFIEC Data Analytics Implementation Matrix](#) provides information on which FFIEC objectives are addressed by the data analytics architecture, including detailed descriptions of how the implementation meets the requirements of each covered objective.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this data analytics reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-Transform-Load process

[PolyBase](#) can load data into Azure SQL Database without the need for a separate extract, transform, load or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing Azure Active Directory by making the deployed Active Directory infrastructure a subdomain of an Azure Active Directory forest.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.

- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Data Warehouse for FFIEC Financial Services

10/18/2018 • 18 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of a data warehouse architecture in Azure suitable for the collection, storage, and retrieval of financial data regulated by the Federal Financial Institution Examination Council (FFIEC).

This reference architecture, implementation guide, and threat model provide a foundation for customers to comply with FFIEC requirements. This solution provides a baseline to help customers deploy workloads to Azure in a FFIEC compliant manner, however, this solution should not be used as-is in a production environment because additional configuration is required.

Achieving FFIEC-compliance requires that qualified auditors certify a production customer solution. Audits are overseen by examiners from FFIEC's member agencies, including the Board of Governors of the Federal Reserve System (FRB), the Federal Deposit Insurance Corporation (FDIC), the National Credit Union Administration (NCUA), the Office of the Comptroller of the Currency (OCC), and the Consumer Financial Protection Bureau (CFPB). These examiners certify that audits are completed by assessors who maintain independence from the audited institution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture which implements a high-performance and secure cloud data warehouse. There are two separate data tiers in this architecture: one where data is imported, stored, and staged within a clustered SQL environment, and another for the Azure SQL Data Warehouse where the data is loaded using an extract, transform, load tool (e.g. [PolyBase](#) T-SQL queries) for processing. Once data is stored in Azure SQL Data Warehouse, analytics can run at a massive scale.

Azure offers a variety of reporting and analytics services for the customer. This solution includes SQL Server Reporting Services (SSRS) for quick creation of reports from the Azure SQL Data Warehouse. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

Data in the Azure SQL Data Warehouse is stored in relational tables with columnar storage, a format that significantly reduces the data storage costs while improving query performance. Depending on usage requirements, Azure SQL Data Warehouse compute resources can be scaled up or down or shut off completely if there are no active processes requiring compute resources.

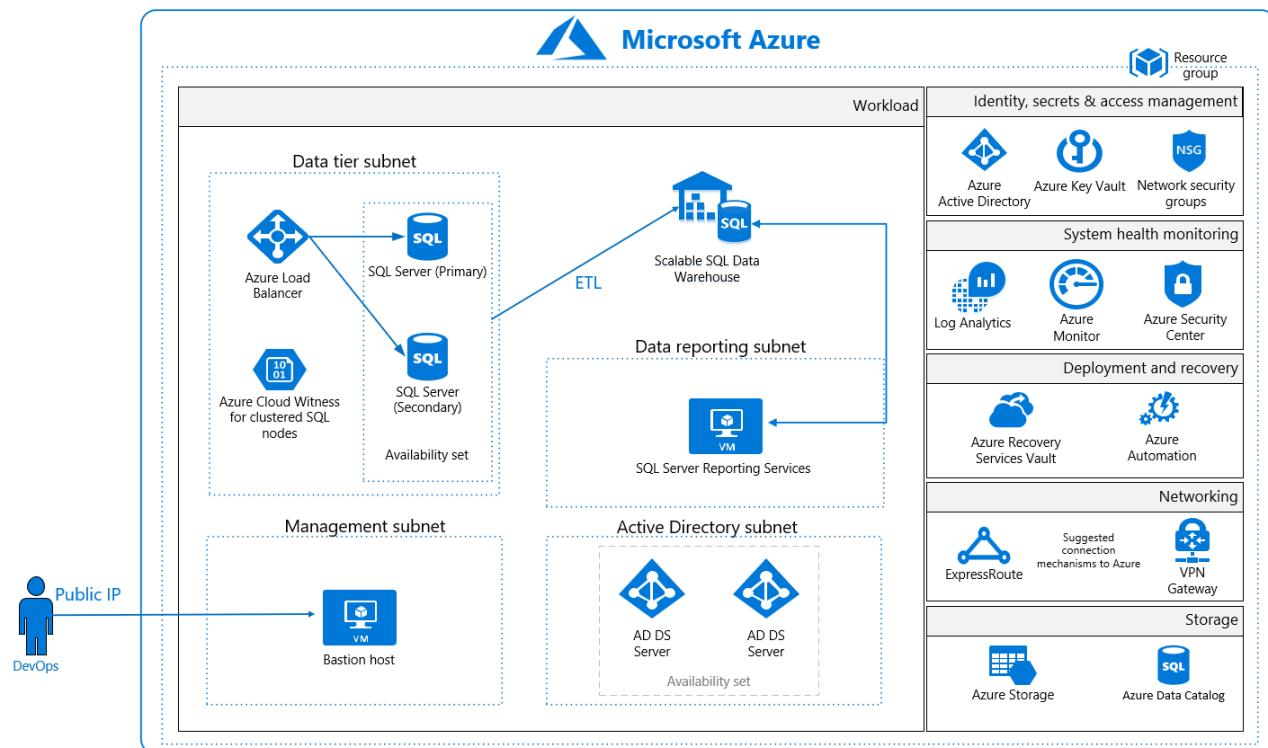
A SQL Load Balancer manages SQL traffic, ensuring high performance. All virtual machines in this reference architecture deploy in an availability set with SQL Server instances configured in an Always On availability group for high-availability and disaster-recovery capabilities.

This data warehouse reference architecture also includes an Active Directory tier for management of resources within the architecture. The Active Directory subnet enables easy adoption under a larger Active Directory forest structure, allowing for continuous operation of the environment even when access to the larger forest is unavailable. All virtual machines are domain-joined to the Active Directory tier and use Active Directory group policies to enforce security and compliance configurations at the operating system level.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and again stored as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

A virtual machine serves as a management bastion host, providing a secure connection for administrators to access deployed resources. The data loads into the staging area through this management bastion host. **Microsoft recommends configuring a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- Availability Sets
 - Active Directory Domain Controllers
 - SQL Cluster Nodes and Witness
- Azure Active Directory
- Azure Data Catalog
- Azure Key Vault
- Azure Monitor
- Azure Security Center
- Azure Load Balancer
- Azure Storage
- Azure Log Analytics
- Azure Virtual Machines

- (1) Bastion Host
- (2) Active Directory domain controller
- (2) SQL Server Cluster Node
- (1) SQL Server Witness
- Azure Virtual Network
 - (1) /16 Network
 - (4) /24 Networks
 - (4) Network security groups
- Recovery Services Vault
- SQL Data Warehouse
- SQL Server Reporting Services

Deployment architecture

The following section details the deployment and implementation elements.

SQL Data Warehouse: [SQL Data Warehouse](#) is an Enterprise Data Warehouse (EDW) that leverages Massively Parallel Processing (MPP) to quickly run complex queries across petabytes of data, allowing users to efficiently identify financial data. Users can use simple PolyBase T-SQL queries to import big data into the SQL Data Warehouse and utilize the power of MPP to run high-performance analytics.

SQL Server Reporting Services (SSRS): [SQL Server Reporting Services](#) provides quick creation of reports with tables, charts, maps, gauges, matrixes, and more for Azure SQL Data Warehouse.

Data Catalog: [Data Catalog](#) makes data sources easily discoverable and understandable by the users who manage the data. Common data sources can be registered, tagged, and searched for financial data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system.

Virtual network

This reference architecture defines a private virtual network with an address space of 10.0.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual virtual machine level. The following network security groups exist:

- A network security group for the Data Tier (SQL Server Clusters, SQL Server Witness, and SQL Load Balancer)
- A network security group for the management bastion host
- A network security group for Active Directory

- A network security group for SQL Server Reporting Services

Each of the network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [network security group's diagnostic logs](#)

Subnets: Each subnet is associated with its corresponding network security group.

Data at rest

The architecture protects data at rest through multiple measures, including encryption and database auditing.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by FFIEC.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Extended Properties](#) can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated financial data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to

protect sensitive data in Azure SQL Database.

- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for virtual machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Business continuity

High availability: Server workloads are grouped in an [Availability Set](#) to help ensure high availability of virtual machines in Azure. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure virtual machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS virtual machine without restoring the entire virtual machine, enabling faster restore times.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

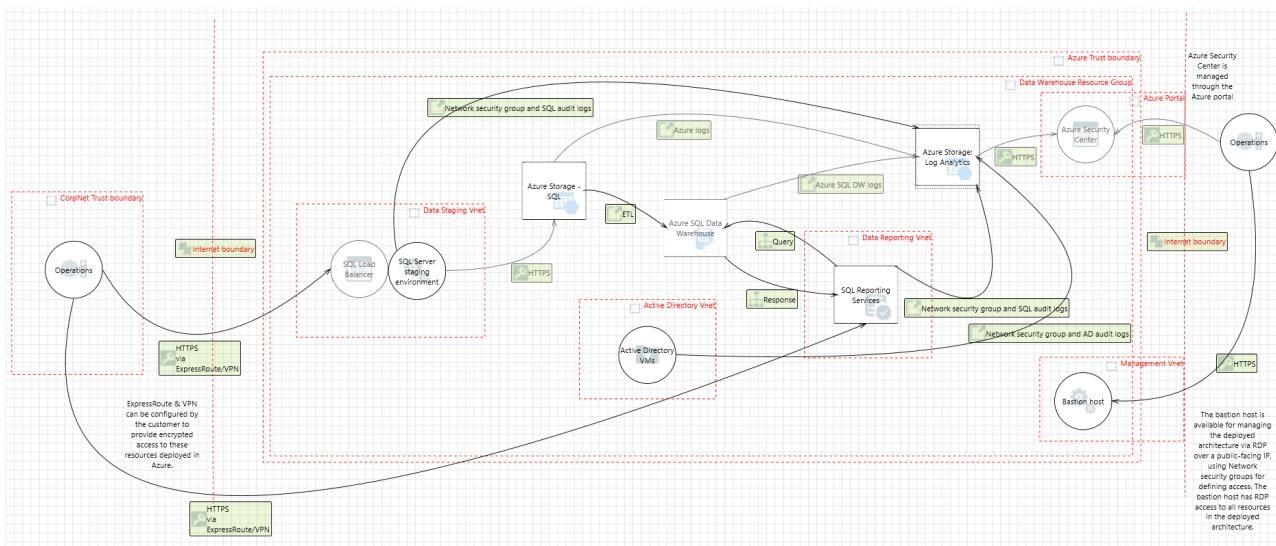
- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – FFIEC Customer Responsibility Matrix](#) lists all objectives required by FFIEC. This matrix details whether the implementation of each objective is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – FFIEC Data Warehouse Implementation Matrix](#) provides information on which FFIEC objectives are addressed by the data warehouse architecture, including detailed descriptions of how the implementation meets the requirements of each covered objective.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this data warehouse reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-Transform-Load process

[PolyBase](#) can load data into Azure SQL Data Warehouse without the need for a separate extract, transform, load or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

Azure Active Directory is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing Azure Active Directory by making the deployed Active Directory infrastructure a subdomain of an Azure Active Directory forest.

Optional services

Azure offers a variety of services to assist with the storage and staging of formatted and unformatted data. The following services can be added to this reference architecture depending on customer requirements:

- [Azure Data Factory](#) is a managed cloud service that is built for complex hybrid extract-transform-load, and data integration projects. Azure Data Factory has capabilities to help trace and locate financial data, including visualization and monitoring tools to identify when data arrived and where it came from. Using Azure Data Factory, customers can create and schedule data-driven workflows called pipelines that ingest data from disparate data stores. These pipelines allow customers to ingest data from both internal and external sources. Customers can then process and transform the data for output into data stores such as Azure SQL Data Warehouse.
- Customers can stage unstructured data in [Azure Data Lake Store](#), which enables the capture of data of any size, type, and ingestion speed in a single place for operational and exploratory analytics. Azure Data Lake has capabilities that enable the extraction and conversion of data. Azure Data Lake Store is compatible with most open source components in the Hadoop ecosystem and integrates nicely with other Azure services such as Azure SQL Data Warehouse.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: IaaS Web Application for FFIEC Financial Services

10/18/2018 • 15 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of an infrastructure as a service (IaaS) environment suitable for the collection, storage, and retrieval of financial data regulated by the Federal Financial Institution Examination Council (FFIEC).

This reference architecture, implementation guide, and threat model provide a foundation for customers to comply with FFIEC requirements. This solution provides a baseline to help customers deploy workloads to Azure in a FFIEC compliant manner, however, this solution should not be used as-is in a production environment because additional configuration is required.

Achieving FFIEC-compliance requires that qualified auditors certify a production customer solution. Audits are overseen by examiners from FFIEC's member agencies, including the Board of Governors of the Federal Reserve System (FRB), the Federal Deposit Insurance Corporation (FDIC), the National Credit Union Administration (NCUA), the Office of the Comptroller of the Currency (OCC), and the Consumer Financial Protection Bureau (CFPB). These examiners certify that audits are completed by assessors who maintain independence from the audited institution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

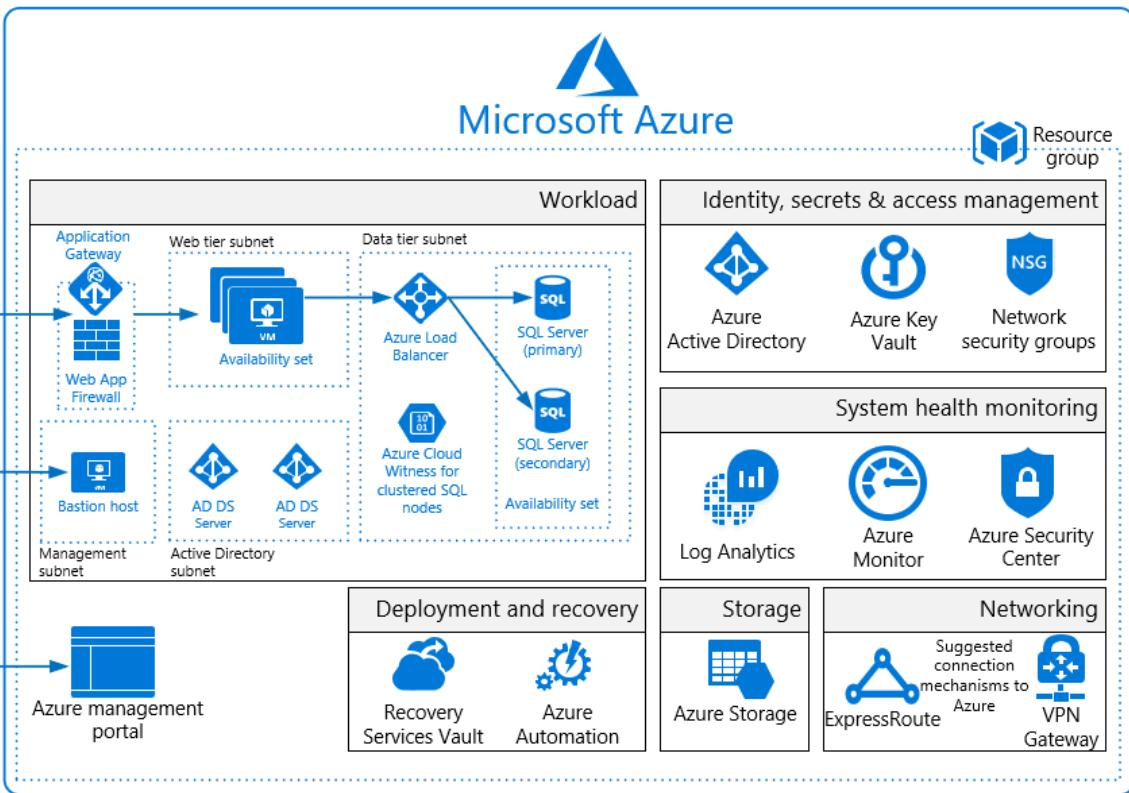
This Azure Security and Compliance Blueprint deploys a reference architecture for an IaaS web application with a SQL Server backend. The architecture includes a web tier, data tier, Active Directory infrastructure, Application Gateway, and Load Balancer. Virtual machines deployed to the web and data tiers are configured in an availability set, and SQL Server instances are configured in an Always On availability group for high availability. Virtual machines are domain-joined, and Active Directory group policies are used to enforce security and compliance configurations at the operating system level.

The entire solution is built upon Azure Storage which customers configure from the Azure portal. Azure Storage encrypts all data with Storage Service Encryption to maintain confidentiality of data at rest. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and again stored as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

A management bastion host provides a secure connection for administrators to access deployed resources.

Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

- Availability Sets
 - (1) Active Directory domain controllers
 - (1) SQL cluster nodes
 - (1) Web/IIS
- Azure Active Directory
- Azure Application Gateway
 - (1) Web Application Firewall
 - Firewall mode: prevention
 - Rule set: OWASP 3.0
 - Listener port: 443
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure Storage
 - (7) Geo-redundant storage accounts
- Azure Log Analytics
- Azure Virtual Machines
 - (1) management/bastion (Windows Server 2016 Datacenter)
 - (2) Active Directory domain controller (Windows Server 2016 Datacenter)
 - (2) SQL Server cluster node (SQL Server 2017 on Windows Server 2016)
 - (2) Web/IIS (Windows Server 2016 Datacenter)
- Azure Virtual Network
 - (1) /16 Network

- (5) /24 Networks
- (5) Network Security Groups
- Cloud Witness
- Recovery Services Vault

Deployment architecture

The following section details the deployment and implementation elements.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: This solution deploys resources in an architecture with a separate web subnet, database subnet, Active Directory subnet, and management subnet inside of a virtual network. Subnets are logically separated by network security group rules applied to the individual subnets to restrict traffic between subnets to only that necessary for system and management functionality.

See the configuration for [network security groups](#) deployed with this solution. Organizations can configure network security groups by editing the file above using [this documentation](#) as a guide.

Each of the subnets has a dedicated network Security Group:

- 1 network Security Group for Application Gateway (LBNSG)
- 1 network Security Group for bastion host (MGTNSG)
- 1 network Security Group for primary and backup domain controllers (ADNSG)
- 1 network Security Group for SQL Servers and Cloud Witness (SQLNSG)
- 1 network Security Group for web tier (WEBNSG)

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. Additionally, all transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by FFIEC.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-

encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.

- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.
- The solution is integrated with Azure Key Vault to manage IaaS virtual machine disk-encryption keys and secrets.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities using an Azure Application Gateway with a web application firewall configured, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Business continuity

High availability: The solution deploys all virtual machines in an [Availability Set](#). Availability sets ensure that the virtual machines are distributed across multiple isolated hardware clusters to improve availability. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS virtual machine without restoring the entire virtual machine, enabling faster restore times.

Cloud Witness: [Cloud Witness](#) is a type of Failover Cluster quorum witness in Windows Server 2016 that

leverages Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations, but it uses the standard publicly available Azure Blob Storage. This eliminates the extra maintenance overhead of virtual machines hosted in a public cloud.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

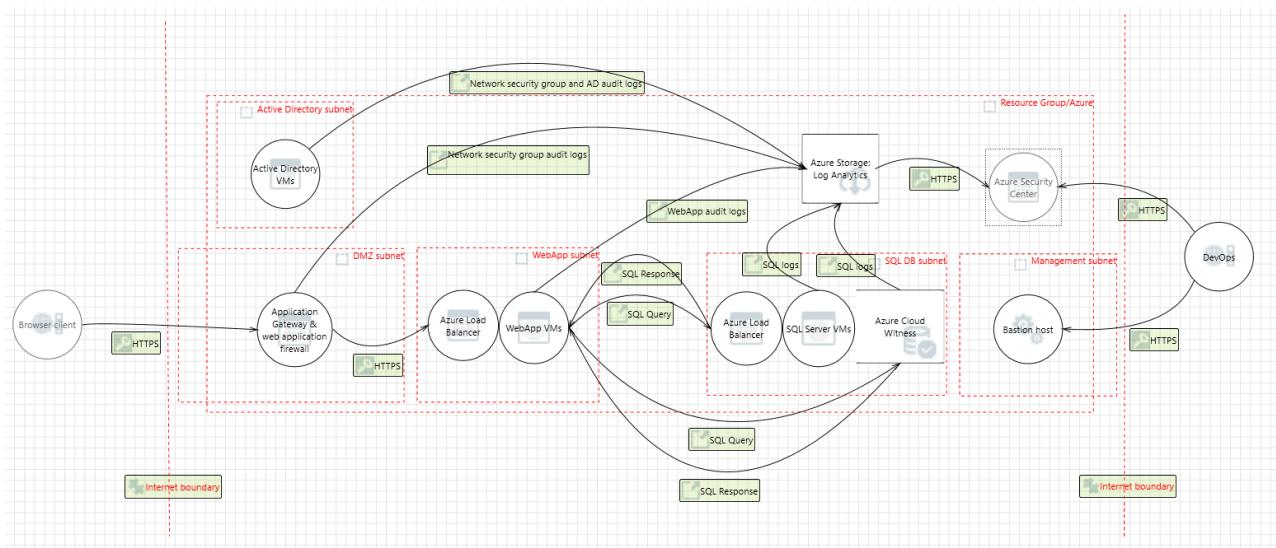
- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – FFIEC Customer Responsibility Matrix](#) lists all security objectives required by FFIEC. This matrix details whether the implementation of each objective is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – FFIEC IaaS Web Application Implementation Matrix](#) provides information on which FFIEC requirements are addressed by the IaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered objective.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this IaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.

- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: PaaS Web Application for FFIEC Financial Services

10/18/2018 • 18 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint Automation provides guidance for the deployment of a platform as a service (PaaS) environment suitable for the collection, storage, and retrieval of financial data regulated by the Federal Financial Institution Examination Council (FFIEC). This solution automates deployment and configuration of Azure resources for a common reference architecture, demonstrating ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own solutions on Azure. The solution implements a subset of requirements from the FFIEC handbooks. For more information about FFIEC requirements and this solution, see the [compliance documentation](#) section.

This Azure Security and Compliance Blueprint Automation deploys a PaaS web application reference architecture with pre-configured security controls to help customers achieve compliance with FFIEC requirements. The solution consists of Azure Resource Manager templates and PowerShell scripts that guide resource deployment and configuration.

This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment. Deploying an application into this environment without modification is not sufficient to completely meet the requirements of FFIEC objectives. Please note the following:

- This architecture provides a baseline to help customers use Azure in a FFIEC-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessment of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Achieving FFIEC-compliance requires that qualified auditors certify a production customer solution. Audits are overseen by examiners from FFIEC's member agencies, including the Board of Governors of the Federal Reserve System (FRB), the Federal Deposit Insurance Corporation (FDIC), the National Credit Union Administration (NCUA), the Office of the Comptroller of the Currency (OCC), and the Consumer Financial Protection Bureau (CFPB). These examiners certify that audits are completed by assessors who maintain independence from the audited institution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Click [here](#) for deployment instructions.

Architecture diagram and components

This Azure Security and Compliance Blueprint Automation deploys a reference architecture for a PaaS web application with an Azure SQL Database backend. The web application is hosted in an isolated Azure App Service Environment, which is a private, dedicated environment in an Azure datacenter. The environment load balances traffic for the web application across virtual machines managed by Azure. This architecture also includes network security groups, an Application Gateway, Azure DNS, and Load Balancer.

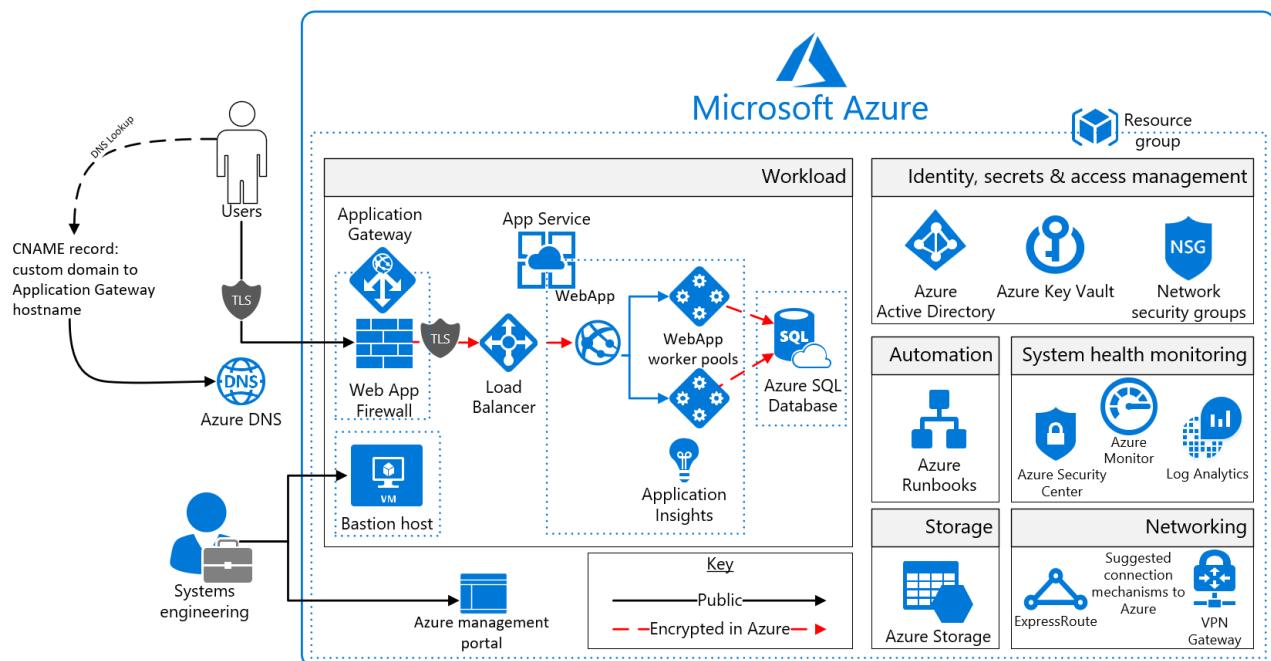
For enhanced analytics and reporting, Azure SQL databases can be configured with columnstore indexes. Azure SQL databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and stored again as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

Azure SQL Database is commonly managed through SQL Server Management Studio, which runs from a local machine configured to access the Azure SQL Database via a secure VPN or ExpressRoute connection.

Furthermore, Application Insights provides real time application performance management and analytics through Log Analytics. **Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are in the [deployment architecture](#) section.

- Application Gateway
 - Web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP
 - Listener port: 443
- Application Insights
- Azure Active Directory
- Azure Application Service Environment v2
- Azure Automation
- Azure DNS
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager

- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Log Analytics
- Azure Virtual Network
 - (1) /16 Network
 - (4) /24 Networks
 - Network security groups
- Azure Web App

Deployment architecture

The following section details the deployment and implementation elements.

Azure Resource Manager: [Azure Resource Manager](#) enables customers to work with the resources in the solution as a group. Customers can deploy, update, or delete all the resources for the solution in a single, coordinated operation. Customers use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help customers manage their resources after deployment.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

App Service Environment v2: The Azure App Service Environment is an App Service feature that provides a fully isolated and dedicated environment for securely running App Service applications at a high scale. This isolation feature is required to meet FFIEC compliance requirements.

App Service Environments are isolated to only run a single customer's applications and are always deployed into a virtual network. This isolation feature enables the reference architecture to have complete tenant isolation, removing it from Azure's multi-tenant environment prohibiting those multi-tenants from enumerating the deployed App Service Environment resources. Customers have fine-grained control over both inbound and outbound application network traffic, and applications can establish high-speed secure connections over virtual networks to on-premises corporate resources. Customers can "auto-scale" with App Service Environment based on load metrics, available budget, or a defined schedule.

Use of App Service Environment for this architecture allows for the following configurations:

- Host inside a secured Azure virtual network and network security rules
- Self-signed internal load balancer certificate for HTTPS communication. As a best practice, Microsoft recommends the use of a trusted certificate authority for enhanced security.
- [Internal Load Balancing mode](#)
- Disable [TLS 1.0](#)
- Change [TLS cipher](#)

- Control [inbound traffic N/W ports](#)
- [Web application firewall – restrict data](#)
- Allow [Azure SQL Database traffic](#)

Azure Web App: [Azure Web Apps](#) enables customers to build and host web applications in the programming language of their choice without managing infrastructure. It offers auto-scaling and high availability, supports both Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repo.

Virtual Network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual virtual machine level. The following network security groups exist:

- 1 network security group for Application Gateway
- 1 network security group for App Service Environment
- 1 network security group for Azure SQL Database
- 1 network Security Group for bastion host

Each of the network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [network security group's diagnostics logs](#)

Subnets: Each subnet is associated with its corresponding network security group.

Azure DNS: The Domain Name System, or DNS, is responsible for translating (or resolving) a website or service name to its IP address. [Azure DNS](#) is a hosting service for DNS domains that provides name resolution using Azure infrastructure. By hosting domains in Azure, users can manage DNS records using the same credentials, APIs, tools, and billing as other Azure services. Azure DNS also supports private DNS domains.

Azure Load Balancer: [Azure Load Balancer](#) allows customers to scale their applications and create high availability for services. Load Balancer supports inbound as well as outbound scenarios, and provides low latency, high throughput, and scales up to millions of flows for all TCP and UDP applications.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by FFIEC.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.

- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities using an Azure Application Gateway with a web application firewall configured, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- **Active Directory Assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **SQL Assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic

distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.

- **Activity Log Analytics:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

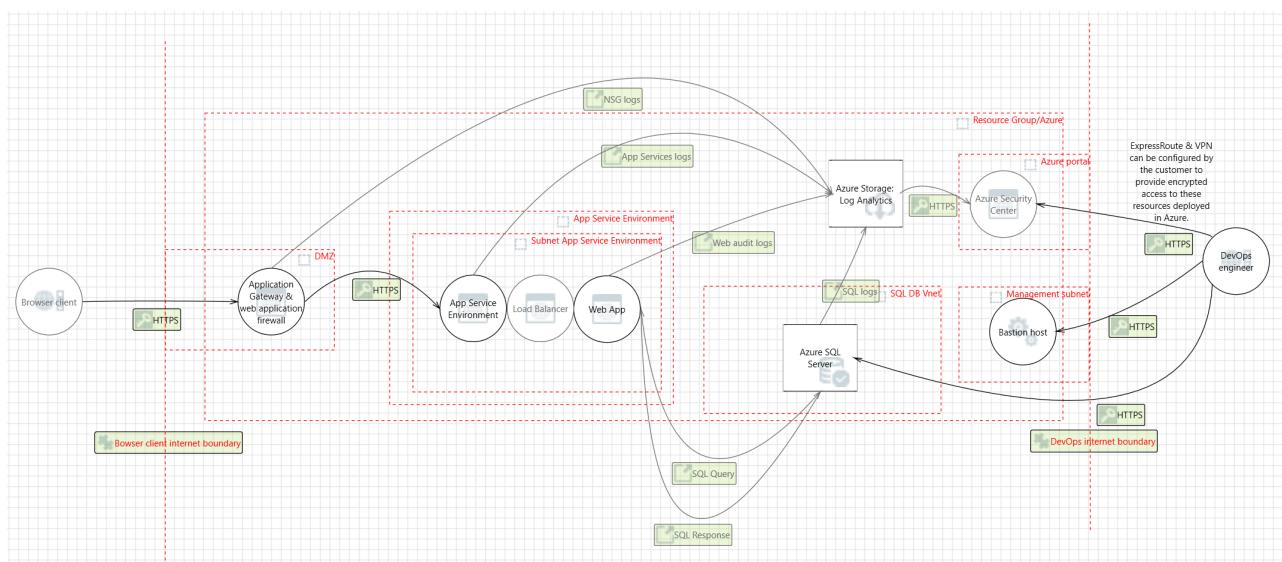
Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Application Insights: [Application Insights](#) is an extensible Application Performance Management service for web developers on multiple platforms. Application Insights detects performance anomalies and customers can use it to monitor the live web application. It includes powerful analytics tools to help customers diagnose issues and to understand what users actually do with their app. It's designed to help customers continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – FFIEC Customer Responsibility Matrix](#) lists all security objectives required by FFIEC. This matrix details whether the implementation of each objective is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – FFIEC PaaS Web Application Implementation Matrix](#) provides information on which FFIEC requirements are addressed by the PaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered objective.

Deploy this solution

This Azure Security and Compliance Blueprint Automation is comprised of JSON configuration files and PowerShell scripts that are handled by Azure Resource Manager's API service to deploy resources within Azure. Detailed deployment instructions are available [here](#).

1. Clone or download [this](#) GitHub repository to your local workstation.
2. Review 0-Setup-AdministrativeAccountAndPermission.md and run the provided commands.
3. Deploy a test solution with Contoso sample data or pilot an initial production environment.
 - 1A-ContosoWebStoreDemoAzureResources.ps1
 - This script deploys Azure resources for a demonstration of a webstore using Contoso sample data.
 - 1-DeployAndConfigureAzureResources.ps1
 - This script deploys the Azure resources needed for supporting a production environment for a customer-owned web application. This environment should be further customized by the customer based on organizational requirements.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this PaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Analytics for GDPR

12/6/2018 • 16 minutes to read • [Edit Online](#)

Overview

The General Data Protection Regulation (GDPR) contains many requirements about collecting, storing, and using personal information, including how organizations identify and secure personal data, accommodate transparency requirements, detect and report personal data breaches, and train privacy personnel and other employees. The GDPR gives individuals greater control over their personal data and imposes many new obligations on organizations that collect, handle, or analyze personal data. The GDPR imposes new rules on organizations that offer goods and services to people in the European Union (EU), or that collect and analyze data tied to EU residents. The GDPR applies no matter where an organization is located.

Microsoft designed Azure with industry-leading security measures and privacy policies to safeguard data in the cloud, including the categories of personal data identified by the GDPR. Microsoft's [contractual terms](#) commit Microsoft to the requirements of processors.

This Azure Security and Compliance Blueprint provides guidance to deploy a data analytics architecture in Azure that assists with the requirements of the GDPR. This solution demonstrates ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own data analytics solutions in Azure. Customers can utilize this reference architecture and follow Microsoft's [four-step process](#) in their journey to GDPR compliance:

1. Discover: Identify which personal data exists and where it resides.
2. Manage: Govern how personal data is used and accessed.
3. Protect: Establish security controls to prevent, detect, and respond to vulnerabilities and data breaches.
4. Report: Keep required documentation and manage data requests and breach notifications.

This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements and should not be used as-is in a production environment. Please note the following:

- The architecture provides a baseline to help customers deploy workloads to Azure in a GDPR-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides an analytics platform upon which customers can build their own analytics tools. The reference architecture outlines a generic use case where customers input data either through bulk data imports by the SQL/Data Administrator or through operational data updates via an Operational User. Both work streams incorporate Azure Functions for importing data into Azure SQL Database. Azure Functions must be configured by the customer through the Azure portal to handle the import tasks unique to each customer's own analytics requirements.

Azure offers a variety of reporting and analytics services for the customer; however, this solution incorporates Azure Machine Learning services in conjunction with Azure SQL Database to rapidly browse through data and

deliver faster results through smarter modeling of customer data. Azure Machine Learning is a form of machine learning intended to increase query speeds by discovering new relationships between datasets. Once the data has been trained through several statistical functions, up to 7 additional query pools (8 total including the customer server) can be synchronized with the same tabular models to spread query workload and reduce response times.

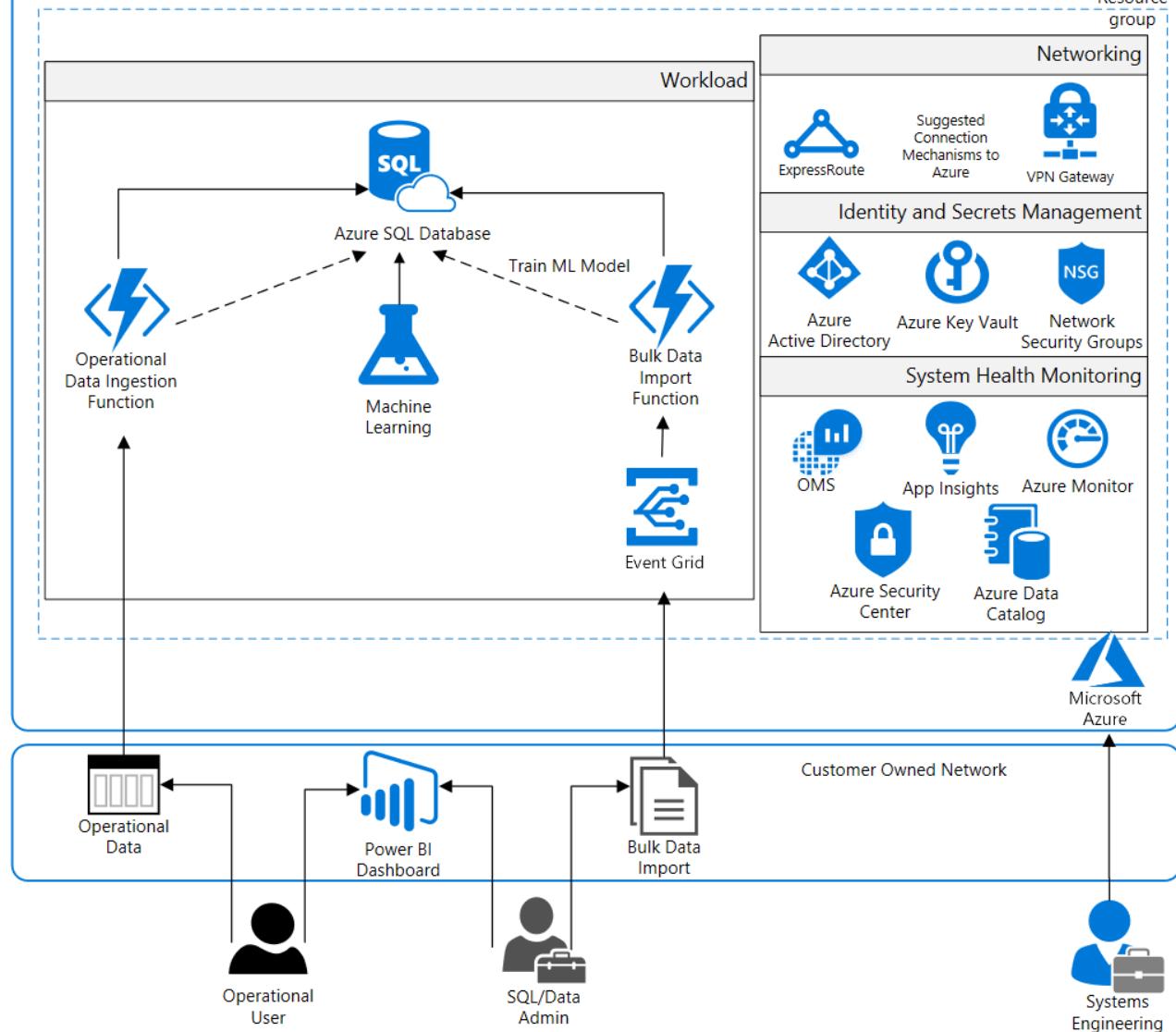
For enhanced analytics and reporting, Azure SQL databases can be configured with columnstore indexes. Both Azure Machine Learning and Azure SQL databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

Once data is uploaded to the Azure SQL Database and trained by Azure Machine Learning, it is digested by both the Operational User and SQL/Data Admin with Power BI. Power BI displays data intuitively and pulls together information across multiple datasets to draw greater insight. Its high degree of adaptability and easy integration with Azure SQL Database ensures that customers can configure it to handle a wide array of scenarios as required by their business needs.

The entire solution is built upon Azure Storage which customers configure from the Azure portal. Azure Storage encrypts all data with Storage Service Encryption to maintain confidentiality of data at rest. Geographic Redundant Storage (GRS) ensures that an adverse event at the customer's primary data center will not result in a loss of data as a second copy will be stored in a separate location hundreds of miles away.

For enhanced security, this architecture manages resources with Azure Active Directory and Azure Key Vault. System health is monitored through Log Analytics and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

Azure SQL Database is commonly managed through SQL Server Management Studio (SSMS), which runs from a local machine configured to access the Azure SQL Database via a secure VPN or ExpressRoute connection. **Azure recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture resource group.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- Azure Functions
- Azure SQL Database
- Azure Machine Learning
- Azure Active Directory
- Azure Key Vault
- Log Analytics
- Azure Monitor
- Azure Storage
- Power BI Dashboard
- Azure Data Catalog
- Azure Security Center
- Application Insights
- Azure Event Grid
- network security groups

Deployment architecture

The following section details the deployment and implementation elements.

Azure Event Grid [Azure Event Grid](#) allows customers to easily build applications with event-based architectures.

Users select the Azure resource they would like to subscribe to, and give the event handler or webhook an endpoint to send the event to. Customers can secure webhook endpoints by adding query parameters to the webhook URL when creating an Event Subscription. Azure Event Grid only supports HTTPS webhook endpoints. Azure Event Grid allows customers to control the level of access given to different users to do various management operations such as list event subscriptions, create new ones, and generate keys. Event Grid utilizes Azure Role-Based Access Control (RBAC).

Azure Functions [Azure Functions](#) is a server-less compute service that enables users to run code on-demand without having to explicitly provision or manage infrastructure. Use Azure Functions to run a script or piece of code in response to a variety of events.

Azure Machine Learning service [Azure Machine Learning](#) is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends.

Azure Data Catalog: [Data Catalog](#) makes data sources easily discoverable and understandable by the users who manage the data. Common data sources can be registered, tagged, and searched for personal data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

Virtual network

This reference architecture defines a private VNet with an address space of 10.0.0.0/16.

Network security groups: [NSGs](#) contain Access Control Lists (ACLs) that allow or deny traffic within a VNet. NSGs can be used to secure traffic at a subnet or individual VM level. The following NSGs exist:

- An NSG for Active Directory
- An NSG for the workload

Each of the NSGs have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [NSG's diagnostics](#)

Subnets: Each subnet is associated with its corresponding NSG.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard personal data in support of organizational security commitments and compliance requirements defined by the GDPR.

Azure Disk Encryption [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [AD authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. TDE provides assurance that stored personal data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted Columns](#) ensure that sensitive personal data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Extended Properties](#) can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated personal data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Dynamic Data Masking \(DDM\)](#) limits sensitive personal data exposure by masking the data to non-privileged users or applications. DDM can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps with the identification of personal data qualifying for GDPR protection, and for reducing access such that it does not exit the database via unauthorized access. **Note:**
Customers will need to adjust DDM settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to personal data in the Azure environment:

- [Azure Active Directory \(AAD\)](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in AAD, including users accessing Azure SQL Database.
- Authentication to the application is performed using AAD. For more information, see [Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses AAD to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in SQL Database](#).
- [Azure Role-Based Access Control \(RBAC\)](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permissions for Azure resources, administrators can allow only certain actions for accessing personal data. Subscription access is limited to the subscription administrator.
- [AAD Privileged Identity Management \(PIM\)](#) enables customers to minimize the number of users who have access to certain information such as personal data. Administrators can use AAD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [AAD Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect personal data and access to such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.

- All keys in Key Vault are protected by specialized hardware security modules (HSMs). The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Security alerts: Azure Security Center enables customers to monitor traffic, collect logs, and analyze data sources for threats. Additionally, Azure Security Center accesses existing configuration of Azure services to provide configuration and service recommendations to help improve security posture and protect personal data. Azure Security Center includes a [threat intelligence report](#) for each detected threat to assist incident response teams investigate and remediate threats.

Logging and auditing

Log Analytics provides extensive logging of system and user activity, as well as system health. The Log Analytics solution collects and analyzes data generated by resources in Azure and on-premises environments.

- **Activity logs:** Activity logs provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** Diagnostic logs include all logs emitted by every resource. These logs include Windows event system logs and Azure Blob storage, tables, and queue logs.
- **Log archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival with a defined retention period of 2 days. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

Additionally, the following monitoring solutions are included as a part of this architecture:

- [AD Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [Antimalware Assessment](#): The Antimalware solution reports on malware, threats, and protection status.
- [Azure Automation](#): The Azure Automation solution stores, runs, and manages runbooks.
- [Security and Audit](#): The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Update Management](#): The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Azure Activity Logs](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.
- [Change Tracking](#): The Change Tracking solution allows customers to easily identify changes in the environment.

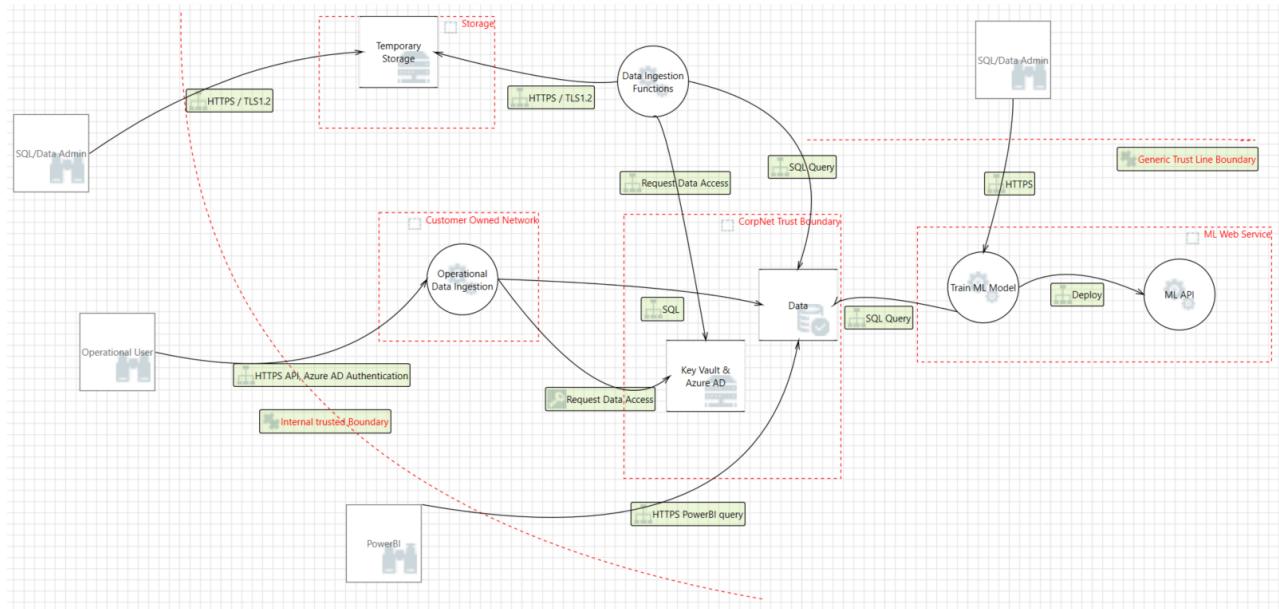
Azure Monitor Azure Monitor helps customers track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in customers' Azure resources.

Application Insights Application Insights is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Use it to monitor live web application. It detects performance anomalies and includes powerful analytics tools to help diagnose issues and to understand what users actually do with the

app. It's designed to help users continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – GDPR Customer Responsibility Matrix](#) lists controller and processor responsibilities for all GDPR articles. Please note that for Azure services, a customer is usually the controller and Microsoft acts as the processor.

The [Azure Security and Compliance Blueprint - GDPR Data Analytics Implementation Matrix](#) provides information on which GDPR articles are addressed by the data analytics architecture, including detailed descriptions of how the implementation meets the requirements of each covered article.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this data analytics reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely “tunnel” information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are

available.

Extract-Transform-Load (ETL) process

[PolyBase](#) can load data into Azure SQL Database without the need for a separate ETL or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with AAD in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing AAD by making the deployed Active Directory infrastructure a subdomain of an AAD forest.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Data Warehouse for GDPR

11/7/2018 • 17 minutes to read • [Edit Online](#)

Overview

The General Data Protection Regulation (GDPR) contains many requirements about collecting, storing, and using personal information, including how organizations identify and secure personal data, accommodate transparency requirements, detect and report personal data breaches, and train privacy personnel and other employees. The GDPR gives individuals greater control over their personal data and imposes many new obligations on organizations that collect, handle, or analyze personal data. The GDPR imposes new rules on organizations that offer goods and services to people in the European Union (EU), or that collect and analyze data tied to EU residents. The GDPR applies no matter where an organization is located.

Microsoft designed Azure with industry-leading security measures and privacy policies to safeguard data in the cloud, including the categories of personal data identified by the GDPR. Microsoft's [contractual terms](#) commit Microsoft to the requirements of processors.

This Azure Security and Compliance Blueprint provides guidance to deploy a data warehouse architecture in Azure that assists with the requirements of the GDPR. This solution demonstrates ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own data warehouse solutions in Azure. Customers can utilize this reference architecture and follow Microsoft's [four-step process](#) in their journey to GDPR compliance:

1. Discover: Identify which personal data exists and where it resides.
2. Manage: Govern how personal data is used and accessed.
3. Protect: Establish security controls to prevent, detect, and respond to vulnerabilities and data breaches.
4. Report: Keep required documentation and manage data requests and breach notifications.

This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements and should not be used as-is in a production environment. Please note the following:

- The architecture provides a baseline to help customers deploy workloads to Azure in a GDPR-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture which implements a high-performance and secure cloud data warehouse. There are two separate data tiers in this architecture: one where data is imported, stored, and staged within a clustered SQL environment, and another for the Azure SQL Data Warehouse where the data is loaded using an ETL tool (e.g. [PolyBase](#) T-SQL queries) for processing. Once data is stored in Azure SQL Data Warehouse, analytics can run at a massive scale.

Azure offers a variety of reporting and analytics services for the customer. This solution includes SQL Server Reporting Services (SSRS) for quick creation of reports from the Azure SQL Data Warehouse. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use

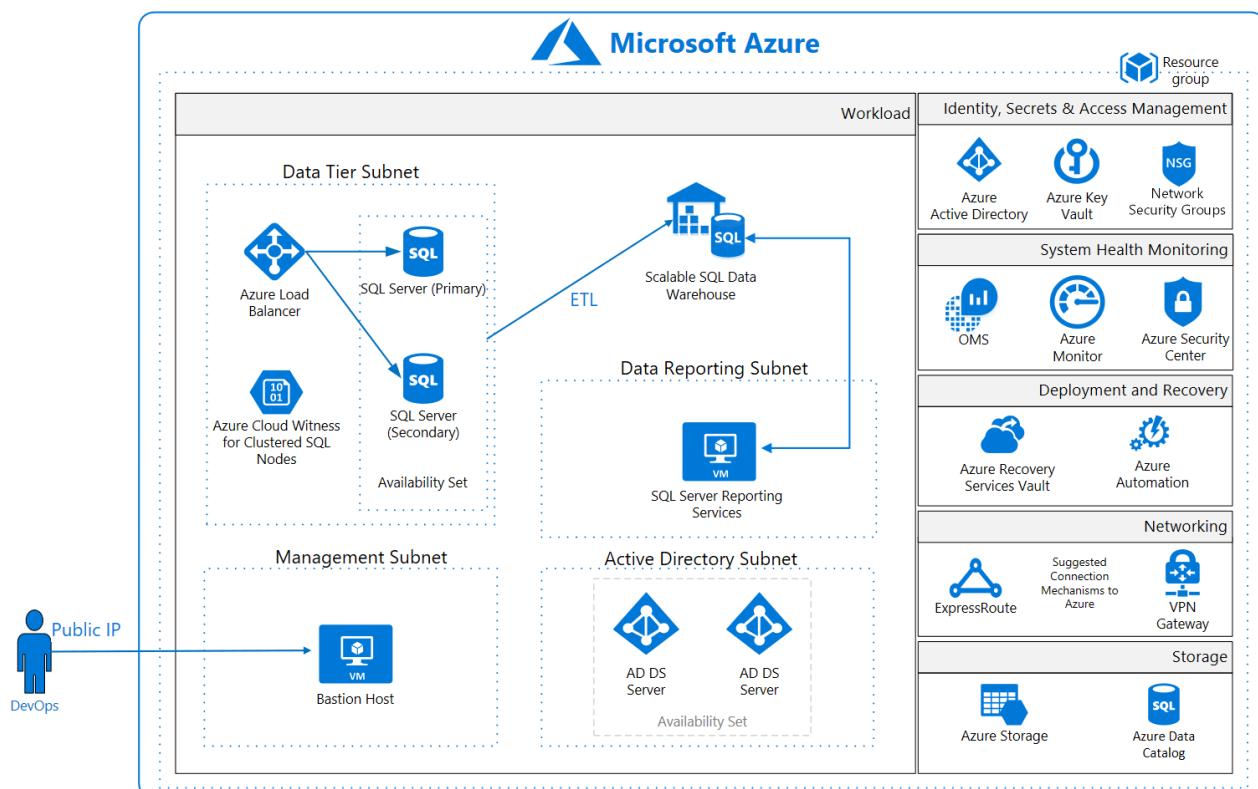
of a trusted certificate authority for enhanced security.

Data in the Azure SQL Data Warehouse is stored in relational tables with columnar storage, a format that significantly reduces the data storage costs while improving query performance. Depending on usage requirements, Azure SQL Data Warehouse compute resources can be scaled up or down or shut off completely if there are no active processes requiring compute resources.

A SQL Load Balancer manages SQL traffic, ensuring high performance. All virtual machines in this reference architecture deploy in an availability set with SQL Server instances configured in an AlwaysOn availability group for high-availability and disaster-recovery capabilities.

This data warehouse reference architecture also includes an Active Directory (AD) tier for management of resources within the architecture. The Active Directory subnet enables easy adoption under a larger AD forest structure, allowing for continuous operation of the environment even when access to the larger forest is unavailable. All virtual machines are domain-joined to the Active Directory tier and use Active Directory group policies to enforce security and compliance configurations at the operating system level.

A virtual machine serves as a management bastion host, providing a secure connection for administrators to access deployed resources. The data loads into the staging area through this management bastion host. **Azure recommends configuring a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- Azure Virtual Machines
 - (1) Bastion Host
 - (2) Active Directory domain controller
 - (2) SQL Server Cluster Node
 - (1) SQL Server Witness
- Availability Sets
 - Active Directory Domain Controllers
 - SQL Cluster Nodes and Witness

- Virtual Network
 - (4) Subnets
 - (4) Network Security Groups
- SQL Data Warehouse
- SQL Server Reporting Services
- Azure SQL Load Balancer
- Azure Active Directory
- Recovery Services Vault
- Azure Key Vault
- Log Analytics
- Azure Data Catalog
- Azure Security Center

Deployment architecture

The following section details the deployment and implementation elements.

SQL Data Warehouse: [SQL Data Warehouse](#) is an Enterprise Data Warehouse (EDW) that leverages Massively Parallel Processing (MPP) to quickly run complex queries across petabytes of data, allowing users to efficiently identify personal data. Users can use simple PolyBase T-SQL queries to import big data into the SQL Data Warehouse and utilize the power of MPP to run high-performance analytics.

SQL Server Reporting Services (SSRS): [SQL Server Reporting Services](#) provides quick creation of reports with tables, charts, maps, gauges, matrixes, and more for Azure SQL Data Warehouse.

Data Catalog: [Data Catalog](#) makes data sources easily discoverable and understandable by the users who manage the data. Common data sources can be registered, tagged, and searched for personal data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the Network Security Group (NSG).

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Log Analytics extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

Virtual network

This reference architecture defines a private VNet with an address space of 10.0.0.0/16.

Network security groups: [NSGs](#) contain Access Control Lists (ACLs) that allow or deny traffic within a VNet. NSGs can be used to secure traffic at a subnet or individual VM level. The following NSGs exist:

- An NSG for the Data Tier (SQL Server Clusters, SQL Server Witness, and SQL Load Balancer)
- An NSG for the management bastion host

- An NSG for Active Directory
- An NSG for SQL Server Reporting Services

Each of the NSGs have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [NSG's diagnostics](#)

Subnets: Each subnet is associated with its corresponding NSG.

Data at rest

The architecture protects data at rest through multiple measures, including encryption and database auditing.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard personal data in support of organizational security commitments and compliance requirements defined by the GDPR.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for OS and data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [AD authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. TDE provides assurance that stored personal data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted Columns](#) ensure that sensitive personal data never appears as plaintext inside the database system. After enabling data encryption, only client applications or app servers with access to the keys can access plaintext data.
- The [Extended Properties](#) feature can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated personal data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database Dynamic Data Masking \(DDM\)](#) limits sensitive personal data exposure by masking the data to non-privileged users or applications. DDM can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps with the identification of personal data qualifying for GDPR protection, and for reducing access such that it does not exit the database via unauthorized access. **Note:**
Customers will need to adjust DDM settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to personal data in the Azure environment:

- [Azure Active Directory \(AAD\)](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in AAD, including users accessing the SQL Database.
- Authentication to the application is performed using AAD. For more information, see [Integrating applications](#)

with Azure Active Directory. Additionally, the database column encryption uses AAD to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in SQL Database](#).

- [Azure Role-based Access Control \(RBAC\)](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permissions for Azure resources, administrators can allow only certain actions for accessing personal data. Subscription access is limited to the subscription administrator.
- [AAD Privileged Identity Management \(PIM\)](#) enables customers to minimize the number of users who have access to certain information such as personal data. Administrators can use AAD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [AAD Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect personal data and access to such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules (HSMs). The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Security alerts: [Azure Security Center](#) enables customers to monitor traffic, collect logs, and analyze data sources for threats. Additionally, Azure Security Center accesses existing configuration of Azure services to provide configuration and service recommendations to help improve security posture and protect personal data. Azure Security Center includes a [threat intelligence report](#) for each detected threat to assist incident response teams investigate and remediate threats.

Business continuity

High availability: Server workloads are grouped in an [Availability Set](#) to help ensure high availability of virtual machines in Azure. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS VM without restoring the entire VM, enabling faster restore times.

Logging and auditing

[Log Analytics](#) provides extensive logging of system and user activity, as well as system health. The [Log Analytics](#) solution collects and analyzes data generated by resources in Azure and on-premises environments.

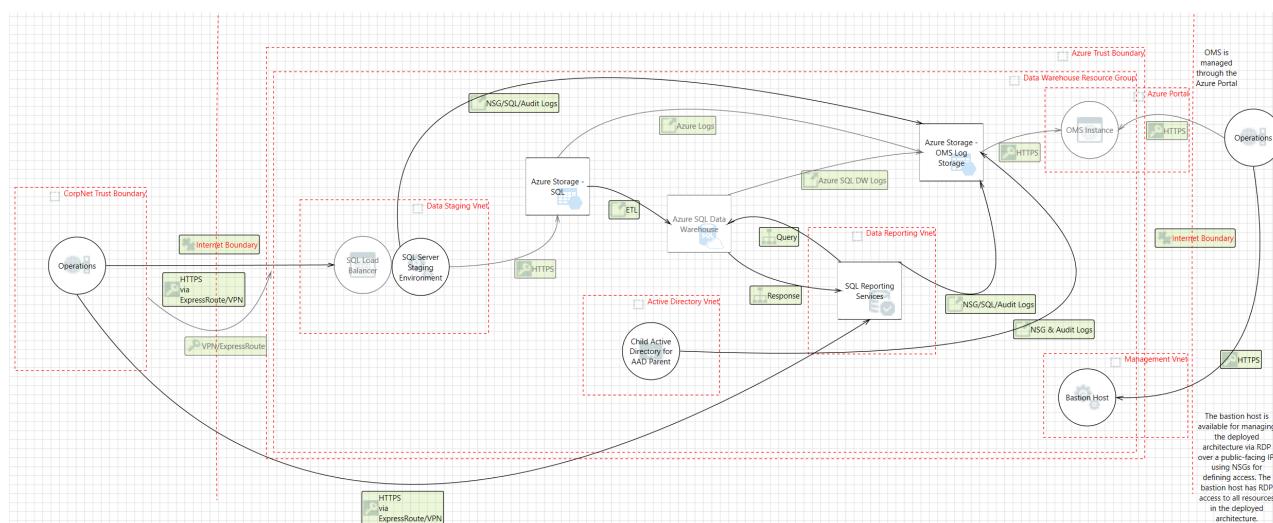
- **Activity Logs:** Activity logs provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic Logs:** Diagnostic logs include all logs emitted by every resource. These logs include Windows event system logs and Azure Blob storage, tables, and queue logs.
- **Log Archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

Additionally, the following Log Analytics solutions are included as a part of this architecture:

- **AD Assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **Antimalware Assessment:** The Antimalware solution reports on malware, threats, and protection status.
- **Azure Automation:** The Azure Automation solution stores, runs, and manages runbooks.
- **Security and Audit:** The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- **SQL Assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- **Update Management:** The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- **Azure Activity Logs:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.
- **Change Tracking:** The Change Tracking solution allows customers to easily identify changes in the environment.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – GDPR Customer Responsibility Matrix](#) lists controller and

processor responsibilities for all GDPR articles. Please note that for Azure services, a customer is usually the controller and Microsoft acts as the processor.

The [Azure Security and Compliance Blueprint - GDPR Data Warehouse Implementation Matrix](#) provides information on which GDPR articles are addressed by the data warehouse architecture, including detailed descriptions of how the implementation meets the requirements of each covered article.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this data warehouse reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely “tunnel” information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-Transform-Load (ETL) process

[PolyBase](#) can load data into Azure SQL Data Warehouse without the need for a separate ETL or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with AAD in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing AAD by making the deployed Active Directory infrastructure a subdomain of an AAD forest.

Optional services

Azure offers a variety of services to assist with the storage and staging of formatted and unformatted data. The following services can be added to this reference architecture depending on customer requirements:

- [Azure Data Factory](#) is a managed cloud service that is built for complex hybrid extract-transform-load (ETL), extract-load-transform (ELT), and data integration projects. Azure Data Factory has capabilities to help trace and locate personal data, including visualization and monitoring tools to identify when data arrived and where it came from. Using Azure Data Factory, customers can create and schedule data-driven workflows called pipelines that ingest data from disparate data stores. These pipelines allow customers to ingest data from both internal and external sources. Customers can then process and transform the data for output into data stores such as Azure SQL Data Warehouse.
- Customers can stage unstructured data in [Azure Data Lake Store](#), which enables the capture of data of any size, type, and ingestion speed in a single place for operational and exploratory analytics. Azure Data Lake has capabilities that enable the extraction and conversion of data. Azure Data Lake Store is compatible with most

open source components in the Hadoop ecosystem and integrates nicely with other Azure services such as Azure SQL Data Warehouse.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - IaaS Web Application for GDPR

11/7/2018 • 15 minutes to read • [Edit Online](#)

Overview

The General Data Protection Regulation (GDPR) contains many requirements about collecting, storing, and using personal information, including how organizations identify and secure personal data, accommodate transparency requirements, detect and report personal data breaches, and train privacy personnel and other employees. The GDPR gives individuals greater control over their personal data and imposes many new obligations on organizations that collect, handle, or analyze personal data. The GDPR imposes new rules on organizations that offer goods and services to people in the European Union (EU), or that collect and analyze data tied to EU residents. The GDPR applies no matter where an organization is located.

Microsoft designed Azure with industry-leading security measures and privacy policies to safeguard data in the cloud, including the categories of personal data identified by the GDPR. Microsoft's [contractual terms](#) commit Microsoft to the requirements of processors.

This Azure Security and Compliance Blueprint provides guidance to deploy an infrastructure as a service (IaaS) environment suitable for a simple Internet-facing web application. This solution demonstrates ways in which customers can meet specific security and compliance requirements of the GDPR and serves as a foundation for customers to build and configure their own IaaS web application solutions in Azure. Customers can utilize this reference architecture and follow Microsoft's [four-step process](#) in their journey to GDPR compliance:

1. Discover: Identify which personal data exists and where it resides.
2. Manage: Govern how personal data is used and accessed.
3. Protect: Establish security controls to prevent, detect, and respond to vulnerabilities and data breaches.
4. Report: Keep required documentation and manage data requests and breach notifications.

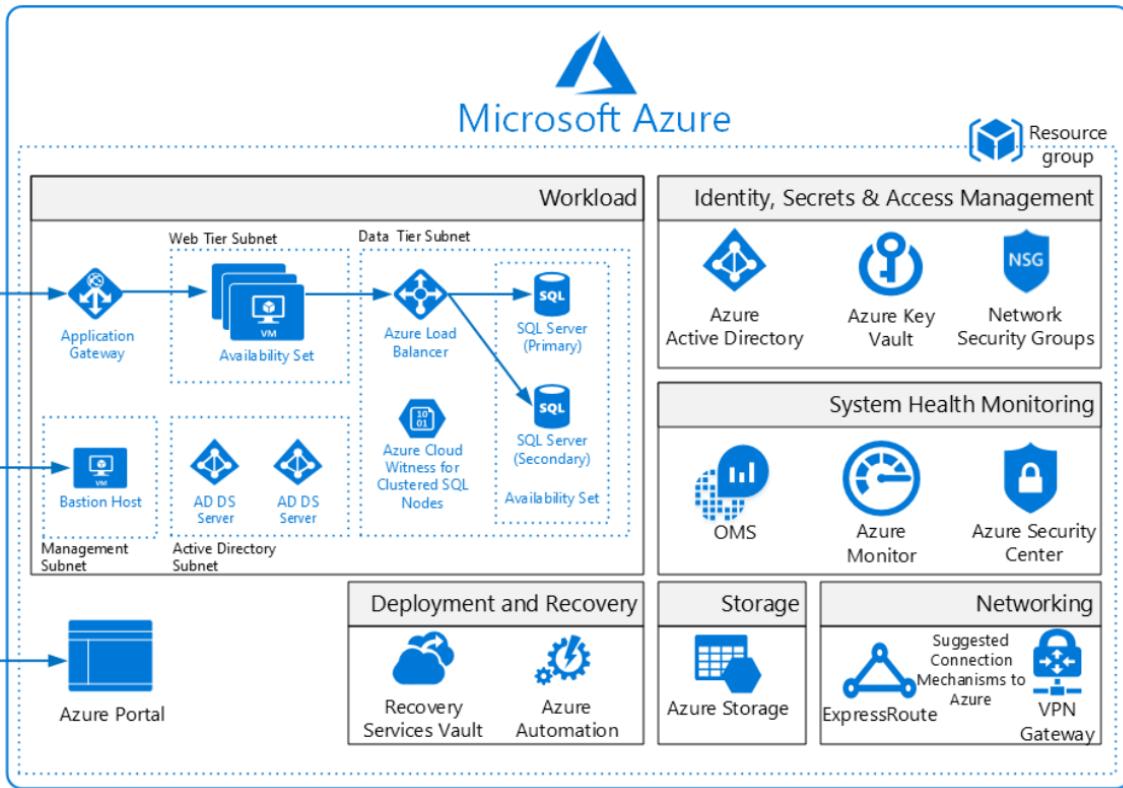
This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements and should not be used as-is in a production environment. Please note the following:

- The architecture provides a baseline to help customers deploy workloads to Azure in a GDPR-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution deploys a reference architecture for an IaaS web application with a SQL Server backend. The architecture includes a web tier, data tier, Active Directory infrastructure, Application Gateway, and Load Balancer. Virtual machines deployed to the web and data tiers are configured in an availability set, and SQL Server instances are configured in an AlwaysOn availability group for high availability. Virtual machines are domain-joined, and Active Directory group policies are used to enforce security and compliance configurations at the operating system level. A management bastion host provides a secure connection for administrators to access deployed resources.

Azure recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

- Azure Virtual Machines
 - (1) management/bastion (Windows Server 2016 Datacenter)
 - (2) Active Directory domain controller (Windows Server 2016 Datacenter)
 - (2) SQL Server cluster node (SQL Server 2017 on Windows Server 2016)
 - (2) Web/IIS (Windows Server 2016 Datacenter)
- Availability Sets
 - (1) Active Directory domain controllers
 - (1) SQL cluster nodes
 - (1) Web/IIS
- Azure Virtual Network
 - (1) /16 virtual networks
 - (5) /24 subnets
 - DNS settings are set to both domain controllers
- Azure Load Balancer
- Azure Application Gateway
 - (1) WAF Application Gateway enabled
 - firewall mode: prevention
 - rule set: OWASP 3.0
 - listener: port 443
- Azure Storage
 - (7) Geo-redundant storage accounts
- Cloud Witness
- Recovery Services vault
- Azure Key Vault
- Azure Active Directory (AAD)

- Azure Resource Manager
- Log Analytics
- Azure Security Center

Deployment architecture

The following section details the deployment and implementation elements.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group (NSG).

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Log Analytics extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: This solution deploys resources in an architecture with a separate web subnet, database subnet, Active Directory subnet, and management subnet inside of a virtual network. Subnets are logically separated by network security group rules applied to the individual subnets to restrict traffic between subnets to only that necessary for system and management functionality.

See the configuration for [network security groups](#) deployed with this solution. Organizations can configure Network Security groups by editing the file above using [this documentation](#) as a guide.

Each of the subnets has a dedicated network security group (NSG):

- 1 NSG for Application Gateway (LBNSG)
- 1 NSG for bastion host (MGTNSG)
- 1 NSG for primary and backup domain controllers (ADNSG)
- 1 NSG for SQL Servers and Cloud Witness (SQLNSG)
- 1 NSG for web tier (WEBNSG)

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. Additionally, all transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through multiple measures, including encryption and database auditing.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard personal data in support of organizational security commitments and compliance requirements defined by the GDPR.

Azure Disk Encryption: Azure Disk Encryption is used to encrypted Windows IaaS virtual machine disks. [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for OS and data disks.

The solution is integrated with Azure Key Vault to help control and manage the disk-encryption keys.

SQL Server: The SQL Server instance uses the following database security measures:

- [AD Authentication and Authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. TDE provides assurance that stored personal data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted Columns](#) ensure that sensitive personal data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- The [Extended Properties](#) feature can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated personal data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database Dynamic Data Masking \(DDM\)](#) limits sensitive personal data exposure by masking the data to non-privileged users or applications. DDM can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps with the identification of personal data qualifying for GDPR protection, and for reducing access such that it does not exit the database via unauthorized access. **Note:** **Customers will need to adjust DDM settings to adhere to their database schema.**

Identity management

The following technologies provide capabilities to manage access to personal data in the Azure environment:

- [Azure Active Directory \(AAD\)](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in AAD, including users accessing the SQL Server.
- Authentication to the application is performed using AAD. For more information, see [Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses AAD to authenticate the application to SQL Server. For more information, see how to [protect sensitive data in SQL Database](#).
- [Azure Role-Based Access Control \(RBAC\)](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permissions for Azure resources, administrators can allow only certain actions for accessing personal data. Subscription access is limited to the subscription administrator.
- [AAD Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain resources. Administrators can use AAD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [AAD Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.
- A deployed IaaS Active Directory instance provides identity management at the OS-level for deployed IaaS virtual machines.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key

Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect personal data and access to such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules (HSMs). The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.
- The solution is integrated with Azure Key Vault to manage IaaS virtual machine disk-encryption keys and secrets.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Security alerts: [Azure Security Center](#) enables customers to monitor traffic, collect logs, and analyze data sources for threats. Additionally, Azure Security Center accesses existing configuration of Azure services to provide configuration and service recommendations to help improve security posture and protect personal data. Azure Security Center includes a [threat intelligence report](#) for each detected threat to assist incident response teams investigate and remediate threats.

Application Gateway: The architecture reduces the risk of security vulnerabilities using an Application Gateway with web application firewall (WAF), and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-End-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (WAF mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Business continuity

High availability: The solution deploys all virtual machines in an [Availability Set](#). Availability sets ensure that the virtual machines are distributed across multiple isolated hardware clusters to improve availability. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS VM without restoring the entire VM, enabling faster restore times.

Cloud Witness: [Cloud Witness](#) is a type of Failover Cluster quorum witness in Windows Server 2016 that leverages Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations, but it uses the standard publicly available Azure Blob Storage. This eliminates the extra maintenance overhead of VMs hosted in a public cloud.

Logging and auditing

Log Analytics provides extensive logging of system and user activity, as well as system health. The [Log Analytics](#) solution collects and analyzes data generated by resources in Azure and on-premises environments.

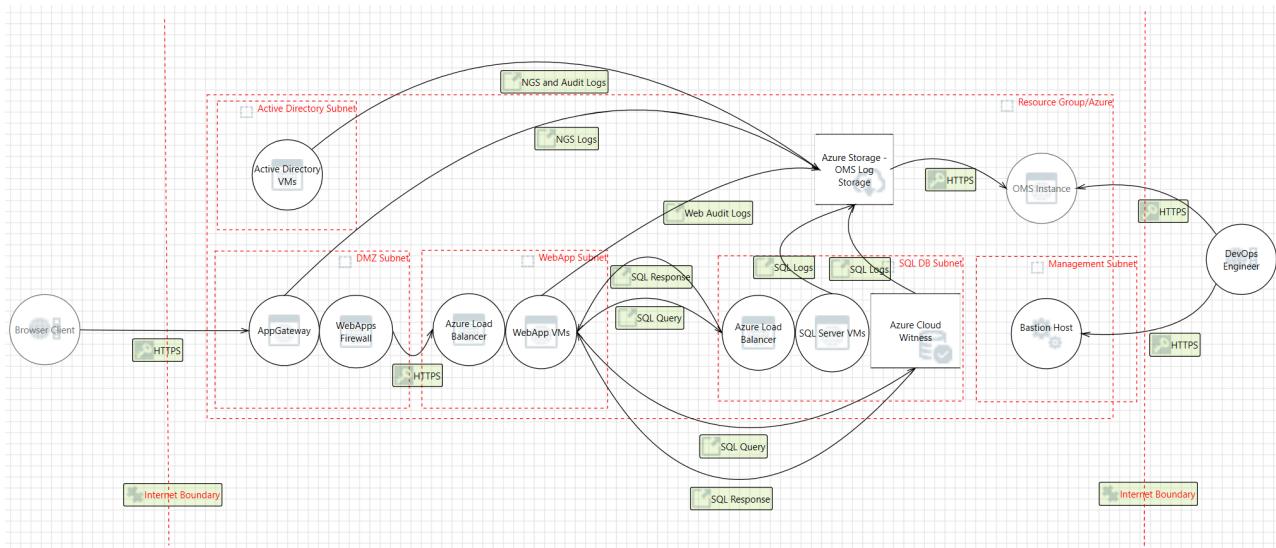
- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs.
- **Log archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

Additionally, the following monitoring solutions are included as a part of this architecture:

- [AD Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [Antimalware Assessment](#): The Antimalware solution reports on malware, threats, and protection status.
- [Azure Automation](#): The Azure Automation solution stores, runs, and manages runbooks.
- [Security and Audit](#): The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Update Management](#): The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Azure Activity Logs](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.
- [Change Tracking](#): The Change Tracking solution allows customers to easily identify changes in the environment.

Threat model

The data flow diagram (DFD) for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - GDPR Customer Responsibility Matrix](#) lists controller and processor responsibilities for all GDPR articles. Please note that for Azure services, a customer is usually the controller and Microsoft acts as the processor.

The [Azure Security and Compliance Blueprint - GDPR IaaS Web Application Implementation Matrix](#) provides information on which GDPR articles are addressed by the IaaS Web Application architecture, including detailed descriptions of how the implementation meets the requirements of each covered article.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this IaaS Web Application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.

- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - PaaS Web Application for GDPR

11/7/2018 • 14 minutes to read • [Edit Online](#)

Overview

The General Data Protection Regulation (GDPR) contains many requirements about collecting, storing, and using personal information, including how organizations identify and secure personal data, accommodate transparency requirements, detect and report personal data breaches, and train privacy personnel and other employees. The GDPR gives individuals greater control over their personal data and imposes many new obligations on organizations that collect, handle, or analyze personal data. The GDPR imposes new rules on organizations that offer goods and services to people in the European Union (EU), or that collect and analyze data tied to EU residents. The GDPR applies no matter where an organization is located.

Microsoft designed Azure with industry-leading security measures and privacy policies to safeguard data in the cloud, including the categories of personal data identified by the GDPR. Microsoft's [contractual terms](#) commit Microsoft to the requirements of processors.

This Azure Security and Compliance Blueprint provides guidance to deploy a platform as a service (PaaS) environment suitable for a simple Internet-facing web application. This solution demonstrates ways in which customers can meet the specific security and compliance requirements of the GDPR and serves as a foundation for customers to build and configure their own PaaS web application solutions in Azure. Customers can utilize this reference architecture and follow Microsoft's [four-step process](#) in their journey to GDPR compliance:

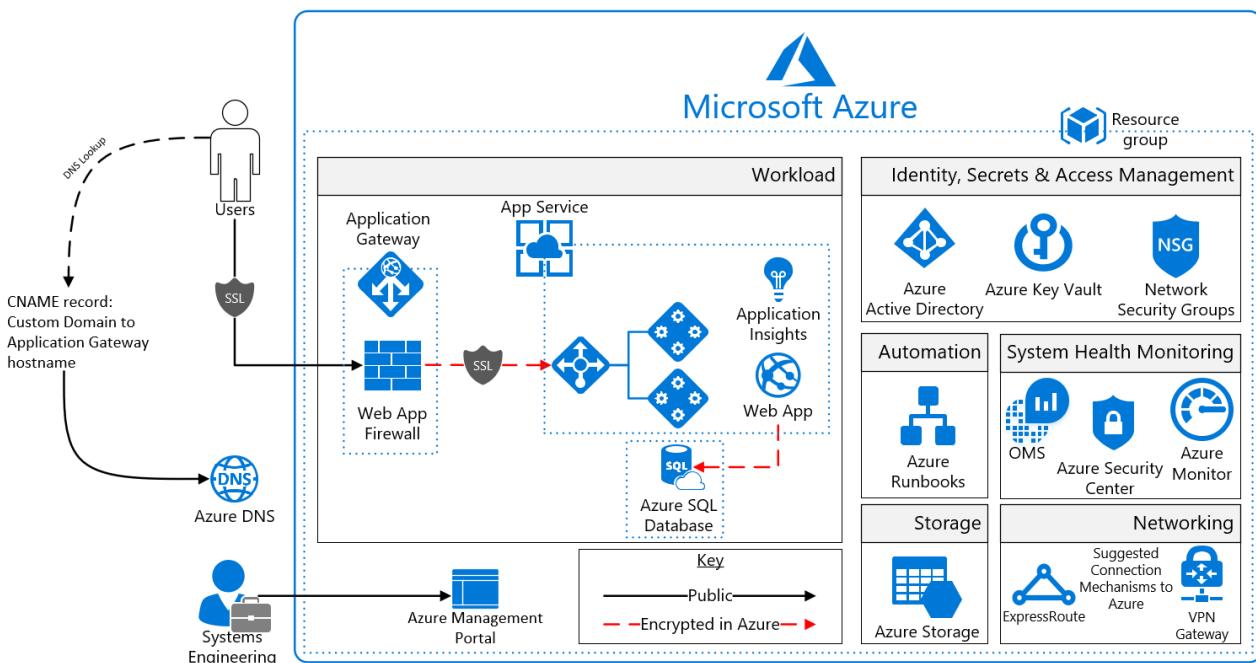
1. Discover: Identify which personal data exists and where it resides.
2. Manage: Govern how personal data is used and accessed.
3. Protect: Establish security controls to prevent, detect, and respond to vulnerabilities and data breaches.
4. Report: Keep required documentation and manage data requests and breach notifications.

This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements and should not be used as-is in a production environment. Please note the following:

- The architecture provides a baseline to help customers deploy workloads to Azure in a GDPR-compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture for a PaaS web application with an Azure SQL Database backend. The web application is hosted in an isolated Azure App Service Environment, which is a private, dedicated environment in an Azure datacenter. The environment load balances traffic for the web application across VMs managed by Azure. This architecture also includes network security groups, an Application Gateway, Azure DNS, and Load Balancer. Furthermore, Azure Monitor provides real-time analytics of system health. **Azure recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

- Azure Active Directory (AAD)
- Azure Key Vault
- Azure SQL Database
- Application Gateway
 - (1) WAF Application Gateway enabled
 - firewall mode: Prevention
 - rule set: OWASP 3.0
 - listener: port 443
- Azure virtual network
- network security groups
- Azure DNS
- Azure Storage
- Azure Monitor
- Application Insights
- Azure Security Center
- App Service Environment v2
- Azure Load Balancer
- Azure Web App
- Azure Resource Manager

Deployment architecture

The following section details the deployment and implementation elements.

Azure Resource Manager: [Azure Resource Manager](#) enables customers to work with the resources in the solution as a group. Customers can deploy, update, or delete all the resources for the solution in a single, coordinated operation. Customers use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help customers manage their resources after deployment.

App Service Environment v2: The [Azure App Service Environment](#) is an App Service feature that provides a fully isolated and dedicated environment for securely running App Service applications at a high scale.

ASEs are isolated to only run a single customer's applications and are always deployed into a virtual network. Customers have fine-grained control over both inbound and outbound application network traffic, and applications can establish high-speed secure connections over virtual networks to on-premises corporate resources.

Use of ASEs for this architecture are allowed for the following controls/configurations:

- Host inside a secured Azure Virtual Network and network security rules
- ASE configured with self-signed ILB certificate for HTTPS communication
- [Internal Load Balancing mode](#) (mode 3)
- Disable [TLS 1.0](#)
- Change [TLS Cipher](#)
- Control [inbound traffic N/W ports](#)
- [WAF – Restrict Data](#)
- Allow [Azure SQL Database traffic](#)

Azure Web App: [Azure Web Apps](#) enables customers to build and host web applications in the programming language of their choice without managing infrastructure. It offers auto-scaling and high availability, supports both Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repo.

Virtual Network

The architecture defines a private VNet with an address space of 10.200.0.0/16.

Network security groups: [NSGs](#) contain Access Control Lists (ACLs) that allow or deny traffic within a VNet. NSGs can be used to secure traffic at a subnet or individual VM level. The following NSGs exist:

- 1 NSG for Application Gateway
- 1 NSG for App Service Environment
- 1 NSG for Azure SQL Database

Each of the NSGs have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [NSG's diagnostics](#)

Subnets: Each subnet is associated with its corresponding NSG.

Azure DNS: The Domain Name System, or DNS, is responsible for translating (or resolving) a website or service name to its IP address. [Azure DNS](#) is a hosting service for DNS domains that provides name resolution using Azure infrastructure. By hosting domains in Azure, users can manage DNS records using the same credentials, APIs, tools, and billing as other Azure services. Azure DNS also supports private DNS domains.

Azure Load Balancer: [Azure Load Balancer](#) allows customers to scale their applications and create high availability for services. Load Balancer supports inbound as well as outbound scenarios, and provides low latency, high throughput, and scales up to millions of flows for all TCP and UDP applications.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard personal data in support of organizational security commitments and compliance requirements defined by the GDPR.

Azure Disk Encryption [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [AD authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [Transparent Data Encryption \(TDE\)](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. TDE provides assurance that stored personal data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted Columns](#) ensure that sensitive personal data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Extended properties](#) can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated personal data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database Dynamic Data Masking \(DDM\)](#) limits sensitive personal data exposure by masking the data to non-privileged users or applications. DDM can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps with the identification of personal data qualifying for GDPR protection, and for reducing access such that it does not exit the database via unauthorized access. **Note:** [**Customers will need to adjust DDM settings to adhere to their database schema.**](#)

Identity management

The following technologies provide capabilities to manage access to personal data in the Azure environment:

- [Azure Active Directory \(AAD\)](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in AAD, including users accessing the Azure SQL Database.
- Authentication to the application is performed using AAD. For more information, see [Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses AAD to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure Role-Based Access Control \(RBAC\)](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permissions for Azure resources, administrators can allow only certain actions for accessing personal data. Subscription access is limited to the subscription administrator.
- [AAD Privileged Identity Management \(PIM\)](#) enables customers to minimize the number of users who have access to certain information such as personal data. Administrators can use AAD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [AAD Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures

automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets Management The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect personal data and access to such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules (HSMs). The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Security Alerts: [Azure Security Center](#) enables customers to monitor traffic, collect logs, and analyze data sources for threats. Additionally, Azure Security Center accesses existing configuration of Azure services to provide configuration and service recommendations to help improve security posture and protect personal data. Azure Security Center includes a [threat intelligence report](#) for each detected threat to assist incident response teams investigate and remediate threats.

Application Gateway The architecture reduces the risk of security vulnerabilities using an Application Gateway with Web Application Firewall (WAF), and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-End-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web Application Firewall \(WAF mode\)](#)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Logging and auditing

Azure Monitor provides extensive logging of system and user activity, as well as system health. It collects and analyzes data generated by resources in Azure and on-premises environments.

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs.
- **Log archiving:** All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements. These logs connect to Azure Log Analytics for processing, storing, and dashboard reporting.

Additionally, the following monitoring solutions are included as a part of this architecture:

- **AD Assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.

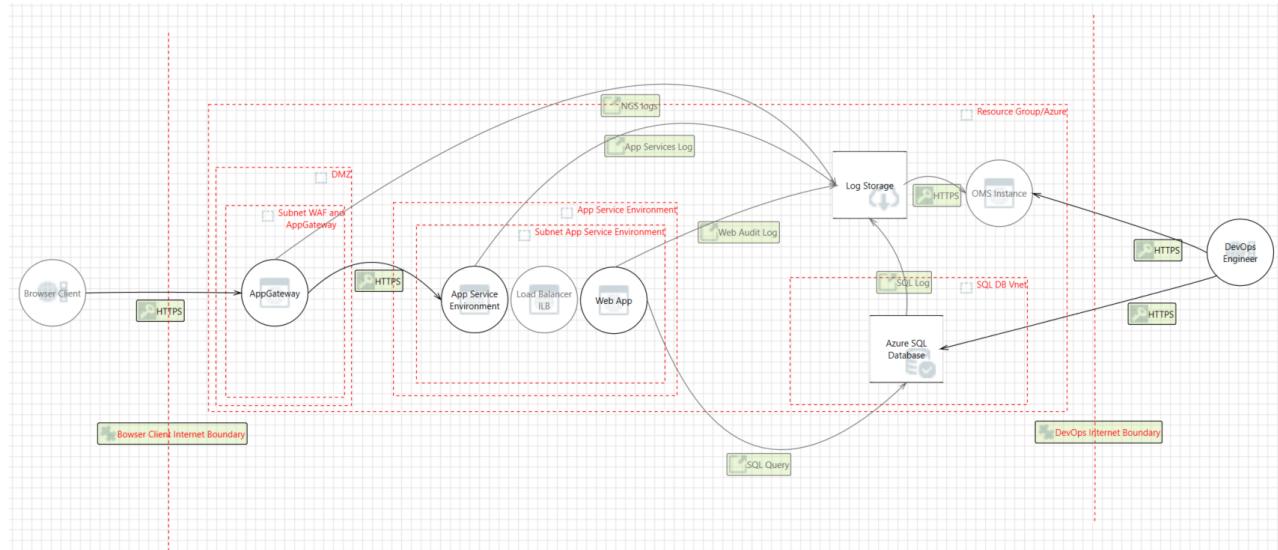
- **Antimalware Assessment:** The Antimalware solution reports on malware, threats, and protection status.
- **Azure Automation:** The Azure Automation solution stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Application Insights and Azure SQL Database.
- **Security and Audit:** The Security and Audit dashboard provides a high-level insight into the security state of resources by providing metrics on security domains, notable issues, detections, threat intelligence, and common security queries.
- **SQL Assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- **Update Management:** The Update Management solution allows customer management of operating system security updates, including a status of available updates and the process of installing required updates.
- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- **Azure Activity Logs:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.
- **Change Tracking:** The Change Tracking solution allows customers to easily identify changes in the environment.

Azure Monitor [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in customers' Azure resources.

Application Insights [Application Insights](#) is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Application Insights detects performance anomalies and customers can use it to monitor the live web application. It includes powerful analytics tools to help customers diagnose issues and to understand what users actually do with their app. It's designed to help customers continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – GDPR Customer Responsibility Matrix](#) lists controller and processor responsibilities for all GDPR articles. Please note that for Azure services, a customer is usually the controller and Microsoft acts as the processor.

The [Azure Security and Compliance Blueprint - GDPR PaaS Web Application Implementation Matrix](#) provides information on which GDPR articles are addressed by the PaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered article.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this PaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - HIPAA/HITRUST Health Data and AI

12/12/2018 • 13 minutes to read • [Edit Online](#)

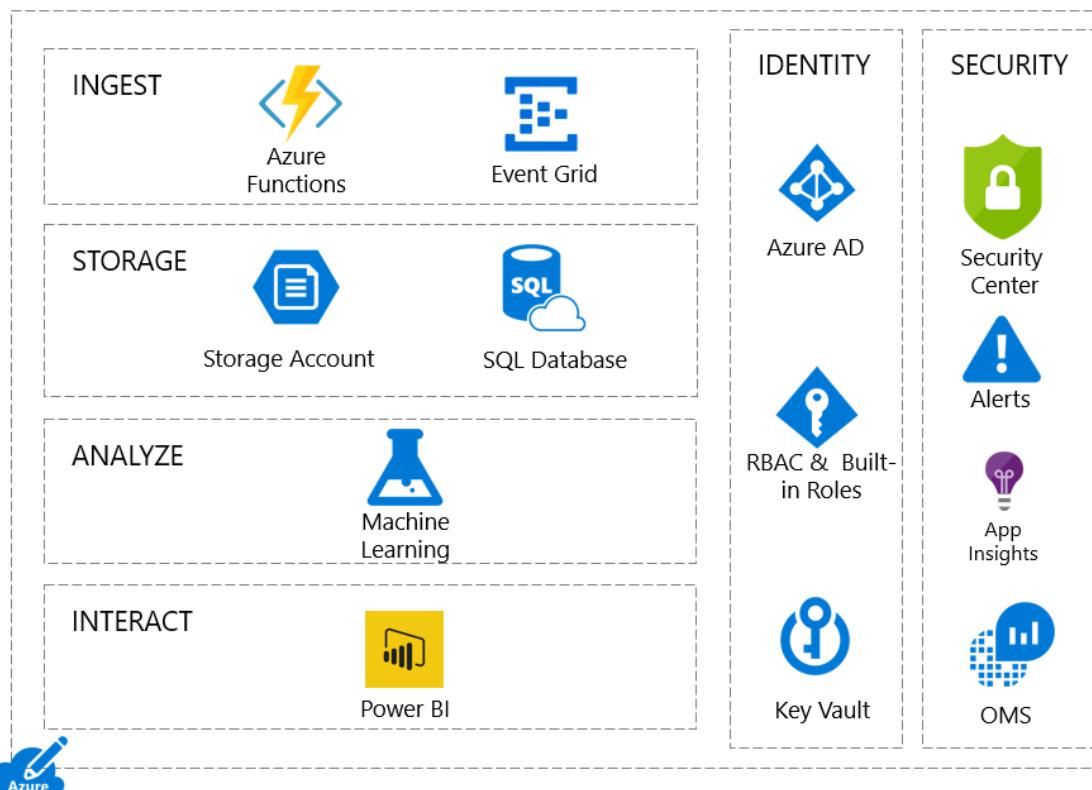
Overview

The Azure Security and Compliance Blueprint - HIPAA/HITRUST Health Data and AI offers a turn-key deployment of an Azure PaaS and IaaS solution to demonstrate how to ingest, store, analyze, interact, identity and Securely deploy solutions with health data while being able to meet industry compliance requirements. The blueprint helps accelerate cloud adoption and utilization for customers with data that is regulated.

The Azure Security and Compliance Blueprint - HIPAA/HITRUST Health Data and AI Blueprint provides tools and guidance to help deploy a secure, Health Insurance Portability and Accountability Act (HIPAA), and Health Information Trust Alliance (HITRUST) ready platform-as-a-service (PaaS) environment for ingesting, storing, analyzing, and interacting with personal and non-personal medical records in a secure, multi-tier cloud environment, deployed as an end-to-end solution.

IaaS solution will demonstrate how to migrate an on-premises SQL based solution to Azure, and to implement a Privileged Access Workstation (PAW) to securely manage cloud-based services and solutions. The IaaS SQL Server database adds potential experimentation data is imported into a SQL IaaS VM, and that VM uses MSI authenticated access to interact a SQL Azure PaaS service. Both these showcases a common reference architecture and is designed to simplify adoption of Microsoft Azure. This provided architecture illustrates a solution to meet the needs of organizations seeking a cloud-based approach to reducing the burden and cost of deployment.

Blueprint components



The solution is designed to consume a sample data set formatted using Fast Healthcare Interoperability Resources

(FHIR), a worldwide standard for exchanging healthcare information electronically, and store it in a secure manner. Customers can then use Azure Machine Learning Studio to take advantage of powerful business intelligence tools and analytics to review predictions made on the sample data. As an example of the kind of experiment Azure Machine Learning Studio can facilitate, the blueprint includes a sample dataset, scripts, and tools for predicting the length of a patient's stay in a hospital facility.

This blueprint is intended to serve as a modular foundation for customers to adjust to their specific requirements, developing new Azure Machine learning experiments to solve both clinical and operational use case scenarios. It is designed to be secure and compliant when deployed; however, customers are responsible for configuring roles correctly and implementing any modifications. Note the following:

- This blueprint provides a baseline to help customers use Microsoft Azure in a HITRUST, and HIPAA environment.
- Although the blueprint was designed to be aligned with HIPAA and HITRUST (through the Common Security Framework -- CSF), it should not be considered compliant until certified by an external auditor per HIPAA and HITRUST certification requirements.
- Customers are responsible for conducting appropriate security and compliance reviews of any solution built using this foundational architecture.

Deploying the automation

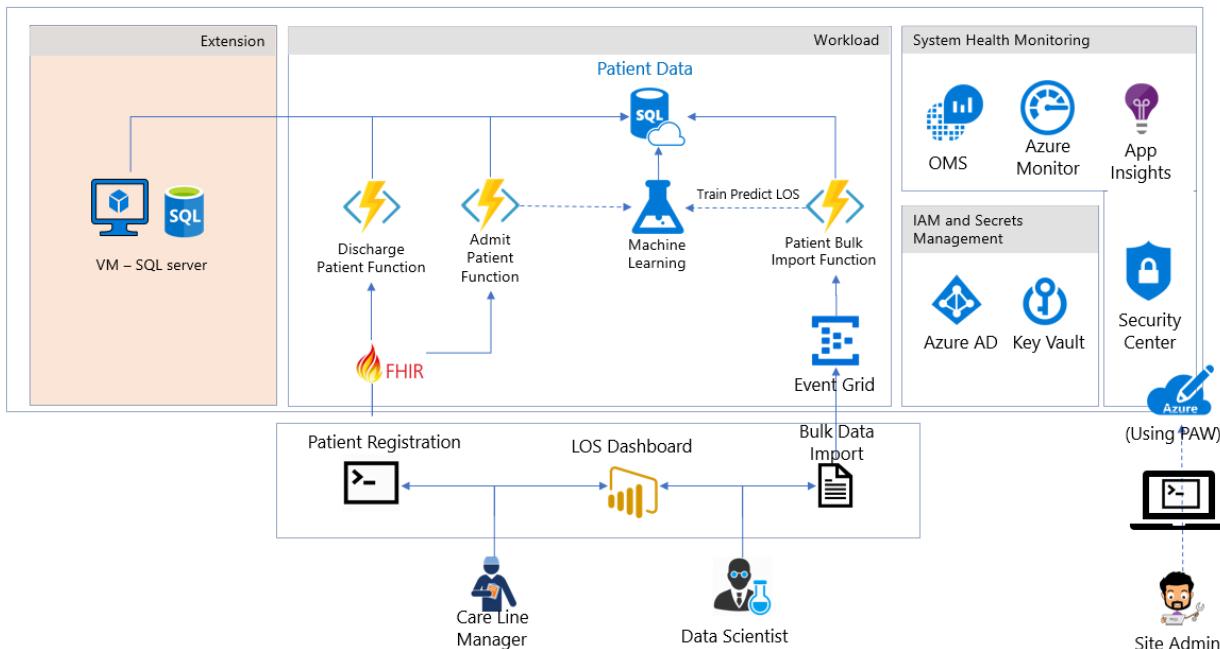
- To deploy the solution, follow the instructions provided in the [deployment guidance](#).
- For a quick overview of how this solution works, watch this [video](#) explaining and demonstrating its deployment.
- Frequently asked question can be found in the [FAQ](#) guidance.
- **Architectural diagram.** The diagram shows the reference architecture used for the blueprint and the example use case scenario.
- **IaaS Extension** This solution will demonstrate how to migrate an on-premise SQL based solution to Azure, and to implement a Privileged Access Workstation to securely manage cloud-based services and solutions.

Solution components

The foundational architecture is composed of the following components:

- **Threat model** A comprehensive threat model is provided in tm7 format for use with the [Microsoft Threat Modeling Tool](#), showing the components of the solution, the data flows between them, and the trust boundaries. The model can help customers understand the points of potential risk in the system infrastructure when developing Machine Learning Studio components or other modifications.
- **Customer implementation matrix** A Microsoft Excel workbook lists the relevant HITRUST requirements and explains how Microsoft and the customer are responsible for meeting each one.
- **Health review.** The solution was reviewed by Coalfire systems, Inc. The Health Compliance (HIPAA, and HITRUST) Review and guidance for implementation provides an auditor's review of the solution, and considerations for transforming the blueprint to a production-ready deployment.

Architectural diagram



Roles

The blueprint defines two roles for administrative users (operators), and three roles for users in hospital management and patient care. A sixth role is defined for an auditor to evaluate compliance with HIPAA and other regulations. Azure Role-based Access Control (RBAC) enables precisely focused access management for each user of the solution through built-in and custom roles. See [Get started with Role-Based Access Control in the Azure portal](#) and [Built-in roles for Azure role-based access control](#) for detailed information about RBAC, roles, and permissions.

Site Administrator

The site administrator is responsible for the customer's Azure subscription. They control the overall deployment, but have no access to patient records.

- Default role assignments: [Owner](#)
- Custom role assignments: N/A
- Scope: Subscription

Database Analyst

The database analyst administers the SQL Server instance and database. They have no access to patient records.

- Built-in role assignments: [SQL DB Contributor](#), [SQL Server Contributor](#)
- Custom role assignments: N/A
- Scope: ResourceGroup

Data Scientist

The data scientist operates the Azure Machine Learning Studio. They can import, export, and manage data, and run reports. The data scientist has access to patient data, but does not have administrative privileges.

- Built-in role assignments: [Storage Account Contributor](#)
- Custom role assignments: N/A
- Scope: ResourceGroup

Chief Medical Information Officer (CMIO)

The CMIO straddles the divide between informatics/technology and healthcare professionals in a healthcare organization. Their duties typically include using analytics to determine if resources are being allocated appropriately within the organization.

- Built-in role assignments: None

Care Line Manager

The care line manager is directly involved with the care of patients. This role requires monitoring the status of individual patients as well as ensuring that staff is available to meet the specific care requirements of their patients. The care line manager is responsible for adding and updating patient records.

- Built-in role assignments: None
- Custom role assignments: Has privilege to run HealthcareDemo.ps1 to do both Patient Admission, and Discharge.
- Scope: ResourceGroup

Auditor

The auditor evaluates the solution for compliance. They have no direct access to the network.

- Built-in role assignments: [Reader](#)
- Custom role assignments: N/A
- Scope: Subscription

Example Use case

The example use case included with this blueprint illustrates how the Blueprint can be used to enable machine learning and analytics on health data in the cloud. Contosoclinic is a small hospital located in the United States. The hospital network administrators want to use Azure Machine Learning Studio to better predict the length of a patient's stay at the time of admittance, in order to increase operational workload efficiency, and enhance the quality of care it can provide.

Predicting length of stay

The example use case scenario uses Azure Machine Learning Studio to predict a newly admitted patient's length of stay by comparing the medical details taken at patient intake to aggregated historical data from previous patients. The blueprint includes a large set of anonymized medical records to demonstrate the training and predictive capabilities of the solution. In a production deployment, customers would use their own records to train the solution for more accurate predictions reflecting the unique details of their environment, facilities, and patients.

Users and roles

Site Administrator -- Alex

Email: Alex_SiteAdmin

Alex's job is to evaluate technologies that can reduce the burden of managing an on-premises network and reduce costs for management. Alex has been evaluating Azure for some time but has struggled to configure the services that he needs to meet the HiTrust compliance requirements to store Patient Data in the cloud. Alex has selected the Azure Health AI to deploy a compliance-ready health solution, which has addressed the requirements to meet the customer requirements for HiTrust.

Data Scientist -- Debra

Email: Debra_DataScientist

Debra is in charge of using and creating models that analyze medical records to provide insights into patient care.

Debra uses SQL and the R statistical programming language to create her models.

Database Analyst -- Danny

Email: Danny_DBAnalyst

Danny is the main contact for anything regarding the Microsoft SQL Server that stores all the patient data for Contosoclinic. Danny is an experienced SQL Server administrator who has recently become familiar with Azure SQL Database.

Chief Medical Information Officer -- Caroline

Caroline is working with Chris the Care Line Manager, and Debra the Data Scientist to determine what factors impact patient length of stay. Caroline uses the predictions from the length-of-stay (LOS) solution to determine if resources are being allocated appropriately in the hospital network. For example, using the dashboard provided in this solution.

Care Line Manager -- Chris

Email: Chris_CareLineManager

As the individual directly responsible for managing patient admission, and discharges at Contosoclinic, Chris uses the predictions generated by the LOS solution to ensure that adequate staff are available to provide care to patients while they are staying in the facility.

Auditor -- Han

Email: Han_Auditor

Han is a certified auditor who has experience auditing for ISO, SOC, and HiTrust. Han was hired to review Contosoclinic's network. Han can review the Customer Responsibility Matrix provided with the solution to ensure that the blueprint and LOS solution can be used to store, process, and display sensitive personal data.

Design configuration

This section details the default configurations and security measures built into the Blueprint outlined to:

- **INGEST** data raw sources including FHIR data source
- **STORE** sensitive information
- **ANALYZE** and predict outcomes
- **INTERACT** with the results and predictions
- **IDENTITY** management of solution
- **SECURITY** enabled features

IDENTITY

Azure Active Directory and role-based access control (RBAC)

Authentication:

- [Azure Active Directory \(Azure AD\)](#) is the Microsoft's multi-tenant cloud-based directory and identity management service. All users for the solution were created in Azure Active Directory, including users accessing the SQL Database.
- Authentication to the application is performed using Azure AD. For more information, see [Integrating applications with Azure Active Directory](#).
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting your organization's identities, configures automated responses to detected suspicious actions related to your organization's

identities, and investigates suspicious incidents and takes appropriate action to resolve them.

- [Azure Role-based Access Control \(RBAC\)](#) enables precisely focused access management for Azure. Subscription access is limited to the subscription administrator, and Azure Key Vault access is limited to the site administrator. Strong passwords (12 characters minimum with at least one Upper/Lower letter, number, and special character) are required.
- Multi-factor authentication is supported when the -enableMFA switch is enabled during deployment.
- Passwords expire after 60 days when the -enableADDomainPasswordPolicy switch is enabled during deployment.

Roles:

- The solution makes use of [built-in roles](#) to manage access to resources.
- All users are assigned specific built-in roles by default.

Azure Key Vault

- Data stored in Key Vault includes:
 - Application insight key
 - Patient Data Storage Access key
 - Patient connection string
 - Patient data table name
 - Azure ML Web Service Endpoint
 - Azure ML Service API Key
- Advanced access policies are configured on a need basis
- Key Vault access policies are defined with minimum required permissions to keys and secrets
- All keys and secrets in Key Vault have expiration dates
- All keys in Key Vault are protected by HSM [Key Type = HSM Protected 2048-bit RSA Key]
- All users/identities are granted minimum required permissions using Role Based Access Control (RBAC)
- Applications do not share a Key Vault unless they trust each other and they need access to the same secrets at runtime
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required

INGEST

Azure Functions

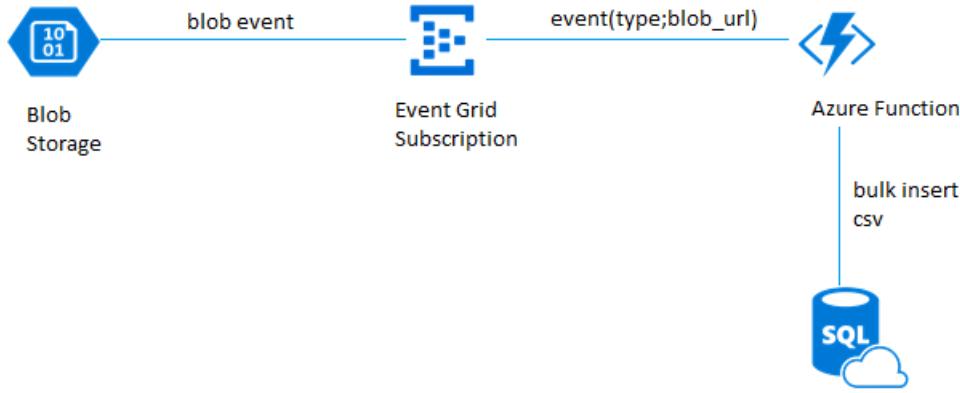
The solution was designed to use [Azure Functions](#) to process the sample length of stay data used in the analytics demo. Three capabilities in the functions have been created.

1. Bulk import of customer data phi data

When using the demo script. .\HealthcareDemo.ps1 with the **BulkPatientAdmission** switch as outlined in **Deploying and running the demo** it executes the following processing pipeline:

1. **Azure Blob Storage** - Patient data .csv file sample uploaded to storage
2. **Event Grid** - Event Publishes data to Azure Function (Bulk import - blob event)
3. **Azure Function** - Performs the processing and stores the data into SQL Storage using the secure function - event(type; blob url)
4. **SQL DB** - The database store for Patient Data using tags for classification, and the ML process is kicked off to do the training experiment.

- **Data Flow**



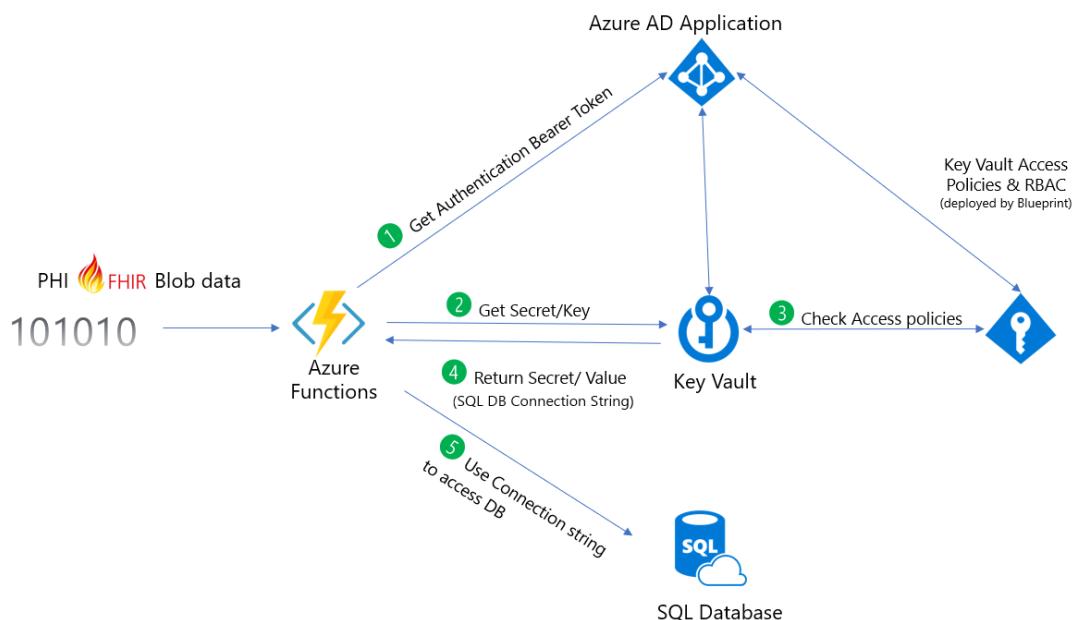
Additionally the azure function was designed to read and protect designated sensitive data in the sample data set using the following tags:

- dataProfile => "ePHI"
- owner => <Site Admin UPN>
- environment => "Pilot"
- department => "Global Ecosystem" The tagging was applied to the sample data set where patient 'names' was identified as clear text.

2. Admission of new patients

When using the demo script .\HealthcareDemo.ps1 with the **BulkPatientadmission** switch as outlined in **Deploying and running the demo** it executes the following processing pipeline:

Using a secure function admit patients SQL



1. Azure Function triggered and the function requests for a [bearer token](#) from Azure Active directory.

2. Key Vault requested for a secret that is associated to the requested token.

**3. Azure Roles validate the request, and authorize access request to the Key Vault.

4. Key Vault returns the secret, in this case the SQL DB Connection string.

5. Azure Function uses the connection string to securely connect to SQL Database and continues further

processing to store ePHI data.

To achieve the storage of the data, a common API schema was implemented following Fast Healthcare Interoperability Resources (FHIR, pronounced fire). The function was provided the following FHIR exchange elements:

- [Patient schema](#) covers the "who" information about a patient.
- [Observation schema](#) covers the central element in healthcare, used to support diagnosis, monitor progress, determine baselines and patterns and even capture demographic characteristics.
- [Encounter schema](#) covers the types of encounters such as ambulatory, emergency, home health, inpatient, and virtual encounters.
- [Condition schema](#) covers detailed information about a condition, problem, diagnosis, or other event, situation, issue, or clinical concept that has risen to a level of concern.

Event Grid

The solution supports Azure Event Grid, a single service for managing routing of all events from any source to any destination, providing:

- [Security and authentication](#)
- [Role-based access control](#) for various management operations such as listing event subscriptions, creating new ones, and generating keys
- Auditing

STORE

SQL Database and Server

- [Transparent Data Encryption \(TDE\)](#) provides real-time encryption and decryption of data stored in the Azure SQL Database, using a key stored in Azure Key Vault.
- [SQL Vulnerability Assessment](#) is an easy to configure tool that can discover, track, and remediate potential database vulnerabilities.
- [SQL Database Threat Detection](#) enabled.
- [SQL Database Auditing](#) enabled.
- [SQL Database metrics and diagnostic logging](#) enabled.
- [Server- and database-level firewall rules](#) have been tightened.
- [Always Encrypted columns](#) are used to protect patient first, middle, and last names. Additionally, the database column encryption also uses Azure Active Directory (AD) to authenticate the application to Azure SQL Database.
- Access to SQL Database and SQL Server is configured according to the principle of least privilege.
- Only required IP addresses are allowed access through the SQL firewall.

Storage accounts

- [Data in motion is transferred using TLS/SSL only.](#)
- Anonymous access is not allowed for containers.
- Alert rules are configured for tracking anonymous activity.
- HTTPS is required for accessing storage account resources.

- Authentication request data is logged and monitored.
- Data in Blob storage is encrypted at rest.

ANALYZE

Machine Learning

- [Logging is enabled](#) for Machine Learning Studio web services.
- Using [Machine Learning Studio](#) requires the development of experiments that provide the ability to predict to a solution set.

SECURITY

Azure Security Center

- [Azure Security Center](#) provides a centralized view of the security state of all your Azure resources. At a glance, you can verify that the appropriate security controls are in place and configured correctly, and you can quickly identify any resources that require attention.
- [Azure Advisor](#) is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, high availability, and security of your Azure resources.

Application Insights

- [Application Insights](#) is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Use it to monitor your live web application. It detects performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability.

Azure Alerts

- [Alerts offer a method of monitoring Azure services and allow you to configure conditions over data. Alerts also provide notifications when an alert condition matches the monitoring data.

Log Analytics

[Log Analytics](#) is a collection of management services.

- Workspace is enabled for Security Center
- Workspace is enabled for Workload Monitoring
- Workload Monitoring is enabled for:
 - Identity and Access
 - Security and Audit
 - Azure SQL DB Analytics
 - [Azure WebApp Analytics](#) Solution
 - Key Vault Analytics
 - Change Tracking
- [Application Insights Connector \(Preview\)](#) is enabled
- [Activity log analytics](#) is enabled

Azure Security and Compliance Blueprint - Data Analytics for NIST SP 800-171

12/6/2018 • 15 minutes to read • [Edit Online](#)

Overview

NIST Special Publication 800-171 provides guidelines for protecting the controlled unclassified information (CUI) that resides in nonfederal information systems and organizations. NIST SP 800-171 establishes 14 families of security requirements for protecting the confidentiality of CUI.

This Azure Security and Compliance Blueprint provides guidance to help customers deploy a data analytics architecture in Azure that implements a subset of NIST SP 800-171 controls. This solution demonstrates ways in which customers can meet specific security and compliance requirements. It also serves as a foundation for customers to build and configure their own data analytics solutions in Azure.

This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements. They shouldn't be used as-is in a production environment. Deploying this architecture without modification is insufficient to completely meet the requirements of NIST SP 800-171. Customers are responsible for conducting appropriate security and compliance assessments of any solution built that uses this architecture. Requirements might vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides an analytics platform upon which customers can build their own analytics tools. The reference architecture outlines a generic use case. Customers can use it to input data through bulk data imports by the SQL/Data administrator. They also can use it to input data through operational data updates via an operational user. Both workstreams incorporate Azure Functions for importing data into Azure SQL Database. Azure Functions must be configured by the customer through the Azure portal to handle the import tasks unique to the customer's analytics requirements.

Azure offers a variety of reporting and analytics services for the customer. This solution uses Azure Machine Learning services and SQL Database to rapidly browse through data and deliver quicker results through smarter modeling of data. Machine Learning is intended to increase query speeds by discovering new relationships between datasets. Initially, data is trained through several statistical functions. Afterward, up to seven additional query pools can be synchronized with the same tabular models to spread query workload and reduce response times. The customer server brings the total of query pools to eight.

For enhanced analytics and reporting, SQL Database can be configured with column store indexes. Machine Learning and SQL Database can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, we recommend the use of a trusted certificate authority for enhanced security.

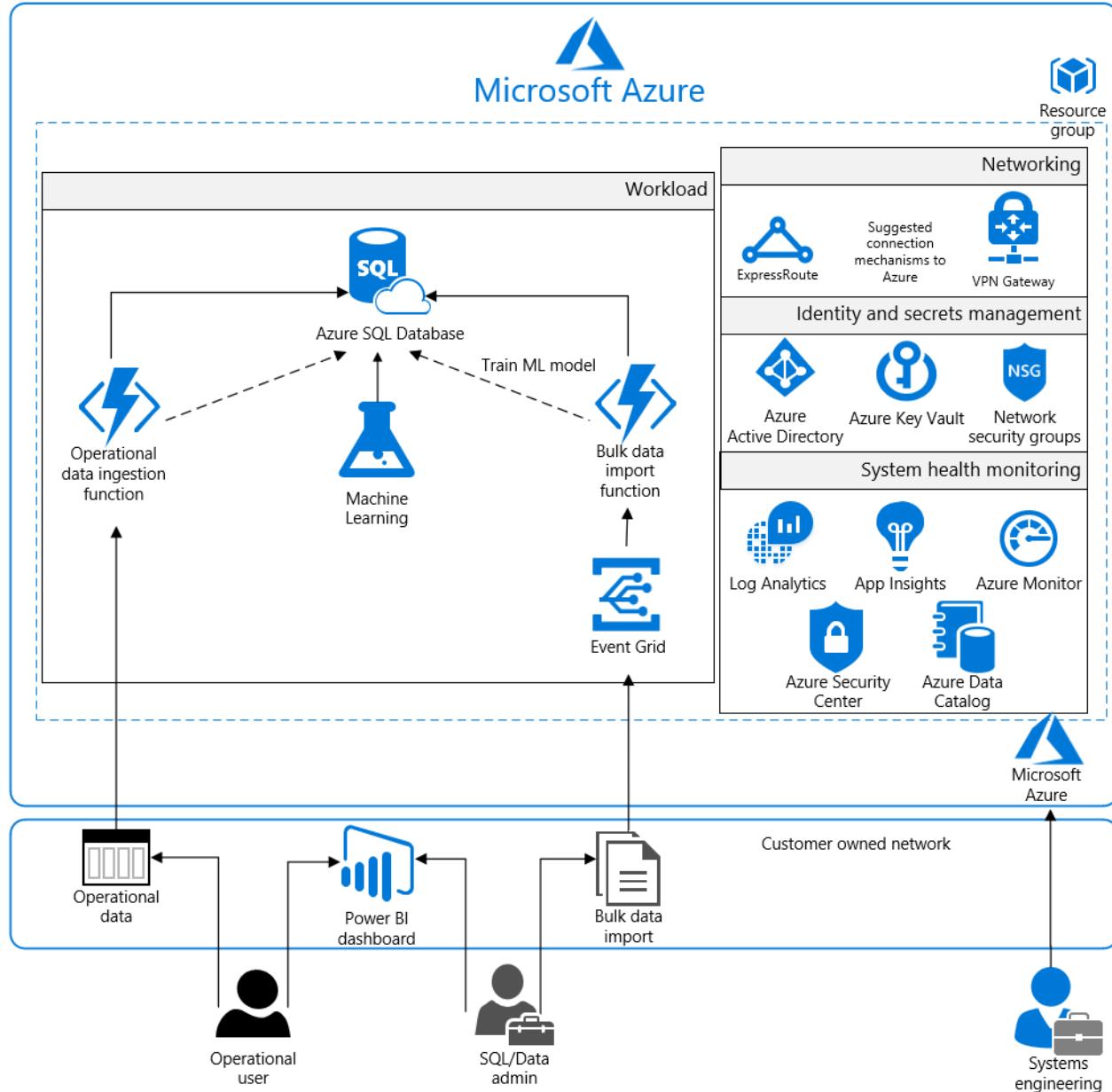
After data is uploaded to the SQL database and trained by Machine Learning, it's digested by the operational user and SQL/Data administrator with Power BI. Power BI displays data intuitively and pulls together information across multiple datasets to draw greater insight. Power BI has a high degree of adaptability and easily integrates with SQL Database. Customers can configure it to handle a wide array of scenarios that are required by their business needs.

The entire solution is built upon Azure Storage, which customers configure from the Azure portal. Storage encrypts all data with Storage Service Encryption to maintain confidentiality of data at rest. Geo-redundant storage ensures

that an adverse event at the customer's primary data center doesn't result in a loss of data. A second copy is stored in a separate location hundreds of miles away.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory (Azure AD) role-based access control (RBAC) is used to control access to deployed resources. These resources include customer keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs. System health is displayed in a single dashboard that's easy to use.

SQL Database is commonly managed through SQL Server Management Studio. It runs from a local machine configured to access the SQL database via a secure VPN or an Azure ExpressRoute connection. *We recommend that you configure a VPN or ExpressRoute connection for management and data import into the resource group.*



This solution uses the following Azure services. For more information, see the [deployment architecture](#) section.

- Application Insights
- Azure Active Directory
- Azure Data Catalog
- Azure Disk Encryption
- Azure Event Grid
- Azure Functions

- Azure Key Vault
- Azure Log Analytics
- Azure Machine Learning
- Azure Monitor
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Virtual Network
 - (1) /16 Network
 - (2) /24 Networks
 - (2) Network security groups
- Power BI Dashboard

Deployment architecture

The following section details the deployment and implementation elements.

Azure Event Grid: With [Event Grid](#), customers can easily build applications with event-based architectures. Users select the Azure resource they want to subscribe to. Then they give the event handler or webhook an endpoint to send the event to. Customers can secure webhook endpoints by adding query parameters to the webhook URL when they create an event subscription. Event Grid supports only HTTPS webhook endpoints. With Event Grid, customers can control the level of access given to different users to do various management operations. Users can list event subscriptions, create new ones, and generate keys. Event Grid uses Azure RBAC.

Azure Functions: [Azure Functions](#) is a serverless compute service that runs code on-demand. You don't have to explicitly provision or manage infrastructure. Use Azure Functions to run a script or piece of code in response to a variety of events.

Azure Machine Learning service: [Machine Learning](#) is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends.

Azure Data Catalog: [Data Catalog](#) makes data sources easy to discover and understand by the users who manage the data. Common data sources can be registered, tagged, and searched for data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog. A reference to the data source location is included. The metadata is indexed to make each data source easy to discover via search. Indexing also makes it understandable to the users who discover it.

Virtual network

This reference architecture defines a private virtual network with an address space of 10.0.0.0/16.

Network security groups: [Network security groups](#) (NSGs) contain access control lists that allow or deny traffic within a virtual network. NSGs can be used to secure traffic at a subnet or individual virtual machine level. The following NSGs exist:

- An NSG for Active Directory
- An NSG for the workload

Each of the NSGs has specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [NSG's diagnostics](#)

Subnets: Each subnet is associated with its corresponding NSG.

Data in transit

Azure encrypts all communications to and from Azure data centers by default. All transactions to Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet requirements for encrypted data at rest, all [Storage](#) uses [Storage Service Encryption](#). This feature helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by NIST SP 800-171.

Azure Disk Encryption: [Disk Encryption](#) uses the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [transparent data encryption](#). It performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data hasn't been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur. It provides security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted columns](#) ensure that sensitive data never appears as plain text inside the database system. After data encryption is enabled, only client applications or application servers with access to the keys can access plain-text data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to nonprivileged users or applications. It can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. Dynamic data masking helps to reduce access so that sensitive data doesn't exit the database via unauthorized access. *Customers are responsible for adjusting settings to adhere to their database schema.*

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure AD](#) is the Microsoft multitenant cloud-based directory and identity management service. All users for this solution are created in Azure AD and include the users who access SQL Database.
- Authentication to the application is performed by using Azure AD. For more information, see how to [integrate applications with Azure AD](#). The database column encryption also uses Azure AD to authenticate the application to SQL Database. For more information, see how to [protect sensitive data in SQL Database](#).
- [Azure RBAC](#) can be used by administrators to define fine-grained access permissions. With it, they can grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permissions for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) can be used by customers to minimize the number of users who have access to certain information, such as data. Administrators can use Azure AD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality also can be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities that affect an organization's identities. It configures automated responses to detected suspicious actions related to an organization's identities. It also

investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Key Vault capabilities help customers protect data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security-module-protected 2048-bit RSA key.
- All users and identities are granted minimum required permissions by using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Security Center also accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Security Center uses a variety of detection capabilities to alert customers of potential attacks that target their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Security Center has a set of [predefined security alerts](#) that are triggered when a threat or suspicious activity takes place. Customers can use [custom alert rules](#) to define new security alerts based on data that's already collected from their environment.

Security Center provides prioritized security alerts and incidents. Security Center makes it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat. Incident response teams can use the reports when they investigate and remediate threats.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Storage logs, Key Vault audit logs, and Azure Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. Users can configure the retention period, up to 730 days, to meet their specific requirements.

Log Analytics: Logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. After data is collected, it's organized into separate tables for each data type within Log Analytics workspaces. In this way, all data can be analyzed together, regardless of its original source. Security Center integrates with Log Analytics. Customers can use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- **Active Directory assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval. It provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **SQL assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval. It provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.

- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic distribution. It also reports on how many agents are unresponsive and the number of agents that submit operational data.
- **Activity Log Analytics:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

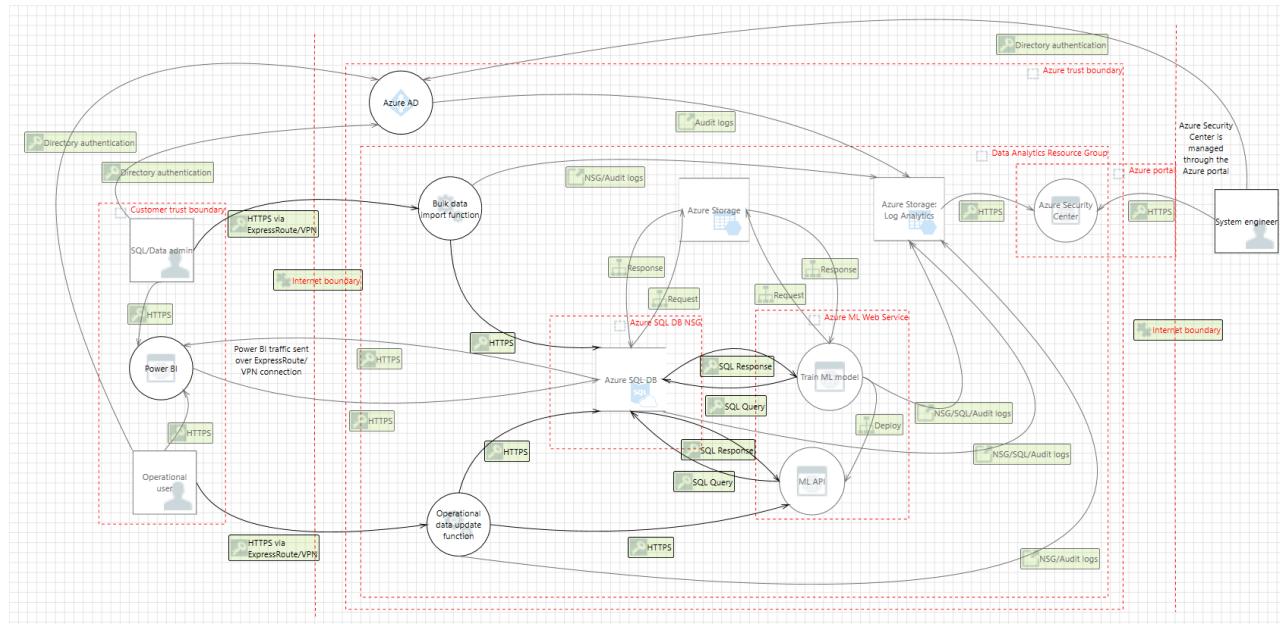
Azure Automation: [Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from SQL Database. Customers can use the Automation [Change Tracking](#) solution to easily identify changes in the environment.

Azure Monitor: [Monitor](#) helps users track performance, maintain security, and identify trends. Organizations can use it to audit, create alerts, and archive data. They also can track API calls in their Azure resources.

Application Insights: [Application Insights](#) is an extensible Application Performance Management service for web developers on multiple platforms. It detects performance anomalies and includes powerful analytics tools. The tools help diagnose issues and help customers understand what users do with the app. It's designed to help users continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found here. This model can help customers understand the points of potential risk in the system infrastructure when they make modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - NIST SP 800-171 Customer Responsibility Matrix](#) lists all security controls required by NIST SP 800-171. This matrix details whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - NIST SP 800-171 Data Analytics Control Implementation Matrix](#) provides information on which NIST SP 800-171 controls are addressed by the data analytics architecture. It includes detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) must be configured to securely establish a connection to the resources deployed as a part of this data analytics reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection takes place over the Internet. Customers can use the connection to securely "tunnel" information inside an encrypted link between their network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel traverses the Internet with a site-to-site VPN, Microsoft offers another even more secure connection option. ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. ExpressRoute connections connect directly to a customer's telecommunication provider. As a result, the data doesn't travel over the Internet and isn't exposed to it. These connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-transform-load process

[PolyBase](#) can load data into SQL Database without the need for a separate extract-transform-load or import tool. PolyBase allows access to data through T-SQL queries. The Microsoft business intelligence and analysis stack and third-party tools that are compatible with SQL Server can be used with PolyBase.

Azure AD setup

[Azure AD](#) is essential to managing the deployment and provisioning access to personnel who interact with the environment. On-premises Active Directory can be integrated with Azure AD in [four clicks](#). Customers also can tie the deployed Active Directory infrastructure (domain controllers) to Azure AD. To do this, make the deployed Active Directory infrastructure a subdomain of an Azure AD forest.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - Data Warehouse for NIST SP 800-171

10/18/2018 • 18 minutes to read • [Edit Online](#)

Overview

NIST Special Publication 800-171 provides guidelines for protecting the controlled unclassified information (CUI) that resides in nonfederal information systems and organizations. NIST SP 800-171 establishes 14 families of security requirements for protecting the confidentiality of CUI.

This Azure Security and Compliance Blueprint provides guidance to help customers deploy a data warehouse architecture in Azure that implements a subset of NIST SP 800-171 controls. This solution demonstrates ways in which customers can meet specific security and compliance requirements. It also serves as a foundation for customers to build and configure their own data warehouse solutions in Azure.

This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements. They shouldn't be used as-is in a production environment. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture. Requirements might vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture that implements a high-performance and secure cloud data warehouse. There are two separate data tiers in this architecture. One tier is where data is imported, stored, and staged within a clustered SQL environment. Another tier is for the SQL data warehouse. With this tier, the data is loaded by using an extract-transform-load (ETL) tool (for example, PolyBase T-SQL queries) for processing. After data is stored in SQL Data Warehouse, analytics can run at a massive scale.

Azure offers a variety of reporting and analytics services for the customer. This solution includes SQL Server Reporting Services for quick creation of reports from the SQL data warehouse. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, we recommend the use of a trusted certificate authority for enhanced security.

Azure SQL Data Warehouse stores data in relational tables with columnar storage. This format significantly reduces data storage costs while it improves query performance. Depending on usage requirements, SQL Data Warehouse compute resources can be scaled up or down or shut off completely if no active processes require compute resources.

A SQL Server load balancer manages SQL traffic to ensure high performance. All virtual machines (VMs) in this reference architecture deploy in an availability set with SQL Server instances configured in an Always On availability group. This configuration provides high-availability and disaster-recovery capabilities.

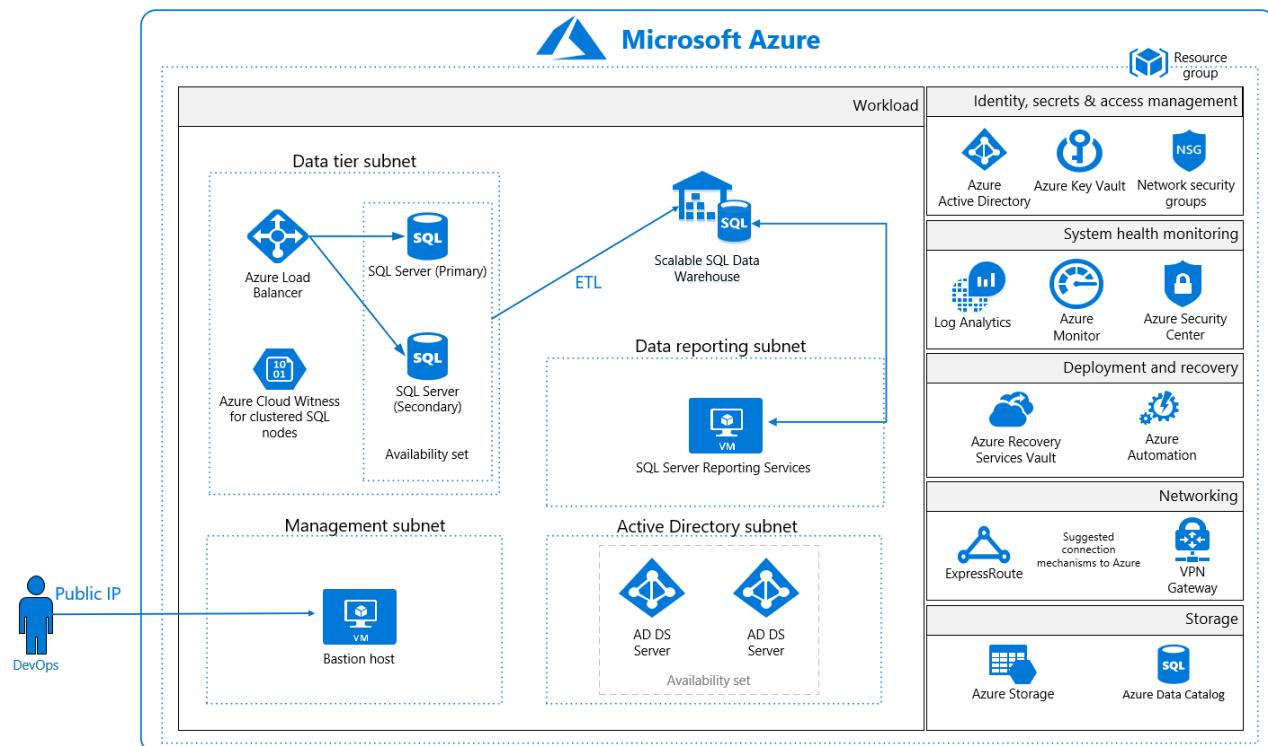
This data warehouse reference architecture also includes an Active Directory tier for management of resources within the architecture. The Active Directory subnet enables easy adoption under a larger Active Directory forest structure. This way, the environment can operate continuously, even when access to the larger forest is unavailable. All VMs are domain joined to the Active Directory tier. They use Active Directory group policies to enforce security and compliance configurations at the operating system level.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected data center

for resiliency. Geo-redundant storage ensures that data is replicated to a secondary data center hundreds of miles away and again stored as three copies within that data center. This arrangement prevents an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory (Azure AD) role-based access control (RBAC) is used to control access to deployed resources. These resources include customer keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs. System health is displayed in a single dashboard that's easy to use.

A VM serves as a management bastion host. It provides a secure connection for administrators to access deployed resources. The data loads into the staging area through this management bastion host. *We recommend that you configure a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.*



This solution uses the following Azure services. For more information, see the [deployment architecture](#) section.

- Availability sets
 - Active Directory domain controllers
 - SQL Server cluster nodes and witness
- Azure Active Directory
- Azure Data Catalog
- Azure Key Vault
- Azure Monitor
- Azure Security Center
- Azure Load Balancer
- Azure Storage
- Azure Log Analytics
- Azure Virtual Machines
 - (1) Bastion host
 - (2) Active Directory domain controller
 - (2) SQL Server cluster node

- (1) SQL Server witness
- Azure Virtual Network
 - (1) /16 network
 - (4) /24 networks
 - (4) Network security groups
- Recovery Services vault
- SQL Data Warehouse
- SQL Server Reporting Services

Deployment architecture

The following section details the deployment and implementation elements.

Azure SQL Data Warehouse: [SQL Data Warehouse](#) is an enterprise data warehouse that takes advantage of massively parallel processing to quickly run complex queries across petabytes of data. Users can use simple PolyBase T-SQL queries to import big data into the SQL data warehouse and use the power of massively parallel processing to run high-performance analytics.

SQL Server Reporting Services: [SQL Server Reporting Services](#) provides quick creation of reports with tables, charts, maps, gauges, matrixes, and more for SQL Data Warehouse.

Azure Data Catalog: [Data Catalog](#) makes data sources easy to discover and understand by the users who manage the data. Common data sources can be registered, tagged, and searched for data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog. A reference to the data source location is included. The metadata is indexed to make each data source easy to discover via search. Indexing also makes it understandable to the users who discover it.

Bastion host: The bastion host is the single point of entry that users can use to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by allowing only remote traffic from public IP addresses on a safe list. To permit remote desktop traffic, the source of the traffic must be defined in the network security group.

This solution creates a VM as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#).
- [Azure Diagnostics extension](#).
- [Azure Disk Encryption](#) using Key Vault.
- An [auto-shutdown policy](#) to reduce consumption of VM resources when not in use.
- [Windows Defender Credential Guard](#) is enabled so that credentials and other secrets run in a protected environment that's isolated from the running operating system.

Virtual network

This reference architecture defines a private virtual network with an address space of 10.0.0.0/16.

Network security groups: [Network security groups](#) (NSGs) contain access control lists that allow or deny traffic within a virtual network. NSGs can be used to secure traffic at a subnet or individual VM level. The following NSGs exist:

- An NSG for the data tier (SQL Server clusters, SQL Server witness, and SQL load balancer)
- An NSG for the management bastion host
- An NSG for Active Directory
- An NSG for SQL Server Reporting Services

Each of the NSGs has specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account.
- Log Analytics is connected to the [NSG's diagnostics](#).

Subnets: Each subnet is associated with its corresponding NSG.

Data at rest

The architecture protects data at rest through multiple measures. These measures include encryption and database auditing.

Azure Storage: To meet requirements for encrypted data at rest, all [Storage](#) uses [Storage Service Encryption](#). This feature helps protect and safeguard data in support of organizational security commitments and compliance requirements.

Azure Disk Encryption: [Disk Encryption](#) uses the BitLocker feature of Windows to provide volume encryption for operating system and data disks. The solution integrates with Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL Database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [transparent data encryption](#). It performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data hasn't been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur. It provides security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted columns](#) ensure that sensitive data never appears as plain text inside the database system. After data encryption is enabled, only client applications or app servers with access to the keys can access plain-text data.
- [Extended properties](#) can be used to discontinue the processing of data subjects. Users can add custom properties to database objects. They also can tag data as "Discontinued" to support application logic to prevent the processing of associated financial data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to nonprivileged users or applications. It can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. Dynamic data masking helps to reduce access so that sensitive data doesn't exit the database via unauthorized access. *Customers are responsible for adjusting settings to adhere to their database schema.*

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure AD](#) is the Microsoft multitenant cloud-based directory and identity management service. All users for this solution are created in Azure AD and include the users who access the SQL database.
 - Authentication to the application is performed by using Azure AD. For more information, see how to [integrate applications with Azure AD](#). The database column encryption also uses Azure AD to authenticate the application to SQL Database. For more information, see how to [protect sensitive data in SQL Database](#).
 - [Azure RBAC](#) can be used by administrators to define fine-grained access permissions. With it, they can grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted access for Azure resources, administrators can allow only certain actions for accessing resources and data.
- Subscription access is limited to the subscription administrator.

- [Azure Active Directory Privileged Identity Management](#) can be used by customers to minimize the number of users who have access to certain information, such as data. Administrators can use Azure AD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality also can be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities that affect an organization's identities. It configures automated responses to detected suspicious actions related to an organization's identities. It also investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Key Vault capabilities help customers protect data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security-module-protected 2048-bit RSA key.
- All users and identities are granted minimum required permissions by using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Patch management: Windows VMs deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch VMs when needed.

Malware protection: [Microsoft Antimalware](#) for VMs provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software. Customers can configure alerts that generate when known malicious or unwanted software attempts to install or run on protected VMs.

Azure Security Center: With [Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Security Center also accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Security Center uses a variety of detection capabilities to alert customers of potential attacks that target their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Security Center has a set of [predefined security alerts](#) that are triggered when a threat or suspicious activity takes place. Customers can use [custom alert rules](#) to define new security alerts based on data that's already collected from their environment.

Security Center provides prioritized security alerts and incidents. Security Center makes it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat. Incident response teams can use the reports when they investigate and remediate threats.

This reference architecture also uses the [vulnerability assessment](#) capability in Security Center. After it's configured, a partner agent (for example, Qualys) reports vulnerability data to the partner's management platform. In turn, the partner's management platform provides vulnerability and health monitoring data back to Security Center. Customers can use this information to quickly identify vulnerable VMs.

Business continuity

High availability: Server workloads are grouped in an [availability set](#) to help ensure high availability of VMs in Azure. At least one VM is available during a planned or unplanned maintenance event, which meets the 99.95% Azure SLA.

Recovery Services vault: The [Recovery Services vault](#) houses backup data and protects all configurations of VMs in this architecture. With a Recovery Services vault, customers can restore files and folders from an IaaS VM without restoring the entire VM. This process speeds up restore times.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Storage logs, Key Vault audit logs, and Azure Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. Users can configure the retention period, up to 730 days, to meet their specific requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. After data is collected, it's organized into separate tables for each data type within Log Analytics workspaces. In this way, all data can be analyzed together, regardless of its original source. Security Center integrates with Log Analytics. Customers can use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

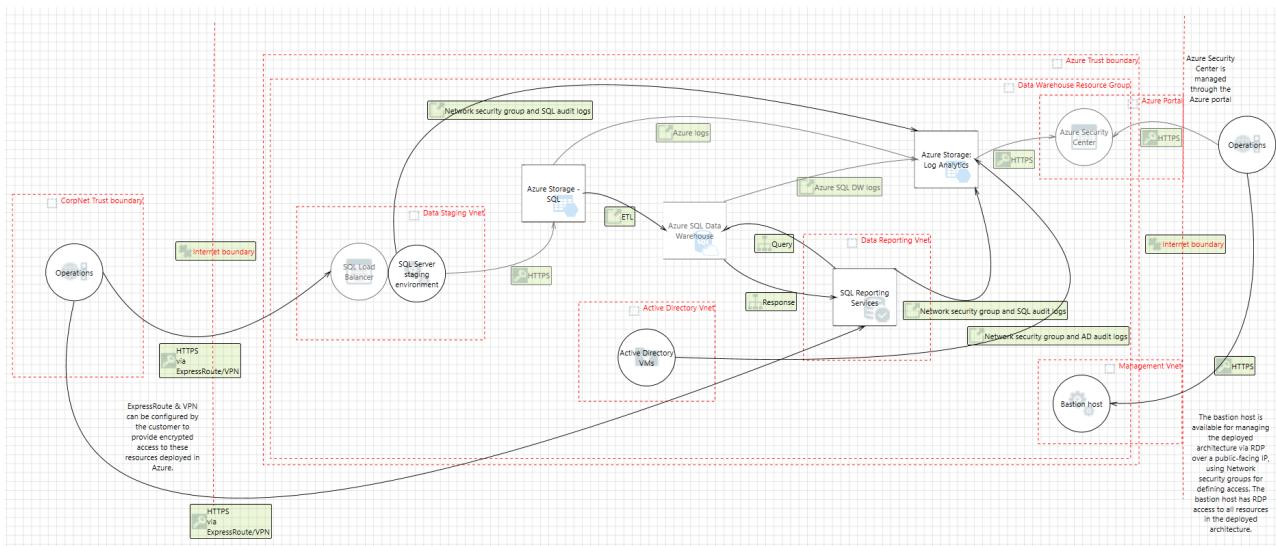
- **Active Directory assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval. It provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **SQL assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval. It provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic distribution. It also reports how many agents are unresponsive and the number of agents that submit operational data.
- **Activity Log Analytics:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from SQL Database. Customers can use the Automation [Change Tracking](#) solution to easily identify changes in the environment.

Azure Monitor: [Monitor](#) helps users track performance, maintain security, and identify trends. Organizations can use it to audit, create alerts, and archive data. They also can track API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found here. This model can help customers understand the points of potential risk in the system infrastructure when they make modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – NIST SP 800-171 Customer Responsibility Matrix](#) lists all security controls required by NIST SP 800-171. This matrix details whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - NIST SP 800-171 Data Warehouse Control Implementation Matrix](#) provides information on which NIST SP 800-171 controls are covered by the data warehouse architecture. It includes detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) must be configured to securely establish a connection to the resources deployed as a part of this data warehouse reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection takes place over the Internet. Customers can use this connection to securely “tunnel” information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel traverses the Internet with a site-to-site VPN, Microsoft offers another even more secure connection option. ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. ExpressRoute connections connect directly to the customer's telecommunication provider. As a result, the data doesn't travel over the Internet and isn't exposed to it. These connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-Transform-Load process

[PolyBase](#) can load data into SQL Data Warehouse without the need for a separate ETL or import tool. PolyBase allows access to data through T-SQL queries. The Microsoft business intelligence and analysis stack and third-party tools that are compatible with SQL Server can be used with PolyBase.

Azure AD setup

[Azure AD](#) is essential to managing the deployment and provisioning access to personnel interacting with the

environment. On-premises Active Directory can be integrated with Azure AD in [four clicks](#). Customers also can tie the deployed Active Directory infrastructure (domain controllers) to Azure AD. To do this, make the deployed Active Directory infrastructure a subdomain of an Azure AD forest.

Optional services

Azure offers a variety of services to assist with the storage and staging of formatted and unformatted data. The following services can be added to this reference architecture based on customer requirements:

- [Azure Data Factory](#) is a managed cloud service that's built for complex hybrid ETL, extract-load-transform, and data integration projects. Data Factory has capabilities to help trace and locate data. Visualization and monitoring tools identify when data arrived and where it came from. Customers can create and schedule data-driven workflows, called pipelines, that ingest data from disparate data stores. They can use the pipelines to ingest data from internal and external sources. Customers can then process and transform the data for output into data stores, such as SQL Data Warehouse.
- Customers can stage unstructured data in [Azure Data Lake Store](#) to capture data of any size, type, and ingestion speed in one place for operational and exploratory analytics. Azure Data Lake has capabilities that enable the extraction and conversion of data. Data Lake Store is compatible with most open source components in the Hadoop ecosystem. It also integrates nicely with other Azure services, such as SQL Data Warehouse.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - IaaS Web Application for NIST SP 800-171

10/18/2018 • 14 minutes to read • [Edit Online](#)

Overview

NIST Special Publication 800-171 provides guidelines for protecting the controlled unclassified information (CUI) that resides in nonfederal information systems and organizations. NIST SP 800-171 establishes 14 families of security requirements for protecting the confidentiality of CUI.

This Azure Security and Compliance Blueprint provides guidance to help customers deploy a web application architecture in Azure that implements a subset of NIST SP 800-171 controls. This solution demonstrates ways in which customers can meet specific security and compliance requirements. It also serves as a foundation for customers to build and configure their own web applications in Azure.

This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements. They shouldn't be used as-is in a production environment. Deploying this architecture without modification is insufficient to completely meet the requirements of NIST SP 800-171. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture. Requirements might vary based on the specifics of each customer's implementation.

Architecture diagram and components

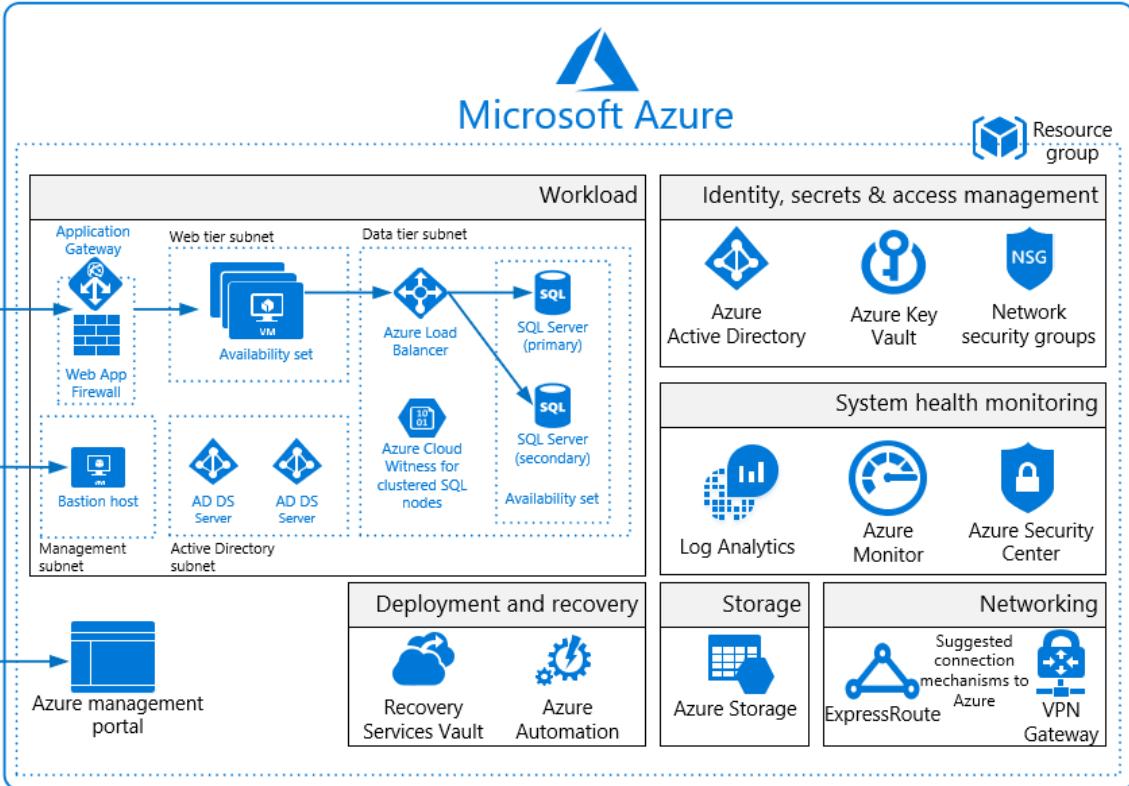
This Azure Security and Compliance Blueprint deploys a reference architecture for an IaaS web application with a SQL Server back end. The architecture includes a web tier, data tier, Active Directory infrastructure, Azure Application Gateway, and Azure Load Balancer. Virtual machines (VMs) deployed to the web and data tiers are configured in an availability set. SQL Server instances are configured in an Always On availability group for high availability. VMs are domain joined. Active Directory group policies enforce security and compliance configurations at the operating system level.

The entire solution is built upon Azure Storage, which customers configure from the Azure portal. Storage encrypts all data with Storage Service Encryption to maintain confidentiality of data at rest. Geo-redundant storage ensures that an adverse event at the customer's primary data center doesn't result in a loss of data. A second copy is stored in a separate location hundreds of miles away.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory (Azure AD) role-based access control (RBAC) is used to control access to deployed resources and keys in Azure Key Vault. System health is monitored through Azure Monitor. Customers configure both monitoring services to capture logs. System health is displayed in a single dashboard that's easy to use.

A management bastion host provides a secure connection for administrators to access deployed resources.

Microsoft recommends that you configure a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.



This solution uses the following Azure services. For more information, see the [deployment architecture](#) section.

- Azure Virtual Machines
 - (1) Management/bastion (Windows Server 2016 Datacenter)
 - (2) Active Directory domain controller (Windows Server 2016 Datacenter)
 - (2) SQL Server cluster node (SQL Server 2017 on Windows Server 2016)
 - (2) Web/IIS (Windows Server 2016 Datacenter)
- Azure Virtual Network
 - (1) /16 network
 - (5) /24 networks
 - (5) Network security groups
- Availability sets
 - (1) Active Directory domain controllers
 - (1) SQL cluster nodes
 - (1) Web/IIS
- Azure Application Gateway
 - (1) Web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP 3.0
 - Listener port: 443
- Azure Active Directory
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure Storage
- Azure Log Analytics

- Azure Automation
- Cloud Witness
- Recovery Services vault

Deployment architecture

The following section details the deployment and implementation elements.

Bastion host: The bastion host is the single point of entry that users can use to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by allowing only remote traffic from public IP addresses on a safe list. To permit remote desktop traffic, the source of the traffic must be defined in the network security group (NSG).

This solution creates a VM as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#).
- [Azure Diagnostics extension](#).
- [Azure Disk Encryption](#) using Key Vault.
- An [auto-shutdown policy](#) to reduce consumption of VM resources when not in use.
- [Windows Defender Credential Guard](#) is enabled so that credentials and other secrets run in a protected environment that's isolated from the running operating system.

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: This solution deploys resources in an architecture with separate subnets for web, database, Active Directory, and management inside a virtual network. Subnets are logically separated by NSG rules applied to the individual subnets. The rules restrict traffic between subnets to only that necessary for system and management functionality.

See the configuration for [NSGs](#) deployed with this solution. Organizations can configure NSGs by editing the previous file by using [this documentation](#) as a guide.

Each of the subnets has a dedicated NSG:

- One NSG for Application Gateway (LBNSG)
- One NSG for bastion host (MGTNSG)
- One NSG for primary and backup domain controllers (ADNSG)
- One NSG for SQL Servers and Cloud Witness (SQLNSG)
- One NSG for web tier (WEBNSG)

Data in transit

Azure encrypts all communications to and from Azure data centers by default. Additionally, all transactions to Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through multiple measures. These measures include encryption and database auditing.

Azure Storage: To meet requirements for encrypted data at rest, all [Storage](#) uses [Storage Service Encryption](#). This feature helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by the NIST SP 800-171.

Azure Disk Encryption: Disk Encryption is used to encrypted Windows IaaS VM disks. [Disk Encryption](#) uses the BitLocker feature of Windows to provide volume encryption for operating system and data disks. The solution is integrated with Key Vault to help control and manage the disk-encryption keys.

SQL Server: The SQL Server instance uses the following database security measures:

- [SQL Server auditing](#) tracks database events and writes them to audit logs.
- [Transparent data encryption](#) performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data hasn't been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [Encrypted columns](#) ensure that sensitive data never appears as plain text inside the database system. After data encryption is enabled, only client applications or application servers with access to the keys can access plain-text data.
- [Dynamic data masking](#) limits sensitive data exposure by masking the data to nonprivileged users or applications. It can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. Dynamic data masking helps to reduce access so that sensitive data doesn't exit the database via unauthorized access. *Customers are responsible for adjusting settings to adhere to their database schema.*

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure AD](#) is Microsoft's multitenant cloud-based directory and identity management service. All users for this solution are created in Azure AD and include the users who access the SQL Server instance.
- Authentication to the application is performed by using Azure AD. For more information, see how to [integrate applications with Azure AD](#).
- [Azure RBAC](#) can be used by administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permissions for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) can be used by customers to minimize the number of users who have access to certain resources. Administrators can use Azure AD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality also can be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities that affect an organization's identities. It configures automated responses to detected suspicious actions related to an organization's identities. It also investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Key Vault](#) for the management of keys and secrets. Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Key Vault capabilities help customers protect data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security-module-protected 2048-bit RSA key.
- All users and identities are granted minimum required permissions by using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.
- The solution is integrated with Key Vault to manage IaaS VM disk-encryption keys and secrets.

Patch management: Windows VMs deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#)

service through which updated deployments can be created to patch VMs when needed.

Malware protection: Microsoft Antimalware for VMs provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software. Customers can configure alerts that generate when known malicious or unwanted software attempts to install or run on protected VMs.

Azure Security Center: With [Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Security Center also accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Security Center uses a variety of detection capabilities to alert customers of potential attacks that target their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Security Center has a set of [predefined security alerts](#) that are triggered when a threat, or suspicious activity takes place. Customers can use [custom alert rules](#) to define new security alerts based on data that's already collected from their environment.

Security Center provides prioritized security alerts and incidents. Security Center makes it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat. Incident response teams can use the reports when they investigate and remediate threats.

This reference architecture uses the [vulnerability assessment](#) capability in Security Center. After it's configured, a partner agent (for example, Qualys) reports vulnerability data to the partner's management platform. In turn, the partner's management platform provides vulnerability and health monitoring data back to Security Center. Customers can use this information to quickly identify vulnerable VMs.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities by using an application gateway with a web application firewall configured and the OWASP rule set enabled. Additional capabilities include:

- [End-to-end-SSL](#).
- Enable [SSL offload](#).
- Disable [TLS v1.0 and v1.1](#).
- [Web application firewall](#) (prevention mode).
- [Prevention mode](#) with OWASP 3.0 rule set.
- Enable [diagnostics logging](#).
- [Custom health probes](#).
- [Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Security Center also provides a reputation system.

Business continuity

High availability: The solution deploys all VMs in an [availability set](#). Availability sets ensure that the VMs are distributed across multiple isolated hardware clusters to improve availability. At least one VM is available during a planned or unplanned maintenance event, which meets the 99.95% Azure SLA.

Recovery Services vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services vault, customers can restore files and folders from an IaaS VM without restoring the entire VM. This process speeds up restore times.

Cloud Witness: [Cloud Witness](#) is a type of failover cluster quorum witness in Windows Server 2016 that uses Azure as the arbitration point. Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations. It uses the standard publicly available Azure Blob storage. This arrangement eliminates the extra maintenance overhead of VMs hosted in a public cloud.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. Users can configure the retention period, up to 730 days, to meet their specific requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. After data is collected, it's organized into separate tables for each data type within Log Analytics workspaces. In this way, all data can be analyzed together, regardless of its original source. Security Center integrates with Log Analytics. Customers can use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

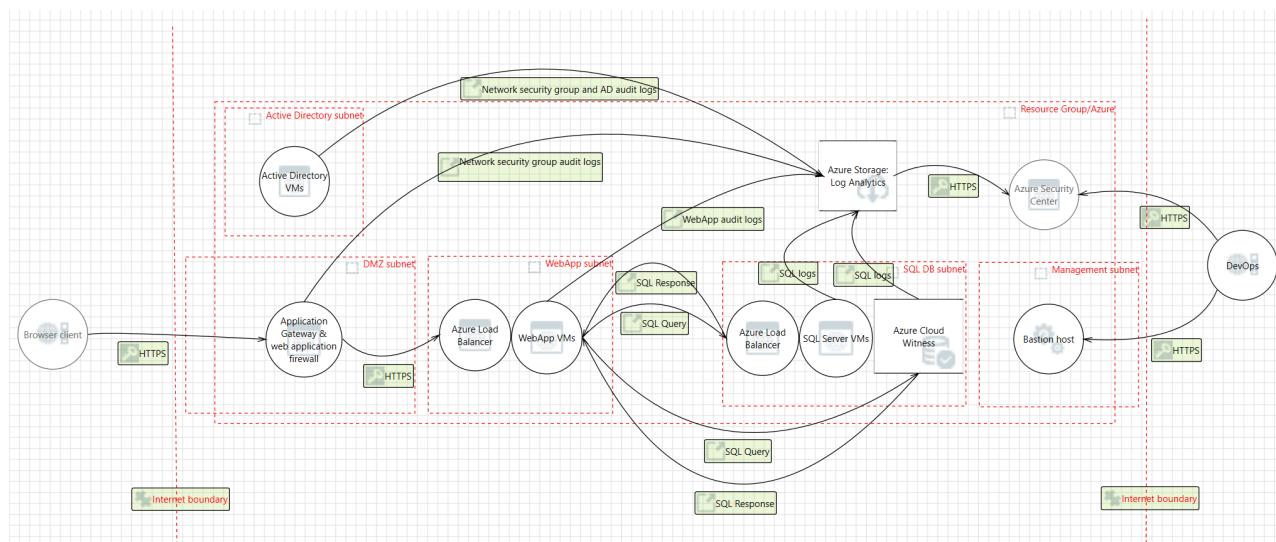
- **Active Directory assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval. It provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **SQL assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval. It provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic distribution. It also reports how many agents are unresponsive and the number of agents that submit operational data.
- **Activity Log Analytics:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from SQL Server. Customers can use the Automation [Change Tracking](#) solution to easily identify changes in the environment.

Azure Monitor: [Monitor](#) helps users track performance, maintain security, and identify trends. Organizations can use it to audit, create alerts, and archive data. They also can track API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found here. This model can help customers understand the points of potential risk in the system infrastructure when they make modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - NIST SP 800-171 Customer Responsibility Matrix](#) lists all security controls required by NIST SP 800-171. This matrix details whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - NIST SP 800-171 IaaS Web Application Control Implementation Matrix](#) provides information on which NIST SP 800-171 controls are addressed by the IaaS web application architecture. It includes detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) must be configured to securely establish a connection to the resources deployed as a part of this IaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet. Customers can use this connection to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel traverses the Internet with a site-to-site VPN, Microsoft offers another even more secure connection option. ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. ExpressRoute connections connect directly to the customer's telecommunication provider. As a result, the data doesn't travel over the Internet and isn't exposed to it. These connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - PaaS Web Application for NIST Special Publication 800-171

10/18/2018 • 16 minutes to read • [Edit Online](#)

Overview

NIST Special Publication 800-171 provides guidelines for protecting the controlled unclassified information (CUI) that resides in nonfederal information systems and organizations. NIST SP 800-171 establishes 14 families of security requirements for protecting the confidentiality of CUI.

This Azure Security and Compliance Blueprint provides guidance to help customers deploy a platform as a service (PaaS) web application in Azure that implements a subset of NIST SP 800-171 controls. This solution demonstrates ways in which customers can meet specific security and compliance requirements. It also serves as a foundation for customers to build and configure their own web application in Azure.

This reference architecture, associated implementation guide, and threat model are intended to serve as a foundation for customers to adapt to their specific requirements. They shouldn't be used as-is in a production environment. Deploying this architecture without modification is insufficient to completely meet the requirements of NIST SP 800-171. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture. Requirements might vary based on the specifics of each customer's implementation.

Architecture diagram and components

This Azure Security and Compliance Blueprint provides a reference architecture for a PaaS web application with an Azure SQL Database back end. The web application is hosted in an isolated App Service environment, which is a private, dedicated environment in an Azure data center. The environment load balances traffic for the web application across virtual machines (VMs) managed by Azure. This architecture also includes network security groups (NSGs), an Azure application gateway, Azure DNS, and Azure Load Balancer.

For enhanced analytics and reporting, Azure SQL databases can be configured with columnstore indexes. Azure SQL databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

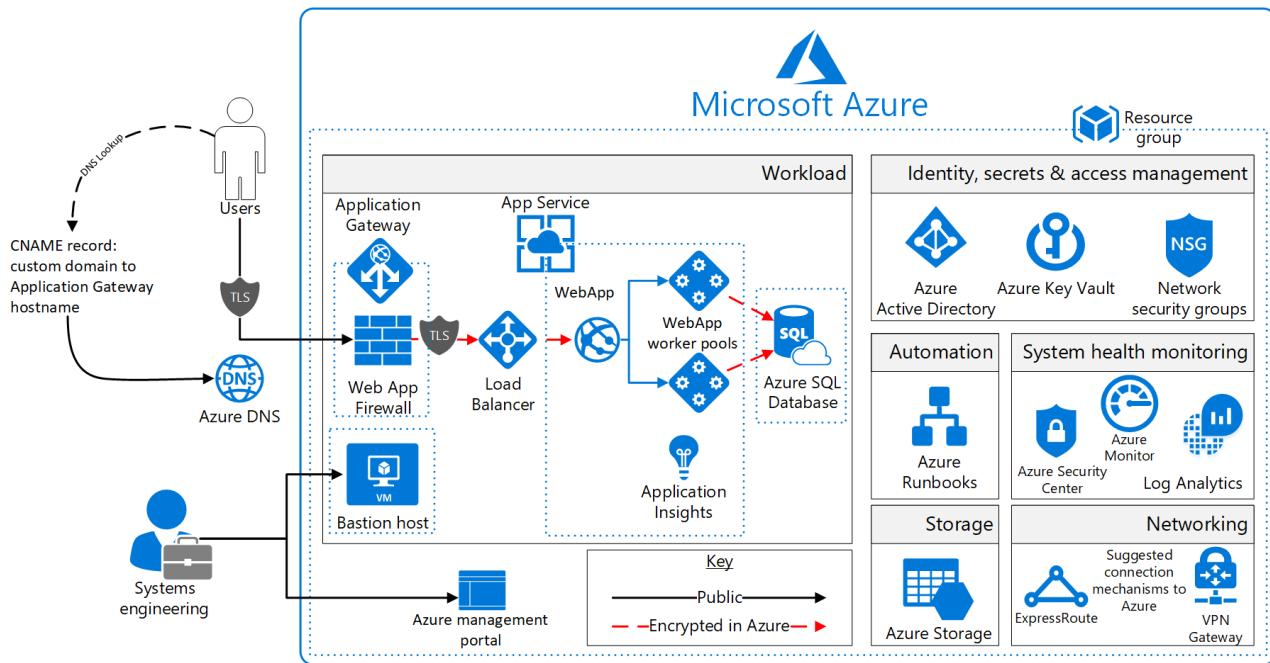
The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected data center for resiliency. Geo-redundant storage ensures that data is replicated to a secondary data center hundreds of miles away and stored again as three copies within that data center. This arrangement prevents an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory (Azure AD) role-based access control (RBAC) is used to control access to deployed resources. These resources include customer keys in Azure Key Vault. System health is monitored through Azure Monitor. Customers configure this monitoring service to capture logs. System health is displayed in a single dashboard that's easy to use.

SQL Database is commonly managed through SQL Server Management Studio. It runs from a local machine that's configured to access the SQL database via a secure VPN or Azure ExpressRoute connection.

Application Insights provides real-time application performance management and analytics through Azure Log

Analytics. Microsoft recommends that you configure a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.



This solution uses the following Azure services. For more information, see the [deployment architecture](#) section.

- Azure Virtual Machines
 - (1) Management/bastion (Windows Server 2016 Datacenter)
- Azure Virtual Network
 - (1) /16 network
 - (4) /24 networks
 - (4) Network security groups
- Azure Application Gateway
 - Web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP
 - Listener port: 443
- Application Insights
- Azure Active Directory
- App Service Environment v2
- Azure Automation
- Azure DNS
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Log Analytics
- Azure Automation
- Azure Web Apps

Deployment architecture

The following section details the deployment and implementation elements.

Azure Resource Manager: [Resource Manager](#) can be used by customers to work with the resources in the solution as a group. Customers can deploy, update, or delete all the resources for the solution in a single, coordinated operation. Customers use a template for deployment. The template can work for different environments, such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help customers manage their resources after deployment.

Bastion host: The bastion host is the single point of entry that users can use to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by allowing only remote traffic from public IP addresses on a safe list. To permit remote desktop traffic, the source of the traffic must be defined in the NSG.

This solution creates a VM as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#).
- [Azure Diagnostics extension](#).
- [Azure Disk Encryption](#) using Key Vault.
- An [auto-shutdown policy](#) to reduce consumption of VM resources when not in use.
- [Windows Defender Credential Guard](#) is enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system.

Web Apps: [Web Apps](#) is an Azure App Service feature. Customers can use it to build and host web applications in the programming language of their choice without managing infrastructure. It offers auto-scaling and high availability. It supports Windows and Linux and enables automated deployments from GitHub, Azure DevOps, or any Git repo.

App Service Environment: [App Service Environment](#) is an App Service feature. It provides a fully isolated and dedicated environment for securely running App Service applications at a high scale.

The App Service environment is isolated to run only a single application. It's always deployed into a virtual network. Because of the isolation feature, the reference architecture has complete tenant isolation, and it's removed from Azure's multitenant environment. Customers have fine-grained control over both inbound and outbound application network traffic. Applications can establish high-speed secure connections over virtual networks to on-premises corporate resources. Customers can "auto-scale" with App Service Environment based on load metrics, available budget, or a defined schedule.

Use of App Service Environment for this architecture provides the following controls and configurations:

- Host inside a secured Azure virtual network and network security rules.
- Self-signed internal load balancer certificate for HTTPS communication. As a best practice, Microsoft recommends the use of a trusted certificate authority for enhanced security.
- [Internal load balancing mode](#) (mode 3).
- Disable [TLS 1.0](#).
- Change [TLS cipher](#).
- Control [inbound traffic N/W ports](#).
- [Web application firewall – restrict data](#).
- Allow [Azure SQL Database traffic](#).

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: [NSGs](#) contain access control lists that allow or deny traffic within a virtual network.

NSGs can be used to secure traffic at a subnet or individual VM level. The following NSGs exist:

- One NSG for Application Gateway
- One NSG for App Service Environment
- One NSG for SQL Database
- One NSG for bastion host

Each of the NSGs has specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each NSG:

- [Diagnostic logs and events](#) are enabled and stored in a storage account.
- Log Analytics is connected to the [NSG's diagnostics](#).

Subnets: Each subnet is associated with its corresponding NSG.

Azure DNS: The Domain Name System (DNS) is responsible for translating (or resolving) a website or service name to its IP address. [Azure DNS](#) is a hosting service for DNS domains that provides name resolution by using Azure infrastructure. By hosting domains in Azure, users can manage DNS records by using the same credentials, APIs, tools, and billing as other Azure services. Azure DNS also supports private DNS domains.

Azure Load Balancer: [Load Balancer](#) can be used by customers to scale their applications and create high availability for services. Load Balancer supports inbound and outbound scenarios. It provides low latency and high throughput and scales up to millions of flows for all TCP and UDP applications.

Data in transit

Azure encrypts all communications to and from Azure data centers by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet requirements for encrypted data at rest, all [Storage](#) uses [Storage Service Encryption](#). This feature helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by NIST SP 800-171.

Azure Disk Encryption: [Disk Encryption](#) uses the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [transparent data encryption](#). It performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data hasn't been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur. It provides security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted columns](#) ensure that sensitive data never appears as plain text inside the database system. After data encryption is enabled, only client applications or application servers with access to the keys can access plain-text data.
- [Dynamic data masking](#) limits sensitive data exposure by masking the data to nonprivileged users or applications. It can automatically discover potentially sensitive data and suggest the appropriate masks to be

applied. Dynamic data masking helps to reduce access so that sensitive data doesn't exit the database via unauthorized access. *Customers are responsible for adjusting settings to adhere to their database schema.*

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure AD](#) is Microsoft's multitenant cloud-based directory and identity management service. All users for this solution are created in Azure AD and include users who access the SQL database.
- Authentication to the application is performed by using Azure AD. For more information, see how to [integrate applications with Azure AD](#). The database column encryption also uses Azure AD to authenticate the application to SQL Database. For more information, see how to [protect sensitive data in SQL Database](#).
- [Azure RBAC](#) can be used by administrators to define fine-grained access permissions. With it, they can grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted access for Azure resources, administrators can allow only certain actions for accessing resources and data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) can be used by customers to minimize the number of users who have access to certain information. Administrators can use Azure AD Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality also can be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities that affect an organization's identities. It configures automated responses to detected suspicious actions related to an organization's identities. It also investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Key Vault](#) for the management of keys and secrets. Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Key Vault capabilities help customers protect data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security-module-protected 2048-bit RSA key.
- All users and identities are granted minimum required permissions by using RBAC.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Security Center also accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Security Center uses a variety of detection capabilities to alert customers of potential attacks that target their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Security Center has a set of [predefined security alerts](#) that are triggered when a threat or suspicious activity takes place. Customers can use [custom alert rules](#) to define new security alerts based on data that's already collected from their environment.

Security Center provides prioritized security alerts and incidents. Security Center makes it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat. Incident response teams can use the reports when they investigate and remediate threats.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities by using an application gateway with a web application firewall configured and the OWASP rule set enabled. Additional capabilities

include:

- [End-to-end-SSL](#).
- Enable [SSL offload](#).
- Disable [TLS v1.0 and v1.1](#).
- [Web application firewall](#) (prevention mode).
- [Prevention mode](#) with OWASP 3.0 rule set.
- Enable [diagnostics logging](#).
- [Custom health probes](#).
- [Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Security Center also provides a reputation system.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. Users can configure the retention period, up to 730 days, to meet their specific requirements.

Azure Log Analytics: Logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. After the data is collected, it's organized into separate tables for each data type within Log Analytics workspaces. In this way, all data can be analyzed together, regardless of its original source. Security Center integrates with Log Analytics. Customers can use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- [Active Directory assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval. It provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval. It provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution. It also reports how many agents are unresponsive and the number of agents that submit operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

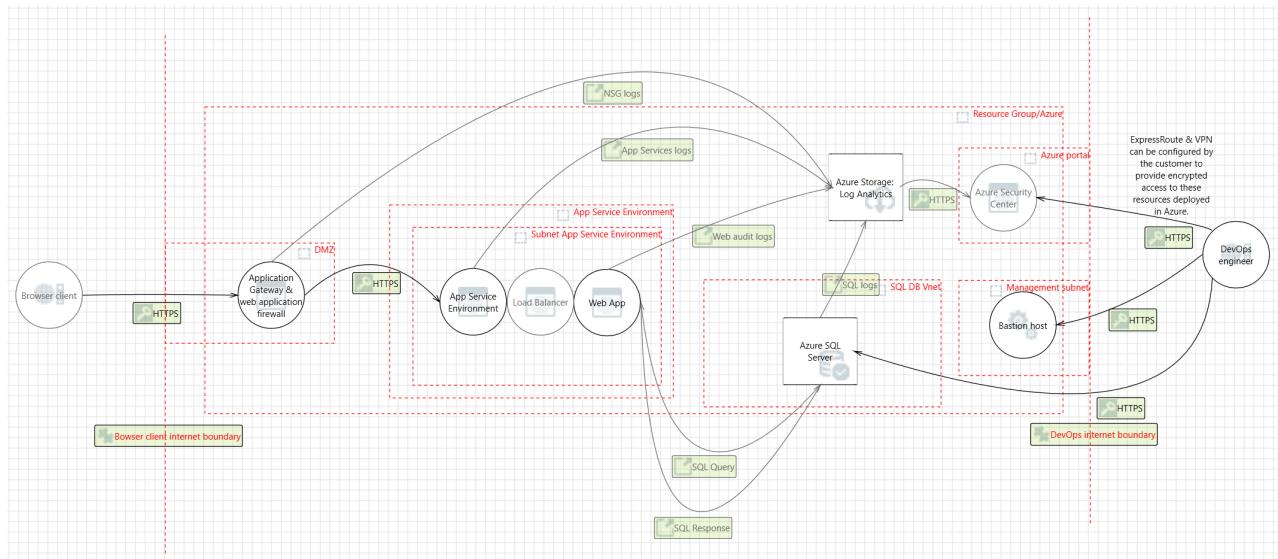
Azure Automation: [Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from SQL Database. Customers can use the Automation [Change Tracking](#) solution to easily identify changes in the environment.

Azure Monitor: [Monitor](#) helps users track performance, maintain security, and identify trends. Organizations can use it to audit, create alerts, and archive data. They also can track API calls in their Azure resources.

Application Insights: [Application Insights](#) is an extensible application performance management service for web developers on multiple platforms. Application Insights detects performance anomalies. Customers can use it to monitor the live web application. Application Insights includes powerful analytics tools to help customers diagnose issues and understand what users do with their app. It's designed to help customers continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found here. This model can help customers understand the points of potential risk in the system infrastructure when they make modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - NIST SP 800-171 Customer Responsibility Matrix](#) lists all security controls required by NIST SP 800-171. This matrix details whether the implementation of each control is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - NIST SP 800-171 PaaS Web Application Control Implementation Matrix](#) provides information on which NIST SP 800-171 controls are addressed by the PaaS web application architecture. It includes detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) must be configured to securely establish a connection to the resources deployed as a part of this PaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection takes place over the Internet and allows customers to securely “tunnel” information inside an encrypted link between the customer’s network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel traverses the Internet with a site-to-site VPN, Microsoft offers another even more secure connection option. ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. ExpressRoute connections connect directly to a customer’s telecommunication provider. As a result, the data doesn’t travel over the Internet and isn’t exposed to it. These connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Analytics for PCI DSS

12/6/2018 • 16 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of a data analytics architecture in Azure that assists with the requirements of Payment Card Industry Data Security Standards (PCI DSS 3.2). It showcases a common reference architecture and demonstrates the proper handling of credit card data (including card number, expiration, and verification data) in a secure, compliant, multi-tier environment. This blueprint demonstrates ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own data analytics solutions in Azure.

This reference architecture, implementation guide, and threat model provide a foundation for customers to comply with PCI DSS 3.2 requirements. This solution provides a baseline to help customers deploy workloads to Azure in a PCI DSS 3.2 compliant manner; however, this solution should not be used as-is in a production environment because additional configuration is required.

Achieving PCI DSS-compliance requires that an accredited Qualified Security Assessor (QSA) certify a production customer solution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This Azure Security and Compliance Blueprint provides an analytics platform upon which customers can build their own analytics tools. The reference architecture outlines a generic use case where customers input data either through bulk data imports by the SQL/Data Administrator or through operational data updates via an Operational User. Both work streams incorporate Azure Functions for importing data into Azure SQL Database. Azure Functions must be configured by the customer through the Azure portal to handle the import tasks unique to each customer's own analytics requirements.

Azure offers a variety of reporting and analytics services for the customers. This solution incorporates Azure Machine Learning services in conjunction with Azure SQL Database to rapidly browse through data and deliver faster results through smarter modeling. Azure Machine Learning increases query speeds by discovering new relationships between datasets. Once the data has been trained through several statistical functions, up to 7 additional query pools (8 total including the customer server) can be synchronized with the same tabular models to spread query workloads and reduce response times.

For enhanced analytics and reporting, Azure SQL databases can be configured with columnstore indexes. Both Azure Machine Learning and Azure SQL databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

Once data is uploaded to the Azure SQL Database and trained by Azure Machine Learning, it is digested by both the Operational User and SQL/Data Admin with Power BI. Power BI displays data intuitively and pulls together information across multiple datasets to draw greater insight. Its high degree of adaptability and easy integration with Azure SQL Database ensures that customers can configure it to handle a wide array of scenarios as required by their business needs.

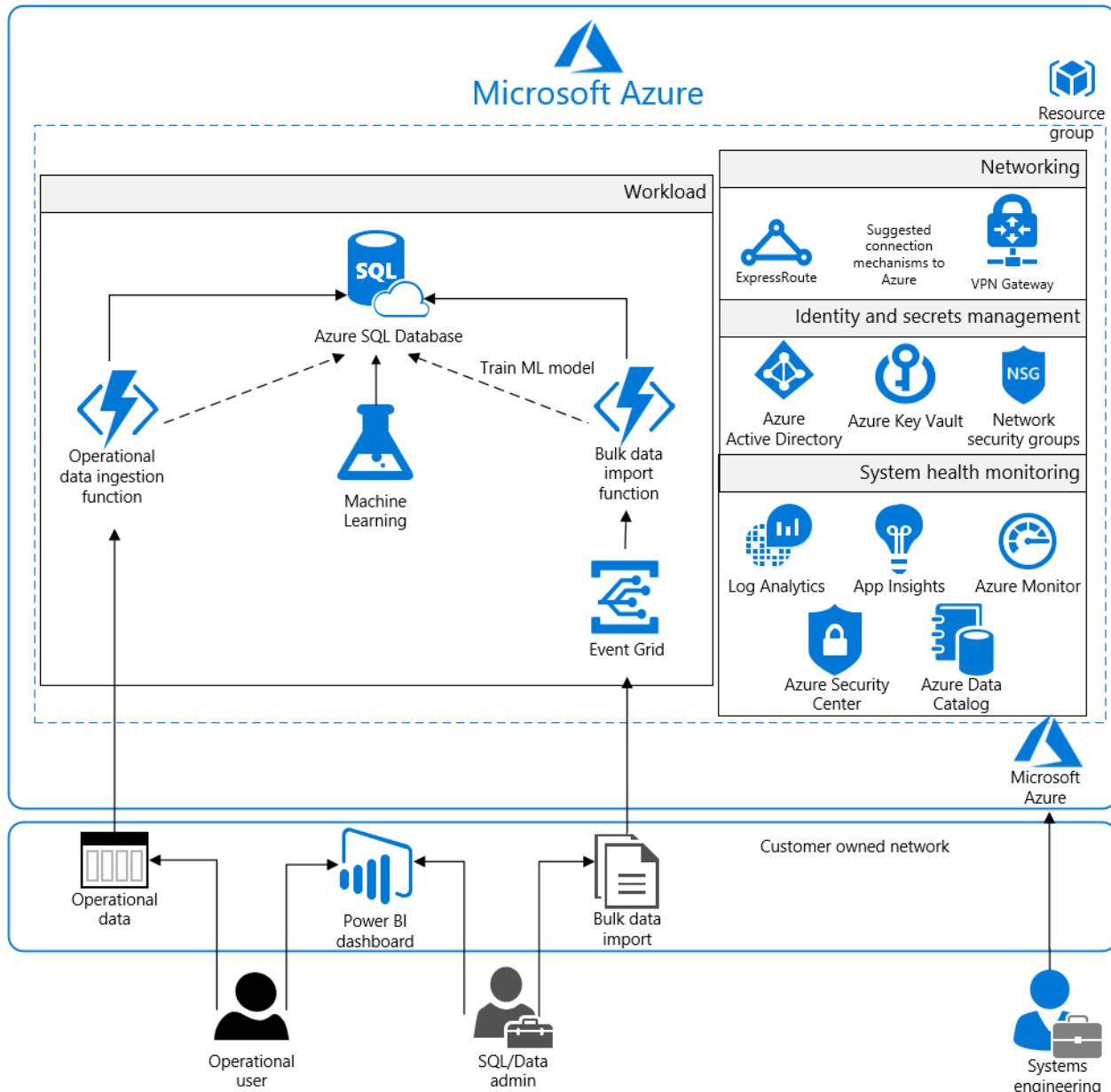
The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to

maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and again stored as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

Azure SQL Database is commonly managed through SQL Server Management Studio (SSMS), which runs from a local machine configured to access the Azure SQL Database via a secure VPN or ExpressRoute connection.

Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture resource group.



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- Application Insights
- Azure Active Directory
- Azure Data Catalog

- Azure Disk Encryption
- Azure Event Grid
- Azure Functions
- Azure Key Vault
- Azure Machine Learning
- Azure Monitor
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Virtual Network
 - (1) /16 Network
 - (2) /24 Networks
 - (2) Network Security Groups
- Power BI Dashboard

Deployment architecture

The following section details the deployment and implementation elements.

Azure Event Grid: [Azure Event Grid](#) allows customers to easily build applications with event-based architectures. Users select the Azure resource they would like to subscribe to and give the event handler or webhook an endpoint to send the event to. Customers can secure webhook endpoints by adding query parameters to the webhook URL when creating an Event Subscription. Azure Event Grid only supports HTTPS webhook endpoints. Azure Event Grid allows customers to control the level of access given to different users to do various management operations such as list event subscriptions, create new ones, and generate keys. Event Grid utilizes Azure role-based access control.

Azure Functions: [Azure Functions](#) is a server-less compute service that enables users to run code on-demand without having to explicitly provision or manage infrastructure. Use Azure Functions to run a script or piece of code in response to a variety of events.

Azure Machine Learning service: [Azure Machine Learning](#) is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends.

Azure Data Catalog: [Data Catalog](#) makes data sources easily discoverable and understandable by the users who manage the data. Common data sources can be registered, tagged, and searched for financial data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual VM level. The following network security groups exist:

- A network security group for Active Directory
- A network security group for the workload

Each of the network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account

- Log Analytics is connected to the [network security group's diagnostic logs](#)

Subnets: Each subnet is associated with its corresponding network security group.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#).

This helps protect and safeguard cardholder data in support of organizational security commitments and compliance requirements defined by PCI DSS 3.2.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Extended Properties](#) can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated financial data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).

- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center

integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

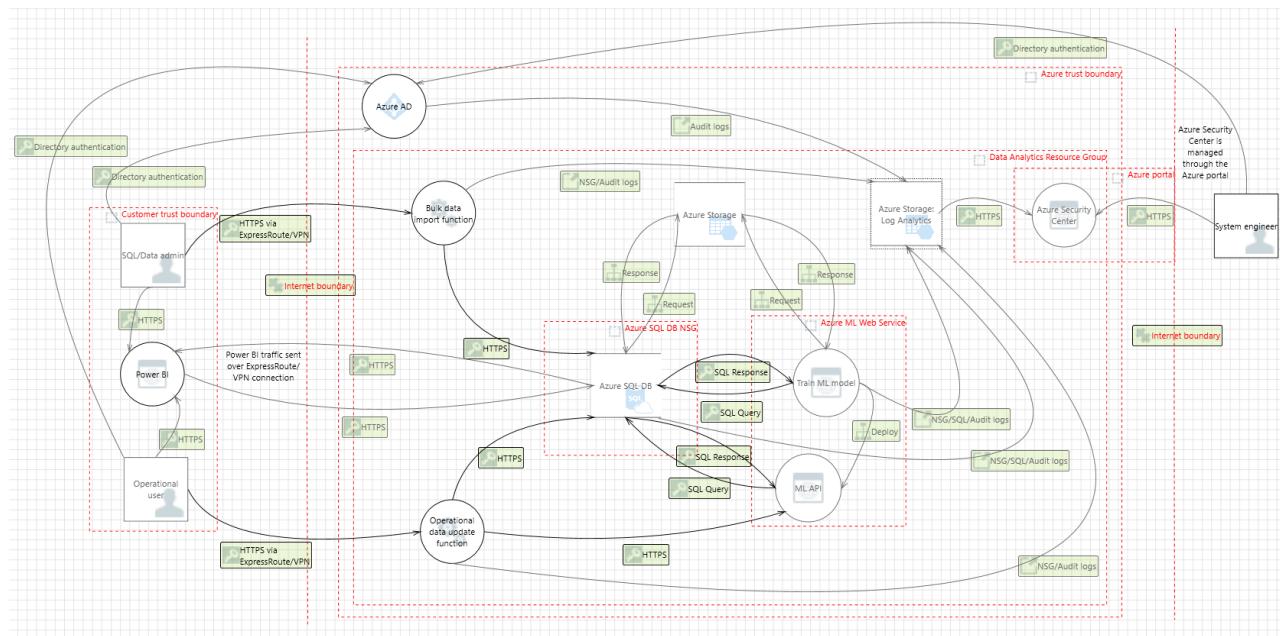
Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Application Insights: [Application Insights](#) is an extensible Application Performance Management (APM) service for web developers on multiple platforms. It detects performance anomalies and includes powerful analytics tools to help diagnose issues and to understand what users actually do with the app. It's designed to help users continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - PCI DSS Customer Responsibility Matrix](#) lists responsibilities for all PCI DSS 3.2 requirements.

The [Azure Security and Compliance Blueprint - PCI DSS Data Analytics Implementation Matrix](#) provides information on which PCI DSS 3.2 requirements are addressed by the data analytics architecture, including detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this data analytics reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-Transform-Load process

[PolyBase](#) can load data into Azure SQL Database without the need for a separate extract, transform, load or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing Azure Active Directory by making the deployed Active Directory infrastructure a subdomain of an Azure Active Directory forest.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can

meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Data Warehouse for PCI DSS

10/18/2018 • 18 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of a data warehouse architecture in Azure that assists with the requirements of Payment Card Industry Data Security Standards (PCI DSS 3.2). It showcases a common reference architecture and demonstrates the proper handling of credit card data (including card number, expiration, and verification data) in a secure, compliant, multi-tier environment. This blueprint demonstrates ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own data warehouse solutions in Azure.

This reference architecture, implementation guide, and threat model provide a foundation for customers to comply with PCI DSS 3.2 requirements. This solution provides a baseline to help customers deploy workloads to Azure in a PCI DSS 3.2 compliant manner; however, this solution should not be used as-is in a production environment because additional configuration is required.

Achieving PCI DSS-compliance requires that an accredited Qualified Security Assessor (QSA) certify a production customer solution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture which implements a high-performance and secure cloud data warehouse. There are two separate data tiers in this architecture: one where data is imported, stored, and staged within a clustered SQL environment, and another for the Azure SQL Data Warehouse where the data is loaded using an extract, transform, load tool (e.g. [PolyBase](#) T-SQL queries) for processing. Once data is stored in Azure SQL Data Warehouse, analytics can run at a massive scale.

Azure offers a variety of reporting and analytics services for the customer. This solution includes SQL Server Reporting Services (SSRS) for quick creation of reports from the Azure SQL Data Warehouse. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

Data in the Azure SQL Data Warehouse is stored in relational tables with columnar storage, a format that significantly reduces the data storage costs while improving query performance. Depending on usage requirements, Azure SQL Data Warehouse compute resources can be scaled up or down or shut off completely if there are no active processes requiring compute resources.

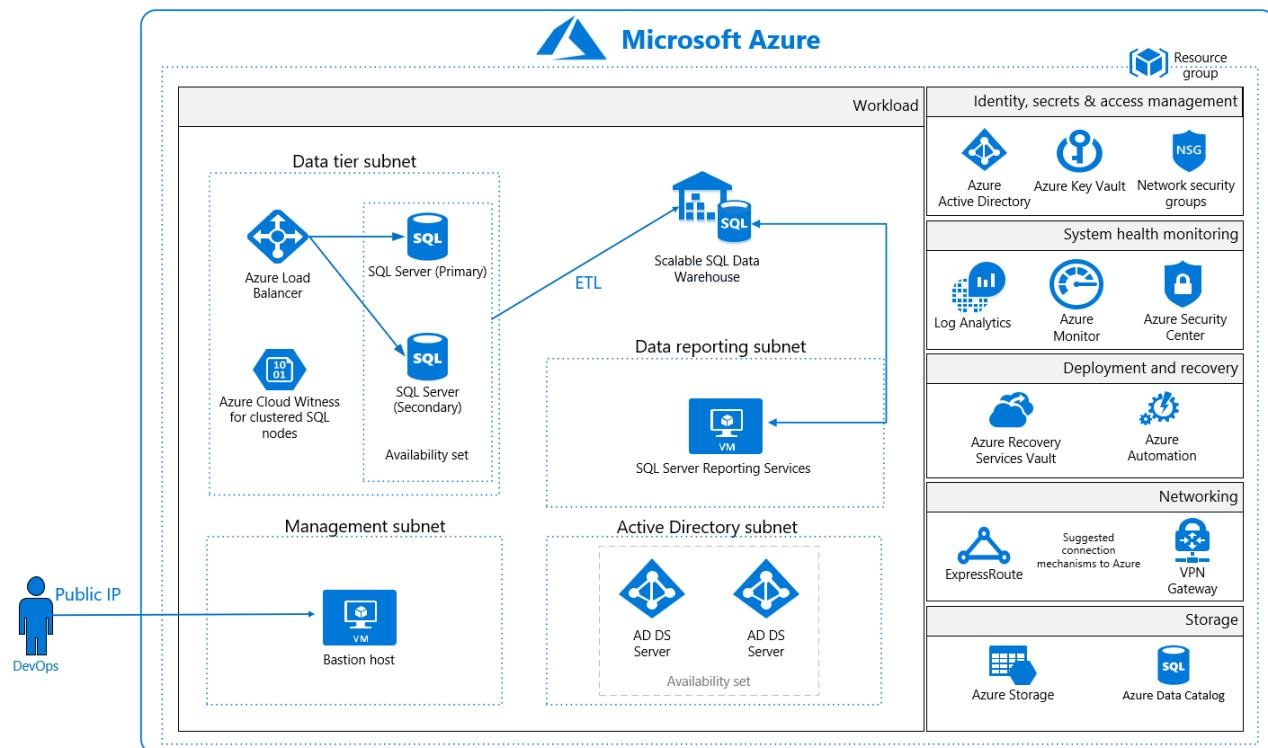
A SQL Load Balancer manages SQL traffic, ensuring high performance. All virtual machines in this reference architecture deploy in an availability set with SQL Server instances configured in an Always On availability group for high-availability and disaster-recovery capabilities.

This data warehouse reference architecture also includes an Active Directory tier for management of resources within the architecture. The Active Directory subnet enables easy adoption under a larger Active Directory forest structure, allowing for continuous operation of the environment even when access to the larger forest is unavailable. All virtual machines are domain-joined to the Active Directory tier and use Active Directory group policies to enforce security and compliance configurations at the operating system level.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and again stored as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

A virtual machine serves as a management bastion host, providing a secure connection for administrators to access deployed resources. The data loads into the staging area through this management bastion host. **Microsoft recommends configuring a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- Availability Sets
 - Active Directory Domain Controllers
 - SQL Cluster Nodes and Witness
- Azure Active Directory
- Azure Data Catalog
- Azure Key Vault
- Azure Monitor
- Azure Security Center
- Azure Load Balancer
- Azure Storage
- Azure Virtual Machines
 - (1) Bastion Host

- (2) Active Directory domain controller
- (2) SQL Server Cluster Node
- (1) SQL Server Witness
- Azure Virtual Network
 - (1) /16 Network
 - (4) /24 Networks
 - (4) Network Security Groups
- Recovery Services Vault
- SQL Data Warehouse
- SQL Server Reporting Services

Deployment architecture

The following section details the deployment and implementation elements.

SQL Data Warehouse: [SQL Data Warehouse](#) is an Enterprise Data Warehouse (EDW) that leverages Massively Parallel Processing (MPP) to quickly run complex queries across petabytes of data, allowing users to efficiently identify financial data. Users can use simple PolyBase T-SQL queries to import big data into the SQL Data Warehouse and utilize the power of MPP to run high-performance analytics.

SQL Server Reporting Services (SSRS): [SQL Server Reporting Services](#) provides quick creation of reports with tables, charts, maps, gauges, matrixes, and more for Azure SQL Data Warehouse.

Data Catalog: [Data Catalog](#) makes data sources easily discoverable and understandable by the users who manage the data. Common data sources can be registered, tagged, and searched for financial data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system.

Virtual network

This reference architecture defines a private virtual network with an address space of 10.0.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual virtual machine level. The following network security groups exist:

- A network security group for the Data Tier (SQL Server Clusters, SQL Server Witness, and SQL Load Balancer)
- A network security group for the management bastion host
- A network security group for Active Directory
- A network security group for SQL Server Reporting Services

Each of the network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [network security group's diagnostic logs](#)

Subnets: Each subnet is associated with its corresponding network security group.

Data at rest

The architecture protects data at rest through multiple measures, including encryption and database auditing.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard cardholder data in support of organizational security commitments and compliance requirements defined by PCI DSS 3.2.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for OS and data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [Extended Properties](#) can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated financial data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).

- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for virtual machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Business continuity

High availability: Server workloads are grouped in an [Availability Set](#) to help ensure high availability of virtual machines in Azure. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure virtual machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS virtual machine without restoring the entire virtual machine, enabling faster restore times.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

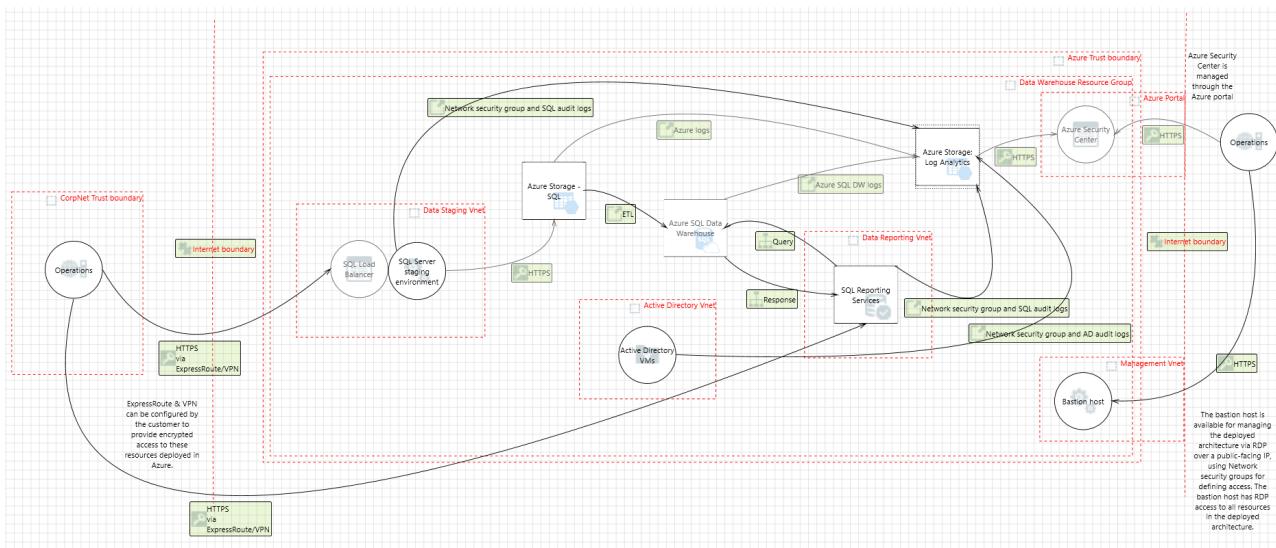
- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – PCI DSS Customer Responsibility Matrix](#) lists responsibilities for all PCI DSS 3.2 requirements.

The [Azure Security and Compliance Blueprint - PCI DSS Data Warehouse Implementation Matrix](#) provides information on which PCI DSS 3.2 requirements are addressed by the data warehouse architecture, including detailed descriptions of how the implementation meets the requirements of each covered control.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this data warehouse reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely “tunnel” information inside an encrypted link between the customer’s network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer’s telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-Transform-Load process

[PolyBase](#) can load data into Azure SQL Data Warehouse without the need for a separate extract, transform, load or import tool. PolyBase allows access to data through T-SQL queries. Microsoft’s business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting

with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing Azure Active Directory by making the deployed Active Directory infrastructure a subdomain of an Azure Active Directory forest.

Optional services

Azure offers a variety of services to assist with the storage and staging of formatted and unformatted data. The following services can be added to this reference architecture depending on customer requirements:

- [Azure Data Factory](#) is a managed cloud service that is built for complex hybrid extract-transform-load, and data integration projects. Azure Data Factory has capabilities to help trace and locate cardholder data, including visualization and monitoring tools to identify when data arrived and where it came from. Using Azure Data Factory, customers can create and schedule data-driven workflows called pipelines that ingest data from disparate data stores. These pipelines allow customers to ingest data from both internal and external sources. Customers can then process and transform the data for output into data stores such as Azure SQL Data Warehouse.
- Customers can stage unstructured data in [Azure Data Lake Store](#), which enables the capture of data of any size, type, and ingestion speed in a single place for operational and exploratory analytics. Azure Data Lake has capabilities that enable the extraction and conversion of data. Azure Data Lake Store is compatible with most open source components in the Hadoop ecosystem and integrates nicely with other Azure services such as Azure SQL Data Warehouse.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: IaaS Web Application for PCI DSS

10/18/2018 • 14 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides guidance for the deployment of a Payment Card Industry Data Security Standards (PCI DSS 3.2) compliant infrastructure as a service (IaaS) environment suitable for the collection, storage, and retrieval of cardholder data. It showcases a common reference architecture and demonstrates the proper handling of credit card data (including card number, expiration, and verification data) in a secure, compliant, multi-tier environment. This blueprint illustrates an end-to-end solution to meet the needs of organizations seeking a cloud-based approach to reducing the burden and cost of deployment.

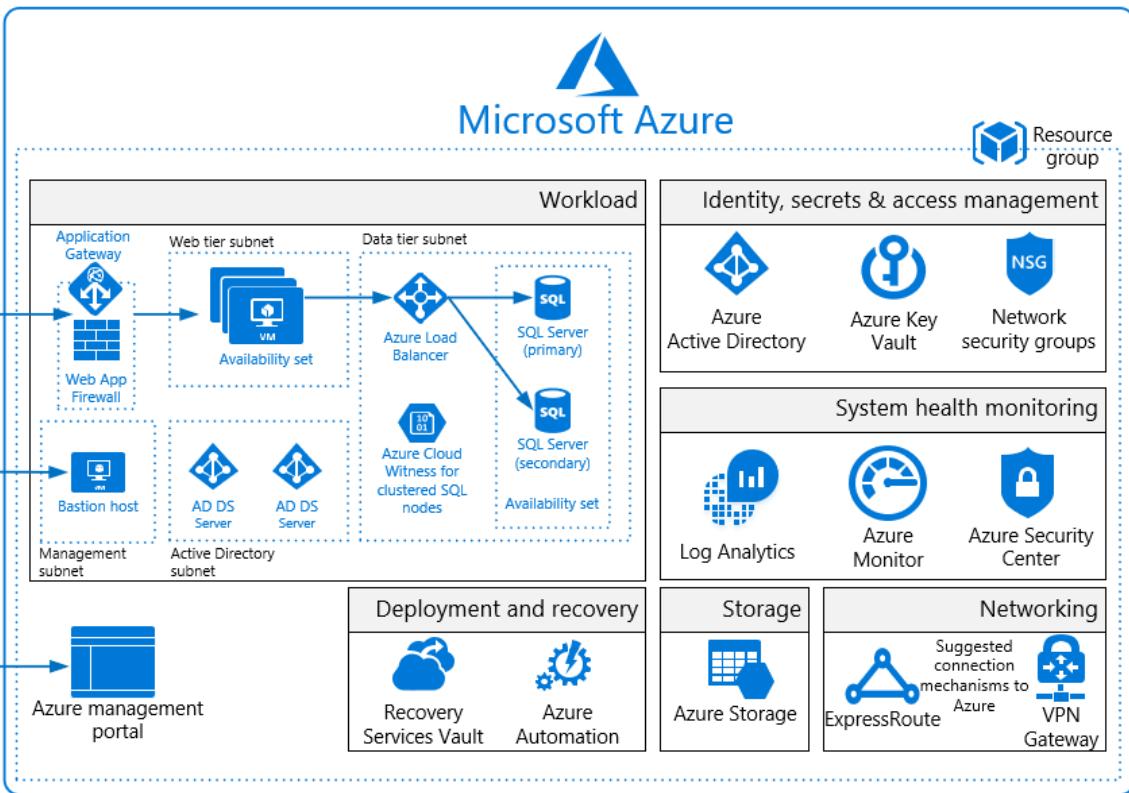
This reference architecture, implementation guide, and threat model provide a foundation for customers to comply with PCI DSS 3.2 requirements. This solution provides a baseline to help customers deploy workloads to Azure in a PCI DSS 3.2 compliant manner; however, this solution should not be used as-is in a production environment because additional configuration is required.

Achieving PCI DSS-compliance requires that an accredited Qualified Security Assessor (QSA) certify a production customer solution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution deploys a reference architecture for an IaaS web application with a SQL Server backend. The architecture includes a web tier, data tier, Active Directory infrastructure, Application Gateway, and Load Balancer. Virtual machines deployed to the web and data tiers are configured in an availability set, and SQL Server instances are configured in an Always On availability group for high availability. Virtual machines are domain-joined, and Active Directory group policies are used to enforce security and compliance configurations at the operating system level. A management bastion host provides a secure connection for administrators to access deployed resources.

Azure recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

- Availability Sets
 - (1) Active Directory domain controllers
 - (1) SQL cluster nodes
 - (1) Web/IIS
- Azure Active Directory
- Azure Application Gateway
 - (1) Web Application Firewall
 - Firewall mode: prevention
 - Rule set: OWASP 3.0
 - Listener port: 443
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure Storage
 - (7) Geo-redundant storage accounts
- Azure Virtual Machines
 - (1) management/bastion (Windows Server 2016 Datacenter)
 - (2) Active Directory domain controller (Windows Server 2016 Datacenter)
 - (2) SQL Server cluster node (SQL Server 2017 on Windows Server 2016)
 - (2) Web/IIS (Windows Server 2016 Datacenter)
- Azure Virtual Network
 - (1) /16 Network
 - (5) /24 Networks

- (5) Network Security Groups
- Cloud Witness
- Recovery Services Vault

Deployment architecture

The following section details the deployment and implementation elements.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the Network Security Group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: This solution deploys resources in an architecture with a separate web subnet, database subnet, Active Directory subnet, and management subnet inside of a virtual network. Subnets are logically separated by network security group rules applied to the individual subnets to restrict traffic between subnets to only that necessary for system and management functionality.

See the configuration for [Network Security Groups](#) deployed with this solution. Organizations can configure Network Security Groups by editing the file above using [this documentation](#) as a guide.

Each of the subnets has a dedicated Network Security Group:

- 1 Network Security Group for Application Gateway (LBNSG)
- 1 Network Security Group for bastion host (MGTNSG)
- 1 Network Security Group for primary and backup domain controllers (ADNSG)
- 1 Network Security Group for SQL Servers and Cloud Witness (SQLNSG)
- 1 Network Security Group for web tier (WEBNSG)

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. Additionally, all transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through multiple measures, including encryption and database auditing.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard cardholder data in support of organizational security commitments and compliance requirements defined by PCI DSS 3.2.

Azure Disk Encryption: Azure Disk Encryption is used to encrypted Windows IaaS virtual machine disks. [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for OS and data disks. The solution is integrated with Azure Key Vault to help control and manage the disk-encryption keys.

SQL Server: The SQL Server instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- SQL Database is configured to use [Transparent Data Encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent Data Encryption provides assurance that stored cardholder data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Always Encrypted Columns](#) ensure that sensitive cardholder data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- The [Extended Properties](#) feature can be used to discontinue the processing of data subjects, as it allows users to add custom properties to database objects and tag data as "Discontinued" to support application logic to prevent the processing of associated cardholder data.
- [Row-Level Security](#) enables users to define policies to restrict access to data to discontinue processing.
- [SQL Database dynamic data masking](#) limits sensitive cardholder data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to cardholder data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.
- The solution is integrated with Azure Key Vault to manage IaaS virtual machine disk-encryption keys and secrets.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities using an Azure Application Gateway with a web application firewall configured, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Business continuity

High availability: The solution deploys all virtual machines in an [Availability Set](#). Availability sets ensure that the virtual machines are distributed across multiple isolated hardware clusters to improve availability. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of

Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS virtual machine without restoring the entire virtual machine, enabling faster restore times.

Cloud Witness: [Cloud Witness](#) is a type of Failover Cluster quorum witness in Windows Server 2016 that leverages Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations, but it uses the standard publicly available Azure Blob Storage. This eliminates the extra maintenance overhead of virtual machines hosted in a public cloud.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

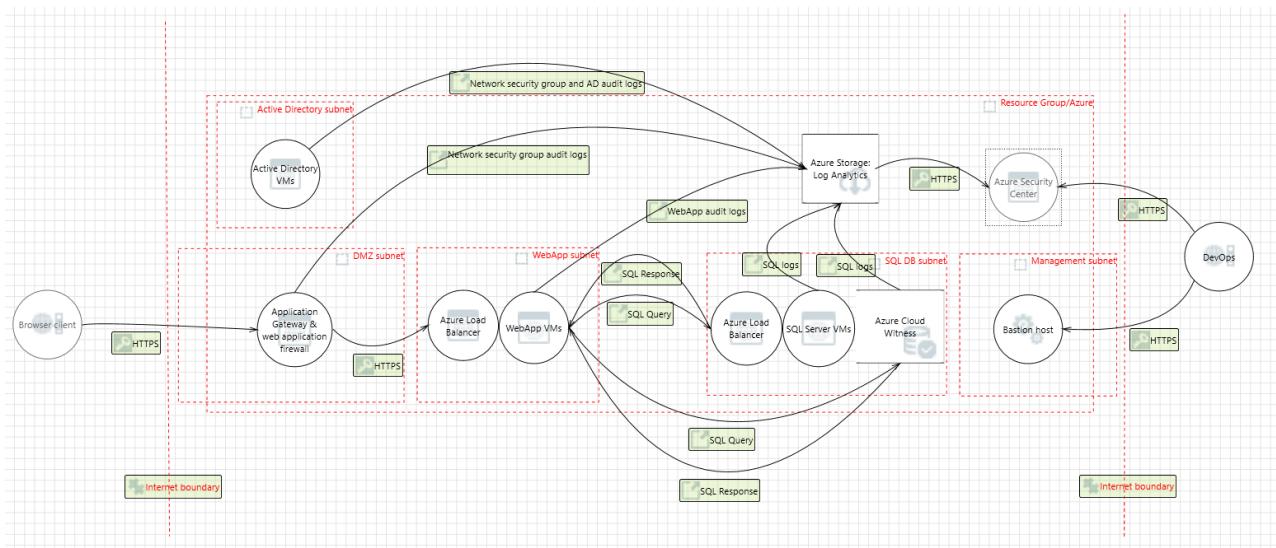
- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram (DFD) for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint - PCI DSS Customer Responsibility Matrix](#) lists responsibilities for all PCI DSS 3.2 requirements.

The [Azure Security and Compliance Blueprint - PCI DSS IaaS Web Application Implementation Matrix](#) provides information on which PCI DSS 3.2 requirements are addressed by the IaaS Web Application architecture, including detailed descriptions of how the implementation meets the requirements of each covered article.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this IaaS Web Application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely “tunnel” information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft

product or solutions.

- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: PaaS Web Application for PCI DSS

10/18/2018 • 17 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint Automation provides guidance for the deployment of a Payment Card Industry Data Security Standards (PCI DSS 3.2) compliant platform as a service (PaaS) environment suitable for the collection, storage, and retrieval of cardholder data. This solution automates deployment and configuration of Azure resources for a common reference architecture, demonstrating ways in which customers can meet specific security and compliance requirements and serves as a foundation for customers to build and configure their own solutions on Azure. The solution implements a subset of requirements from PCI DSS 3.2. For more information about PCI DSS 3.2 requirements and this solution, see the [compliance documentation](#) section.

This Azure Security and Compliance Blueprint Automation automatically deploys a PaaS web application reference architecture with pre-configured security controls to help customers achieve compliance with PCI DSS 3.2 requirements. The solution consists of Azure Resource Manager templates and PowerShell scripts that guide resource deployment and configuration.

This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment. Deploying an application into this environment without modification is not sufficient to completely meet the requirements of PCI DSS 3.2. Please note the following:

- This architecture provides a baseline to help customers use Azure in a PCI DSS 3.2 compliant manner.
- Customers are responsible for conducting appropriate security and compliance assessment of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Achieving PCI DSS-compliance requires that an accredited Qualified Security Assessor (QSA) certify a production customer solution. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Click [here](#) for deployment instructions.

Architecture diagram and components

This Azure Security and Compliance Blueprint Automation deploys a reference architecture for a PaaS web application with an Azure SQL Database backend. The web application is hosted in an isolated Azure App Service Environment, which is a private, dedicated environment in an Azure datacenter. The environment load balances traffic for the web application across virtual machines managed by Azure. This architecture also includes network security groups, an Application Gateway, Azure DNS, and Load Balancer.

For enhanced analytics and reporting, Azure SQL databases can be configured with columnstore indexes. Azure SQL databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

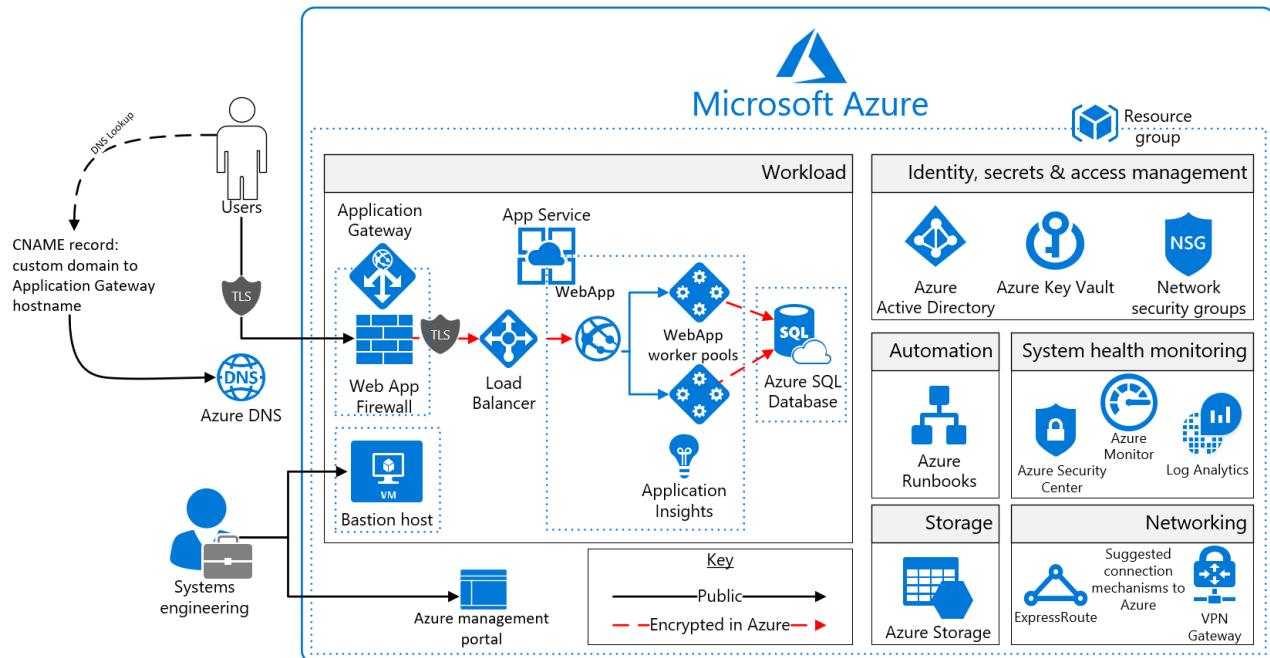
The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and stored again as three copies within that datacenter, preventing an adverse event at the

customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

Azure SQL Database is commonly managed through SQL Server Management Studio, which runs from a local machine configured to access the Azure SQL Database via a secure VPN or ExpressRoute connection.

Furthermore, Application Insights provides real time application performance management and analytics through Log Analytics. **Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment Architecture](#) section.

- App Service Environment v2
- Application Gateway
 - (1) web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP 3.0
 - Listener port: 443
- Application Insights
- Azure Active Directory
- Azure Automation
- Azure DNS
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Virtual Network

- (1) /16 Network
- (4) /24 Networks
- (4) Network Security Groups
- Azure Web App

Deployment architecture

The following section details the deployment and implementation elements.

Azure Resource Manager: [Azure Resource Manager](#) enables customers to work with the resources in the solution as a group. Customers can deploy, update, or delete all the resources for the solution in a single, coordinated operation. Customers use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help customers manage their resources after deployment.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

App Service Environment v2: The Azure App Service Environment is an App Service feature that provides a fully isolated and dedicated environment for securely running App Service applications at a high scale. This isolation feature is required to meet PCI compliance requirements.

App Service Environments are isolated to only run a single customer's applications and are always deployed into a virtual network. This isolation feature enables the reference architecture to have complete tenant isolation, removing it from Azure's multi-tenant environment prohibiting those multi-tenants from enumerating the deployed App Service Environment resources. Customers have fine-grained control over both inbound and outbound application network traffic, and applications can establish high-speed secure connections over virtual networks to on-premises corporate resources. Customers can "auto-scale" with App Service Environment based on load metrics, available budget, or a defined schedule.

Utilize App Service Environments for the following controls/configurations:

- Host inside a secured Azure Virtual Network and network security rules
- Self-signed ILB certificate for HTTPS communication
- [Internal Load Balancing mode](#)
- Disable [TLS 1.0](#)
- Change [TLS Cipher](#)
- Control [inbound traffic N/W ports](#)
- [Web application firewall – restrict data](#)
- Allow [Azure SQL Database traffic](#)

Azure Web App: [Azure Web Apps](#) enables customers to build and host web applications in the programming language of their choice without managing infrastructure. It offers auto-scaling and high availability, supports both

Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repo.

Virtual Network

The architecture defines a private Virtual Network with an address space of 10.200.0.0/16.

Network security groups: [Network security groups](#) contain Access Control Lists (ACLs) that allow or deny traffic within a Virtual Network. Network security groups can be used to secure traffic at a subnet or individual VM level. The following Network security groups exist:

- 1 Network security group for Application Gateway
- 1 Network security group for App Service Environment
- 1 Network security group for Azure SQL Database
- 1 network Security Group for bastion host

Each of the Network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each Network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [Network security group's diagnostics](#)

Subnets: Each subnet is associated with its corresponding Network security group.

Azure DNS: The Domain Name System, or DNS, is responsible for translating (or resolving) a website or service name to its IP address. [Azure DNS](#) is a hosting service for DNS domains that provides name resolution using Azure infrastructure. By hosting domains in Azure, users can manage DNS records using the same credentials, APIs, tools, and billing as other Azure services. Azure DNS also supports private DNS domains.

Azure Load Balancer: [Azure Load Balancer](#) allows customers to scale their applications and create high availability for services. Load Balancer supports inbound as well as outbound scenarios, and provides low latency, high throughput, and scales up to millions of flows for all TCP and UDP applications.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard cardholder data in support of organizational security commitments and compliance requirements defined by PCI DSS 3.2.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.

- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to cardholder data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [Integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing cardholder data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information such as cardholder data. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is an HSM Protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities using an Azure Application Gateway with a web application firewall configured, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type within Log Analytics workspaces, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- **Active Directory Assessment:** The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- **SQL Assessment:** The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- **Agent Health:** The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- **Activity Log Analytics:** The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

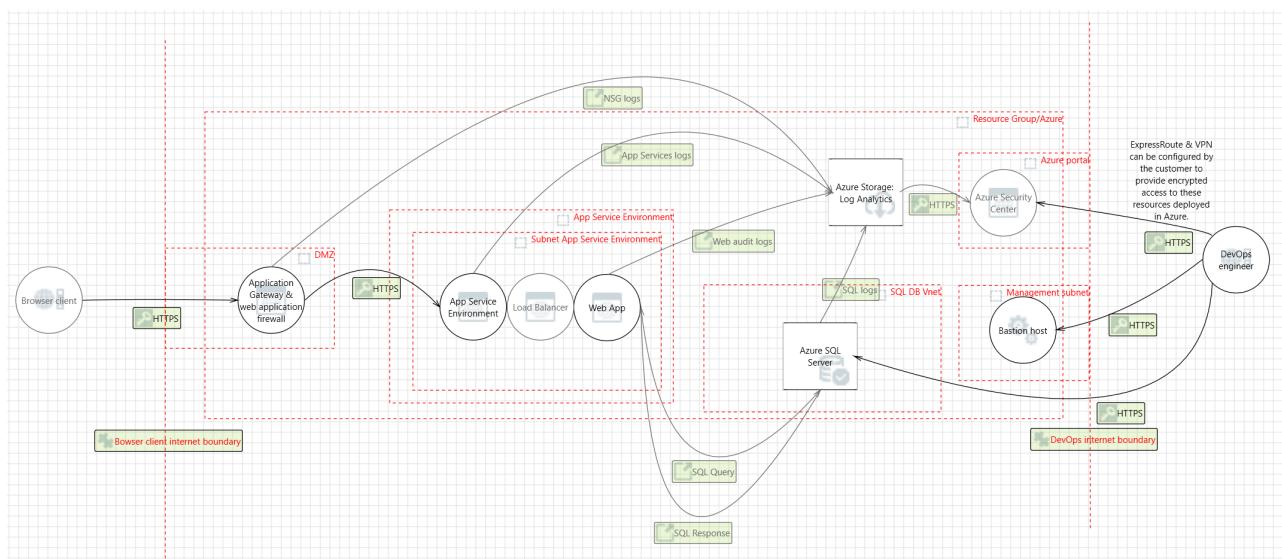
Azure Automation: Azure Automation stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation Change Tracking solution enables customers to easily identify changes in the environment.

Azure Monitor: Azure Monitor helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Application Insights: Application Insights is an extensible Application Performance Management service for web developers on multiple platforms. Application Insights detects performance anomalies and customers can use it to monitor the live web application. It includes powerful analytics tools to help customers diagnose issues and to understand what users actually do with their app. It's designed to help customers continuously improve performance and usability.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – PCI DSS Customer Responsibility Matrix](#) lists controller and processor responsibilities for all PCI DSS 3.2 requirements.

The [Azure Security and Compliance Blueprint – PCI DSS PaaS Web Application Implementation Matrix](#) provides information on which PCI DSS 3.2 requirements are addressed by the PaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered article.

Deploy this solution

This Azure Security and Compliance Blueprint Automation is comprised of JSON configuration files and PowerShell scripts that are handled by Azure Resource Manager's API service to deploy resources within Azure. Detailed deployment instructions are available [here](#).

Quickstart

1. Clone or download [this GitHub repository](#) to your local workstation.
2. Review 0-Setup-AdministrativeAccountAndPermission.md and run the provided commands.
3. Deploy a test solution with Contoso sample data or pilot an initial production environment.
 - 1A-ContosoWebStoreDemoAzureResources.ps1

- This script deploys Azure resources for a demonstration of a webstore using Contoso sample data.
- 1-DeployAndConfigureAzureResources.ps1
 - This script deploys the Azure resources needed for supporting a production environment for a customer-owned web application. This environment should be further customized by the customer based on organizational requirements.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this PaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure Virtual Network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-Site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Trusted Internet Connections guidance

12/6/2018 • 16 minutes to read • [Edit Online](#)

This article covers how government agencies can use Microsoft Azure Government to help achieve compliance with the Trusted Internet Connections (TIC) initiative. The article describes how to use Azure Government across Azure infrastructure as a service (IaaS) and Azure platform as a service (PaaS) offerings.

Trusted Internet Connections overview

The purpose of the TIC initiative is to optimize and standardize the security of individual external network connections that are used by federal agencies. The policy is outlined in the [Office of Management and Budget \(OMB\) Memorandum M-08-05](#).

In November 2007, the OMB established the TIC program to improve federal network perimeter security and incident response functions. TIC is designed to perform network analysis of all inbound and outbound .gov traffic to identify specific signatures and pattern-based data. TIC uncovers behavioral anomalies, such as botnet activity. Agencies are mandated to consolidate their external network connections and route all traffic through intrusion detection and prevention devices known as EINSTEIN. The devices are hosted at a limited number of network endpoints, which are referred to as *trusted internet connections*.

The objective of TIC is for agencies to know:

- Who is on my network (authorized or unauthorized)?
- When is my network accessed and why?
- What resources are accessed?

All agency external connections must currently route through an OMB-approved TIC. Federal agencies are required to participate in the TIC program as a TIC Access Provider (TICAP), or by contracting services with one of the major tier 1 internet service providers. These providers are referred to as Managed Trusted Internet Protocol Service (MTIPS) providers. TIC includes mandatory critical capabilities that are performed by the agency and MTIPS provider. In the current version of TIC, the EINSTEIN version 2 intrusion detection and EINSTEIN version 3 accelerated (3A) intrusion prevention devices are deployed at each TICAP and MTIPS. The agency establishes a "Memorandum of Understanding" with the Department of Homeland Security (DHS) to deploy EINSTEIN capabilities to federal systems.

As part of its responsibility to protect the .gov network, DHS requires the raw data feeds of agency net flow data to correlate incidents across the federal enterprise and perform analyses by using specialized tools. DHS routers provide the ability to collect IP network traffic as it enters or exits an interface. Network administrators can analyze the net flow data to determine the source and destination of traffic, the class of service, and so on. Net flow data is considered to be "non-content data" similar to the header, source IP, destination IP, and so on. Non-content data allows DHS to learn about the content: who was doing what and for how long.

The initiative also includes security policies, guidelines, and frameworks that assume on-premises infrastructure. As government agencies move to the cloud to achieve cost savings, operational efficiency, and innovation, the implementation requirements of TIC can slow down network traffic. The speed and agility with which government users can access their cloud-based data is limited as a result.

Azure networking options

There are three main options to connect to Azure services:

- Direct internet connection: Connect to Azure services directly through an open internet connection. The

medium and the connection are public. Application and transport level encryption are relied upon to ensure privacy. Bandwidth is limited by a site's connectivity to the internet. Use more than one active provider to ensure resiliency.

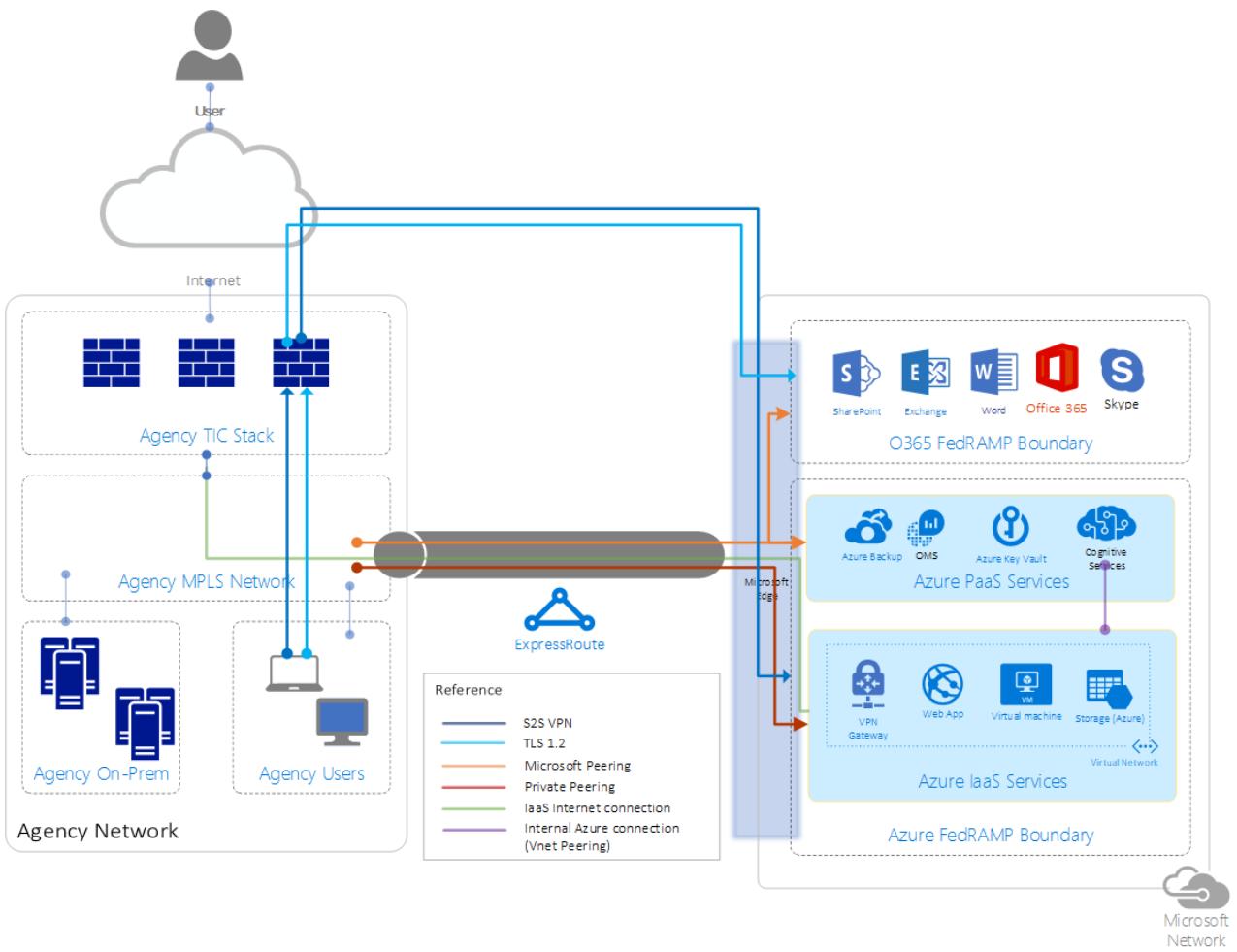
- Virtual private network (VPN): Connect to your Azure virtual network privately by using a VPN gateway. The medium is public because it traverses a site's standard internet connection, but the connection is encrypted in a tunnel to ensure privacy. Bandwidth is limited depending on the VPN devices and the configuration chosen. Azure point-to-site connections are typically limited to 100 Mbps and site-to-site connections are limited to 1.25 Gbps.
- Azure ExpressRoute: ExpressRoute is a direct connection to Microsoft services. Because connectivity is through an isolated fiber channel, the connection can be public or private depending on the configuration that's used. The bandwidth is typically limited to a maximum of 10 Gbps.

There are several ways to meet the TIC Appendix H (Cloud Considerations) requirements, as specified in the Department of Homeland Security's, "Trusted Internet Connections (TIC) Reference Architecture Document, Version 2.0." In this article, DHS TIC guidance is referred to as **TIC 2.0**.

To enable the connection from the **Department or Agency (D/A)** to Azure or Office 365, without routing traffic through the D/A TIC, the D/A must use an encrypted tunnel or a dedicated connection to the cloud service provider (CSP). The CSP services can ensure connectivity to the D/A cloud assets isn't offered to the public internet for direct agency personnel access.

Office 365 is compliant with TIC 2.0 Appendix H by using either ExpressRoute with [Microsoft Peering](#) enabled or an internet connection that encrypts all traffic by using TLS 1.2. D/A end users on the D/A network can connect via their agency network and TIC infrastructure through the internet. All remote internet access to Office 365 is blocked and routes through the agency. The D/A can also connect to Office 365 over an ExpressRoute connection with Microsoft Peering (a type of public peering) enabled.

For Azure only, the second option (VPN) and third option (ExpressRoute) can meet these requirements when they're used in conjunction with services that limit access to the internet.



Azure infrastructure as a service offerings

Compliance with TIC policy by using Azure IaaS is relatively simple because Azure customers manage their own virtual network routing.

The main requirement to help assure compliance with the TIC reference architecture is to ensure your virtual network is a private extension of the D/A network. To be a *private* extension, the policy requires no traffic leave your network except via the on-premises TIC network connection. This process is known as *force tunneling*. For TIC compliance, the process routes all traffic from any system in the CSP environment through an on-premises gateway on an organization's network out to the internet through the TIC.

Azure IaaS TIC compliance is divided into two major steps:

- Step 1: Configuration.
- Step 2: Auditing.

Azure IaaS TIC compliance: Configuration

To configure a TIC-compliant architecture with Azure, you must first prevent direct internet access to your virtual network and then force internet traffic through the on-premises network.

Prevent direct internet access

Azure IaaS networking is conducted via virtual networks that are composed of subnets to which the network interface controllers (NICs) of virtual machines are associated.

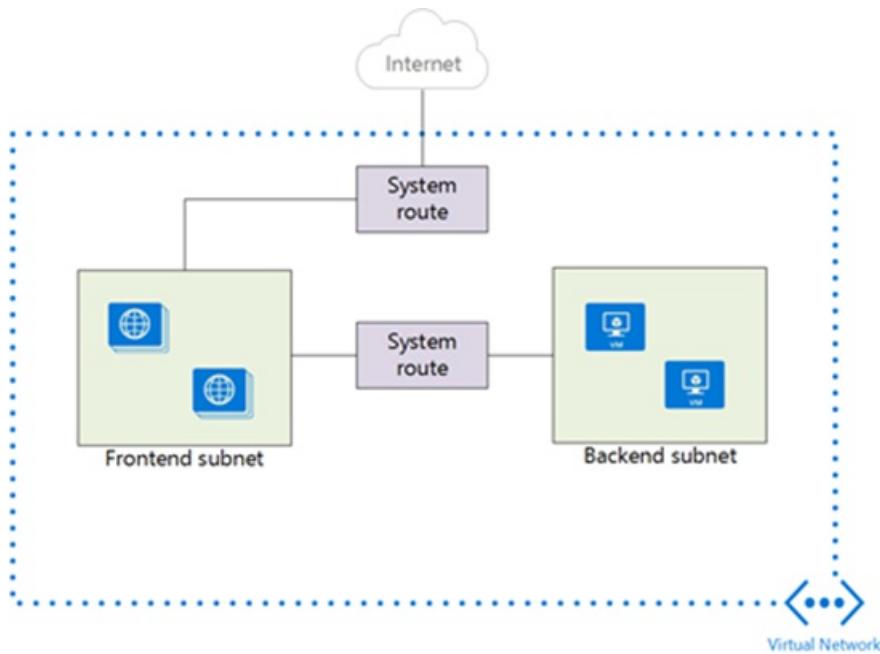
The simplest scenario to support TIC compliance is to assure that a virtual machine, or a collection of virtual machines, can't connect to any external resources. Assure the disconnection from external networks by using network security groups (NSGs). Use NSGs to control traffic to one or more NICs or subnets in your virtual network. An NSG contains access control rules that allow or deny traffic based on traffic direction, protocol, source address and port, and destination address and port. You can change the rules of an NSG at any time, and changes

are applied to all associated instances. To learn more about how to create an NSG, see [Filter network traffic with a network security group](#).

Force internet traffic through an on-premises network

Azure automatically creates system routes and assigns the routes to each subnet in a virtual network. You can't create or remove system routes, but you can override some system routes with custom routes. Azure creates default system routes for each subnet. Azure adds optional default routes to specific subnets, or every subnet, when you use specific Azure capabilities. This type of routing ensures:

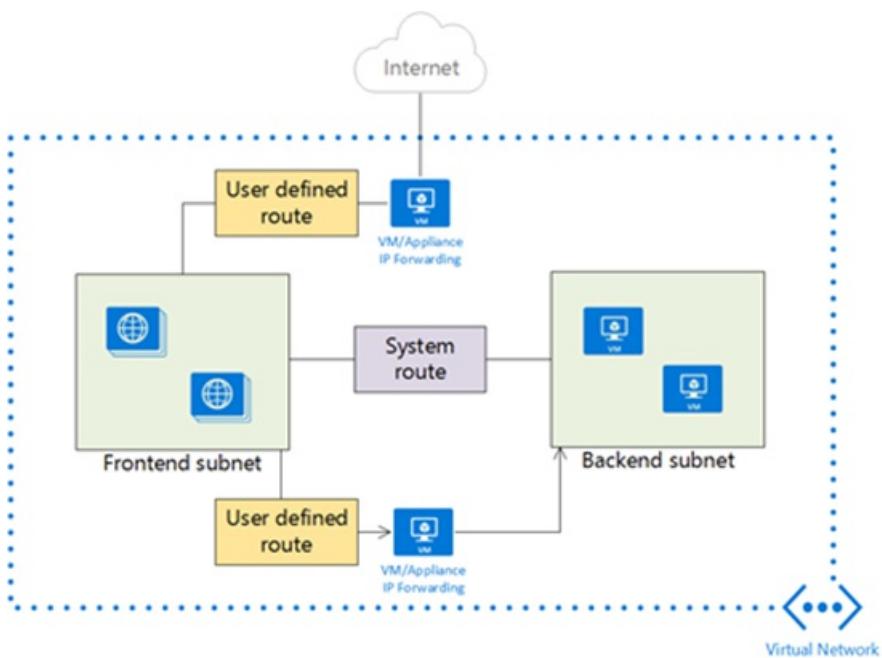
- Traffic that's destined within the virtual network stays within the virtual network.
- IANA-designated private address spaces like 10.0.0.0/8 are dropped, unless they're included in the virtual network address space.
- "Last-resort" routing of 0.0.0.0/0 to the virtual network internet endpoint.



All traffic that leaves the virtual network needs to route through the on-premises connection, to ensure that all traffic traverses the D/A TIC. You create custom routes by creating user-defined routes, or by exchanging Border Gateway Protocol (BGP) routes between your on-premises network gateway and an Azure VPN gateway. For more information about user-defined routes, see [Virtual network traffic routing: User-defined routes](#). For more information about the BGP, see [Virtual network traffic routing: Border Gateway Protocol](#).

Add user-defined routes

If you use a route-based virtual network gateway, you can force tunneling in Azure. Add a user-defined route (UDR) that sets 0.0.0.0/0 traffic to route to a **next hop** of your virtual network gateway. Azure prioritizes user-defined routes over system-defined routes. All non-virtual network traffic is sent to your virtual network gateway, which can then route the traffic to on-premises. After you define the UDR, associate the route with existing subnets or new subnets within all virtual networks in your Azure environment.



Use the Border Gateway Protocol

If you use ExpressRoute or a BGP-enabled virtual network gateway, BGP is the preferred mechanism for advertising routes. For a BGP advertised route of 0.0.0.0/0, ExpressRoute and BGP-aware virtual network gateways ensure the default route applies to all subnets within your virtual networks.

Azure IaaS TIC compliance: Auditing

Azure offers several ways to audit TIC compliance.

View effective routes

Confirm that your default route is propagated by observing the *effective routes* for a particular virtual machine, a specific NIC, or a user-defined route table in [the Azure portal](#) or in [Azure PowerShell](#). The **Effective Routes** show the relevant user-defined routes, BGP advertised routes, and system routes that apply to the relevant entity, as shown in the following figure:

Effective routes			
SOURCE	STATE	ADDRESS PREFIXES	NEXT HOP TYPE
Default	Active	10.9.0.0/16	VnetLocal
Default	Active	10.10.0.0/16	VNetPeering
Default	Active	0.0.0.0/0	Internet

NOTE

You can't view the effective routes for a NIC, unless the NIC is associated with a running virtual machine.

Use Azure Network Watcher

Azure Network Watcher offers several tools to audit TIC compliance. For more information, see [this overview about Network Watcher](#).

Capture network security group flow logs

Use Network Watcher to capture network flow logs that indicate the metadata that surrounds IP traffic. The network flow logs contain the source and destination addresses of targets, and other data. You can use this data

with logs from the virtual network gateway, on-premises edge devices, or the TIC, to monitor that all traffic routes on-premises.

Azure platform as a service offerings

Azure PaaS services, such as Azure Storage, are accessible through an internet-reachable URL. Anyone with approved credentials can access the resource, such as a storage account, from any location without traversing a TIC. For this reason, many government customers incorrectly conclude that Azure PaaS services aren't compliant with TIC policies. Many Azure PaaS services can be compliant with TIC policy. A service is compliant when the architecture is the same as the TIC-compliant IaaS environment ([as previously described](#)) and the service is attached to an Azure virtual network.

When Azure PaaS services are integrated with a virtual network, the service is privately accessible from that virtual network. You can apply custom routing for 0.0.0.0/0 via user-defined routes or BGP. Custom routing ensures that all internet-bound traffic routes on-premises to traverse the TIC. Integrate Azure services into virtual networks by using the following patterns:

- **Deploy a dedicated instance of a service:** An increasing number of PaaS services are deployable as dedicated instances with virtual network-attached endpoints. You can deploy an App Service Environment for PowerApps in "Isolated" mode to allow the network endpoint to be constrained to a virtual network. The App Service Environment can then host many Azure PaaS services, such as Azure Web Apps, Azure API Management, and Azure Functions.
- **Use virtual network service endpoints:** An increasing number of PaaS services allow the option to move their endpoint to a virtual network private IP instead of a public address.

Services that support deployment of dedicated instances into a virtual network or use of service endpoints, as of May 2018, are listed in the following tables.

NOTE

The availability status corresponds to the Azure services that are commercially available. The availability status for Azure services in Azure Government is pending.

Support for service endpoints

SERVICE	AVAILABILITY
Azure Key Vault	Private preview
Azure Cosmos DB	Private preview
Identity services	Private preview
Azure Data Lake	Private preview
Azure Database for PostgreSQL	Private preview
Azure Database for MySQL	Private preview
Azure SQL Data Warehouse	Public preview
Azure SQL Database	General availability (GA)

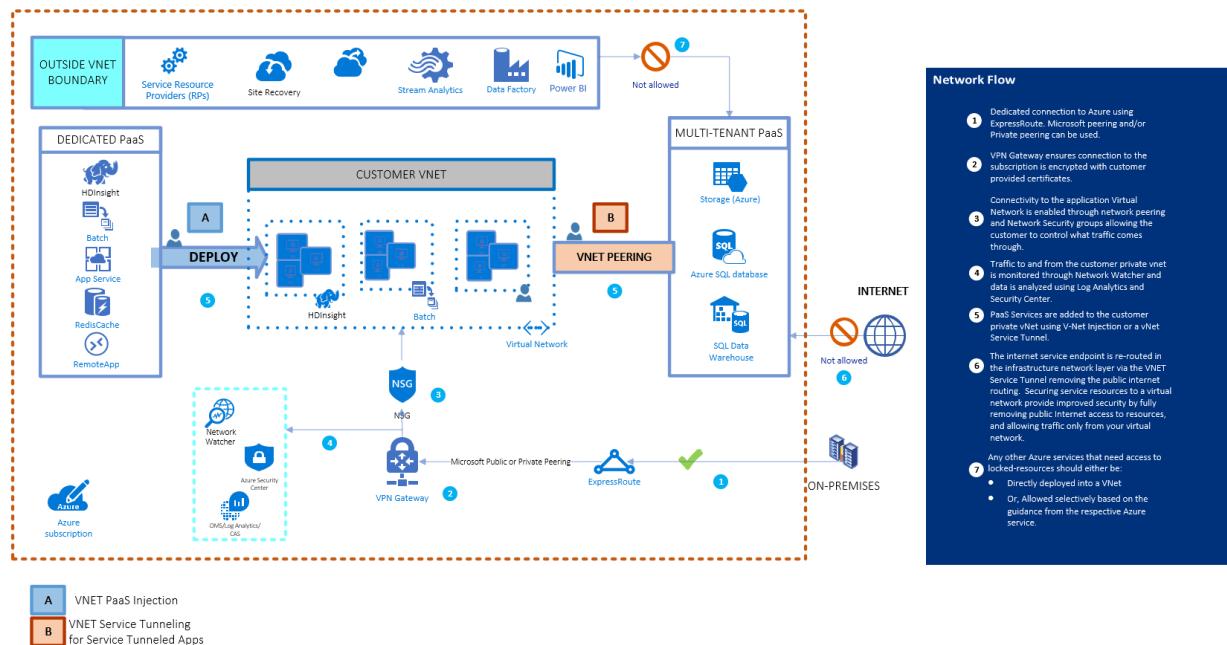
SERVICE	AVAILABILITY
Azure Storage	GA

Support for virtual network injection

SERVICE	AVAILABILITY
Azure SQL Database Managed Instance	Public preview
Azure Kubernetes Service (AKS)	Public preview
Azure Service Fabric	GA
Azure API Management	GA
Azure Active Directory	GA
Azure Batch	GA
App Service Environment	GA
Azure Cache for Redis	GA
Azure HDInsight	GA
Virtual machine scale set	GA
Azure Cloud Services	GA

Virtual network integration details

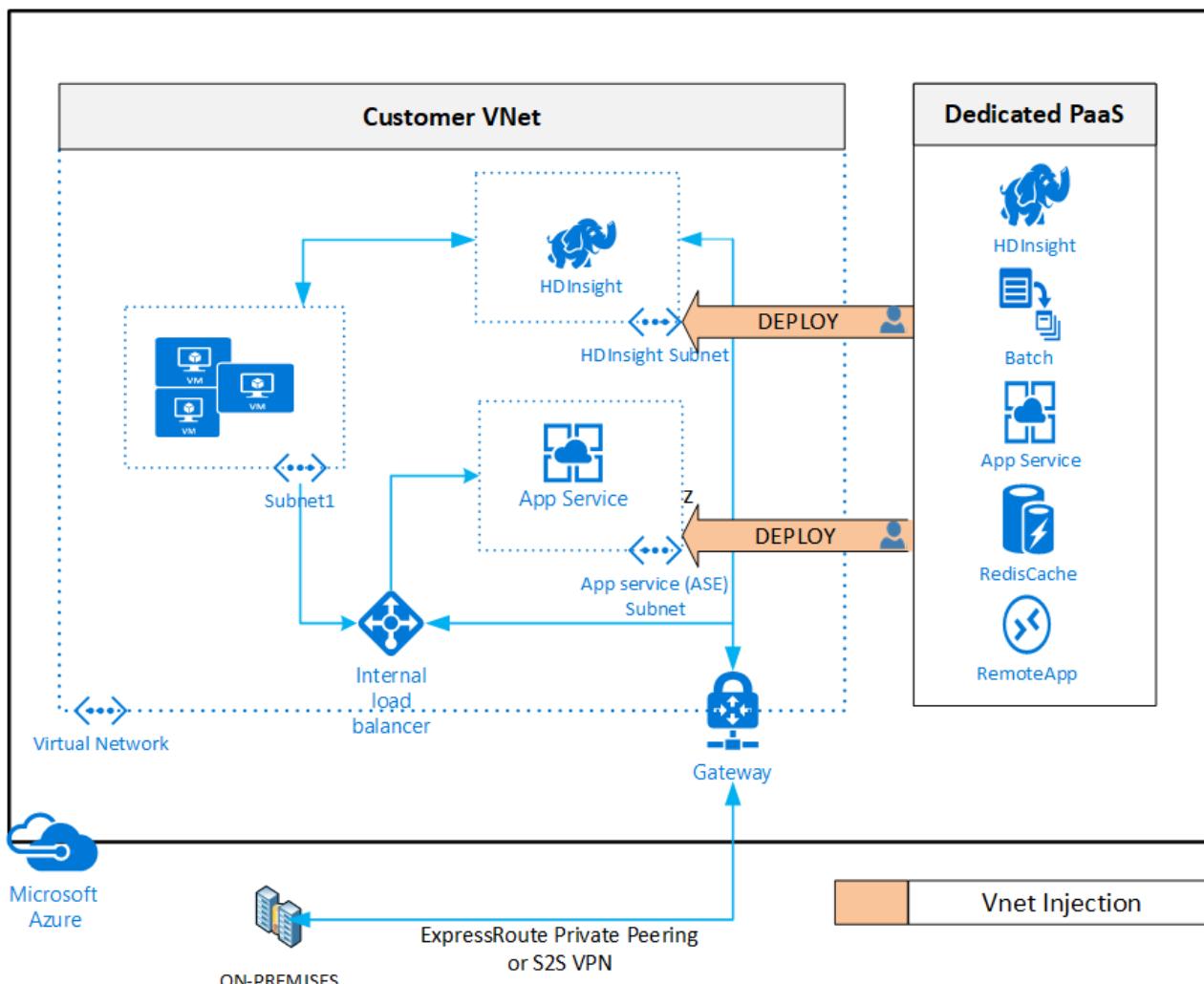
The following diagram shows the general network flow for access to PaaS services. Access is shown from both virtual network injection and virtual network service tunneling. For more information about network service gateways, virtual networks, and service tags, see [Network and application security groups: Service tags](#).



1. A private connection is made to Azure by using ExpressRoute. ExpressRoute private peering with forced tunneling is used to force all customer virtual network traffic over ExpressRoute and back to on-premises. Microsoft Peering isn't required.
2. Azure VPN Gateway, when used in conjunction with ExpressRoute and Microsoft Peering, can overlay end-to-end IPSec encryption between the customer virtual network and the on-premises edge.
3. Network connectivity to the customer virtual network is controlled by using NSGs that allow customers to permit/deny based on IP, port, and protocol.
4. The customer virtual network extends to the PaaS service by creating a service endpoint for the customer's service.
5. The PaaS service endpoint is secured to **default deny all** and to only allow access from specified subnets within the customer virtual network. The default denies also includes connections that originate from the internet.
6. Other Azure services that need to access resources within the customer virtual network should either be:
 - Directly deployed into the virtual network.
 - Selectively allowed, based on the guidance from the respective Azure service.

Option A: Deploy a dedicated instance of a service (virtual network injection)

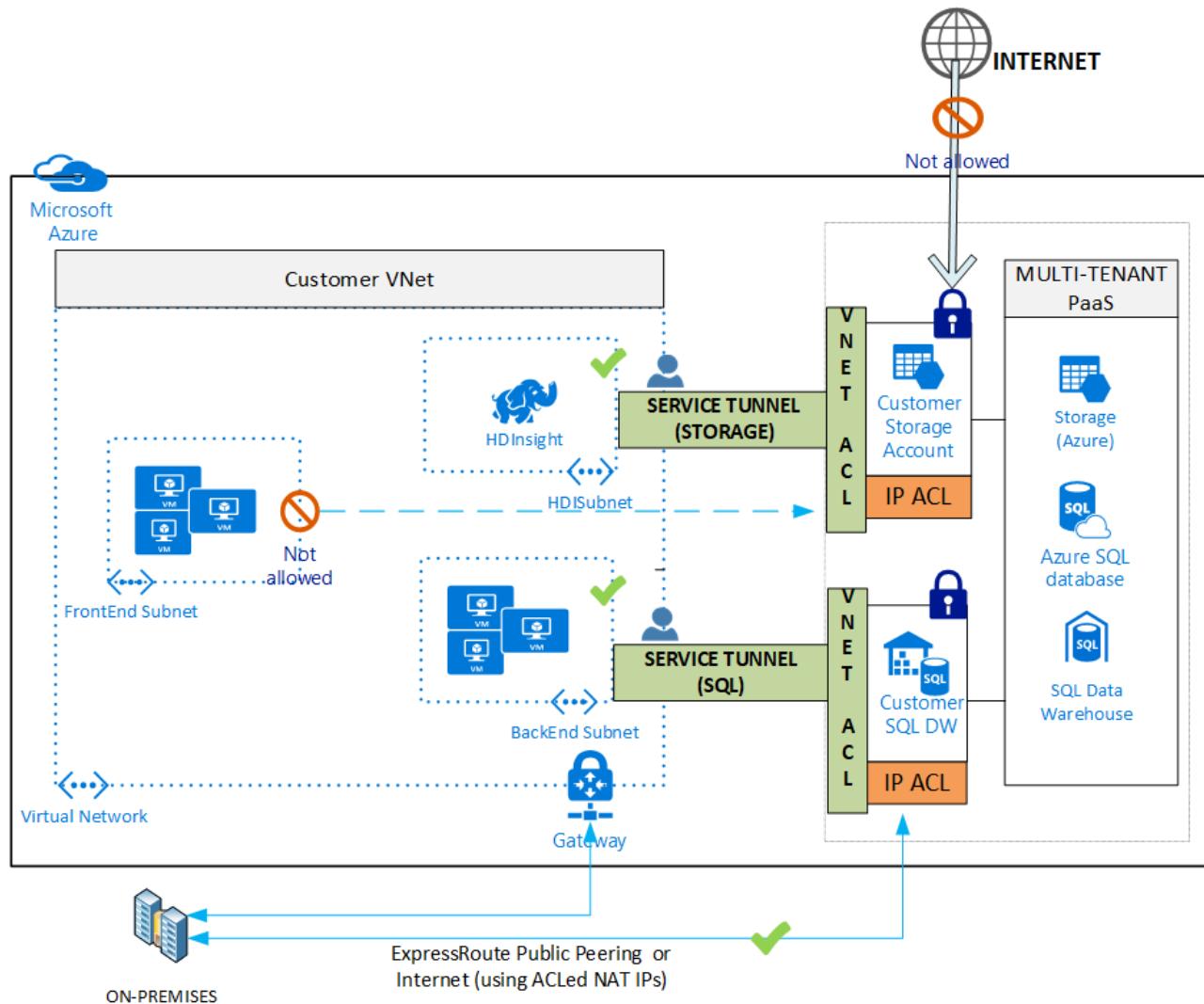
Virtual network injection enables customers to selectively deploy dedicated instances of a given Azure service, such as HDInsight, into their own virtual network. Service instances are deployed into a dedicated subnet in a customer's virtual network. Virtual network injection allows access to service resources through the non-internet routable addresses. On-premises instances use ExpressRoute or a site-to-site VPN to directly access service instances via virtual network address space, instead of opening a firewall to public internet address space. When a dedicated instance is attached to an endpoint, you can use the same strategies as for IaaS TIC compliance. Default routing ensures internet-bound traffic is redirected to a virtual network gateway that's bound for on-premises. You can further control inbound and outbound access through NSGs for the given subnet.



Option B: Use virtual network service endpoints (service tunnel)

An increasing number of Azure multitenant services offer "service endpoints." Service endpoints are an alternate method for integrating to Azure virtual networks. Virtual network service endpoints extend your virtual network IP address space and the identity of your virtual network to the service over a direct connection. Traffic from the virtual network to the Azure service always stays within the Azure backbone network.

After you enable a service endpoint for a service, use policies exposed by the service to restrict connections for the service to that virtual network. Access checks are enforced in the platform by the Azure service. Access to a locked resource is granted only if the request originates from the allowed virtual network or subnet, or from the two IPs that are used to identify your on-premises traffic if you use ExpressRoute. Use this method to effectively prevent inbound/outbound traffic from directly leaving the PaaS service.



Cloud-native tools for network situational awareness

Azure provides cloud-native tools to help ensure that you have the situational awareness that's required to understand the traffic flows of your network. The tools aren't required for compliance with TIC policy. The tools can vastly improve your security capabilities.

Azure Policy

[Azure Policy](#) is an Azure service that provides your organization with better ability to audit and enforce compliance initiatives. Azure Policy is currently available in public preview in Azure services that are commercially available. Azure Policy isn't yet available in Azure Government. Customers can plan and test their Azure Policy rules now to assure future TIC compliance.

Azure Policy is targeted at the subscription level. The service provides a centralized interface where you can perform compliance tasks, including:

- Manage initiatives
- Configure policy definitions
- Audit compliance
- Enforce compliance
- Manage exceptions

Along with many built-in definitions, administrators can define their own custom definitions by using simple JSON templates. Microsoft recommends the prioritization of auditing over enforcement, where possible.

The following sample policies can be used for TIC compliance scenarios:

POLICY	SAMPLE SCENARIO	TEMPLATE
Enforce user-defined route table.	Ensure that the default route on all virtual networks points to an approved virtual network gateway for routing to on-premises.	Get started with this template .
Audit if Network Watcher isn't enabled for a region.	Ensure that Network Watcher is enabled for all used regions.	Get started with this template .
NSG x on every subnet.	Ensure that an NSG (or a set of approved NSGs) with internet traffic blocked is applied to all subnets in every virtual network.	Get started with this template .
NSG x on every NIC.	Ensure that an NSG with internet traffic blocked is applied to all NICs on all virtual machines.	Get started with this template .
Use an approved virtual network for virtual machine network interfaces.	Ensure that all NICs are on an approved virtual network.	Get started with this template .
Allowed locations.	Ensure that all resources are deployed to regions with compliant virtual networks and Network Watcher configuration.	Get started with this template .
Not allowed resource types, such as PublicIPs .	Prohibit the deployment of resource types that don't have a compliance plan. Use this policy to prohibit the deployment of public IP address resources. While NSG rules can be used to effectively block inbound internet traffic, preventing the use of public IPs further reduces the attack surface.	Get started with this template .

Network Watcher traffic analytics

Network Watcher [traffic analytics](#) consume flow log data and other logs to provide a high-level overview of network traffic. The data is useful for auditing TIC compliance and to identify trouble spots. You can use the high-level dashboard to rapidly screen the virtual machines that are communicating with the internet and get a focused list for TIC routing.

Network Watcher - Traffic Analytics

Microsoft

Search (Ctrl+F)

Refresh Send us your feedback FAQ

Total flows Inbound Outbound

2.63M 294K 2.5M 7.5M 2.5M 2.5M

4K 294K 2.5M 9 2.5M

3.9K 61 280K 9 2.5M

This tabular representation of network traffic flow distribution is "not to scale".

Do more Launch Log Search query Documentation

YOUR ENVIRONMENT
Across Azure regions, virtual networks, resources and subnetworks

Deployed Azure regions
8 of 12 total

Action	Count
Inactive	1
Traffic Analytics enabled	5
Allowed malicious	2

Virtual networks
12 total

Action	Count
Active	7
Inactive	5
Allowed malicious	2

External connections
On premise: 0, Azure regions: 1, Public IPs: 4

Enabled NSGs*
53 of 56

Talking to Internet
Ports receiving traffic from Internet: 8, Hosts sending traffic to Internet: 12

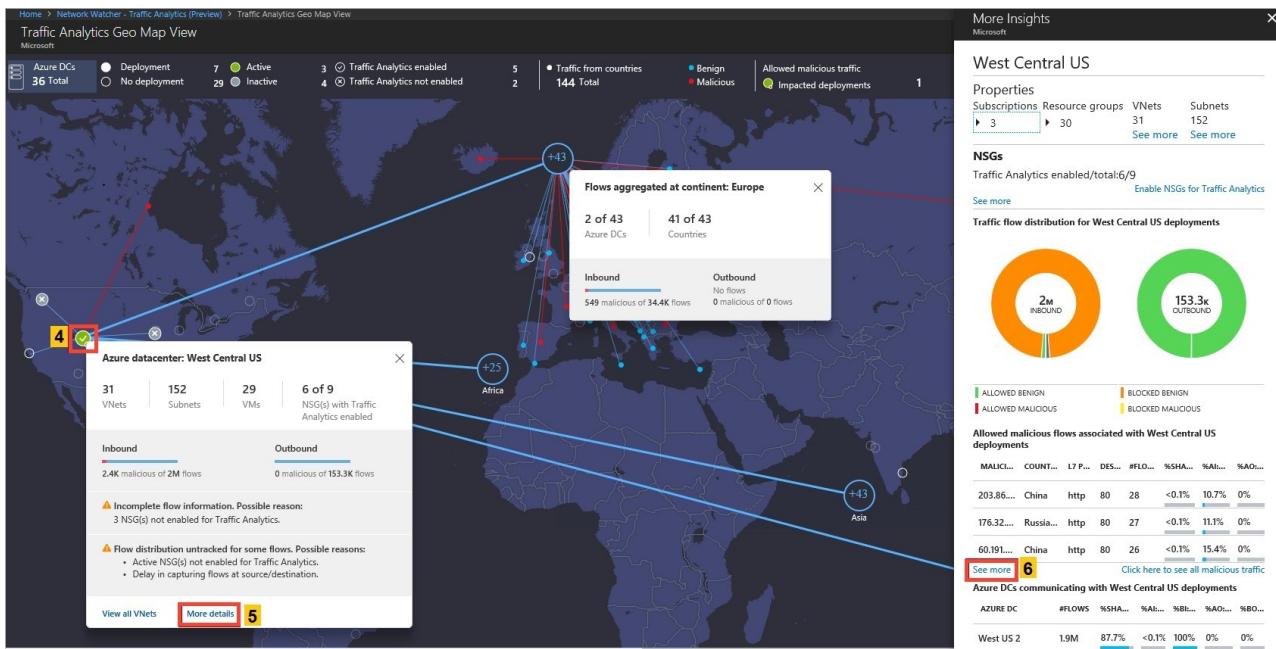
*Enable all NSGs to view richer data

Virtual subnetworks
36 total

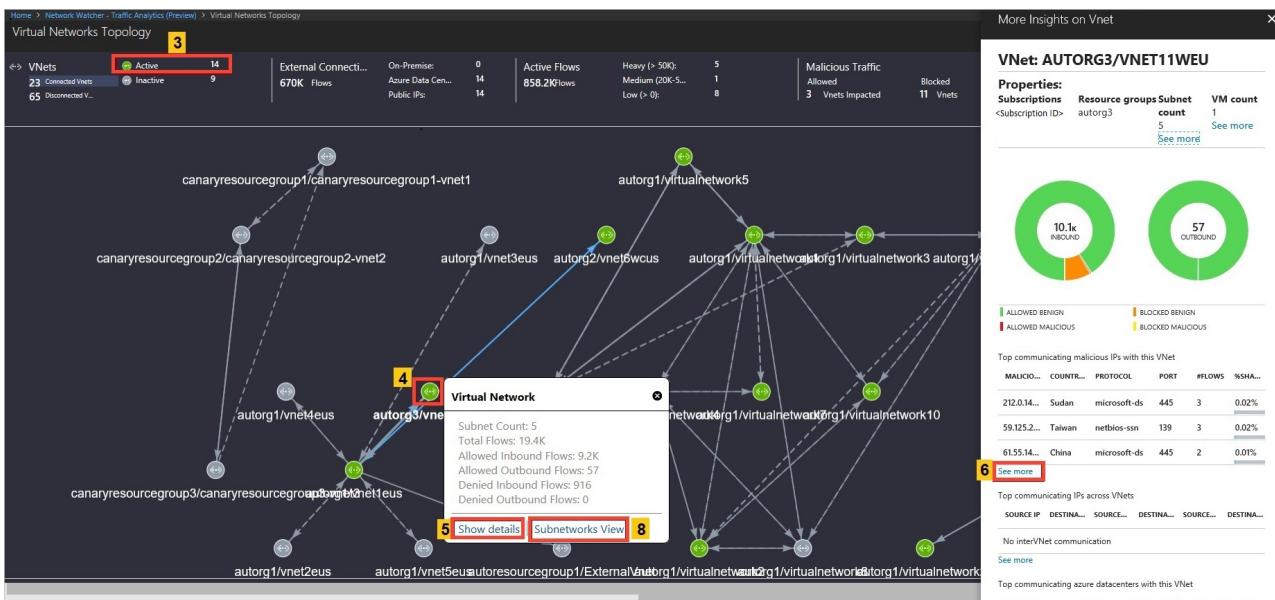
Action	Count
Inactive	8
Allowed malicious	2

TRAFFIC DISTRIBUTION
View analytics of traffic flows across host, subnet and VNet

Use the **Geo Map** to quickly identify the probable physical destinations of internet traffic for your virtual machines. You can identify and triage suspicious locations or pattern changes:



Use the **Virtual Networks Topology** to rapidly survey existing virtual networks:



Network Watcher next hop tests

Networks in regions that are monitored by Network Watcher can conduct next hop tests. In the Azure portal, you can enter a source and destination for a sample network flow for Network Watcher to resolve the next hop destination. Run this test against virtual machines and sample internet addresses to ensure the next hop destination is the expected network virtual gateway.

The screenshot shows the Azure Network Watcher - Next hop interface. On the left, there's a sidebar with various navigation options: Overview, MONITORING, NETWORK DIAGNOSTIC TOOLS (IP flow verify, Next hop selected, Security group view, Packet capture), METRICS, LOGS (NSG flow logs, Diagnostic logs). The main panel is titled "Specify a target virtual machine and destination IP address to view the next hop." It includes fields for Subscription (Microsoft), Resource group (ContosoAppGateway), Virtual machine (webtestvm-backend-a76qj4rjyab-1-aehrwnnchvwx4), Network interface (webtestnic-backend-a76qj4rjyab-1-aehrwnnchvwx4), Source IP address (10.0.4), and Destination IP address. A large blue "Next hop" button is at the bottom.

Conclusions

You can easily configure access for Microsoft Azure, Office 365, and Dynamics 365 to help comply with TIC 2.0 Appendix H guidance, as written and defined May 2018. Microsoft recognizes that the TIC guidance is subject to change. Microsoft endeavors to help customers meet the guidance in a timely manner as new guidance is released.

Appendix: Trusted Internet Connections patterns for common

workloads

CATEGORY	WORKLOAD	IAAS	DEDICATED PAAS / VIRTUAL NETWORK INJECTION	SERVICE ENDPOINTS
Compute	Azure Linux virtual machines	Yes		
Compute	Azure Windows virtual machines	Yes		
Compute	Virtual machine scale sets	Yes		
Compute	Azure Functions		App Service Environment	
Web and mobile	Internal web application		App Service Environment	
Web and mobile	Internal mobile application		App Service Environment	
Web and mobile	API applications		App Service Environment	
Containers	Azure Container Service			Yes
Containers	Azure Kubernetes Service (AKS) *			Yes
Database	Azure SQL Database		Azure SQL Database Managed Instance *	Azure SQL
Database	Azure Database for MySQL			Yes
Database	Azure Database for PostgreSQL			Yes
Database	Azure SQL Data Warehouse			Yes
Database	Azure Cosmos DB			Yes
Database	Azure Cache for Redis		Yes	
Storage	Azure Blob storage	Yes		
Storage	Azure Files	Yes		
Storage	Azure Queue storage	Yes		

CATEGORY	WORKLOAD	IAAS	DEDICATED PAAS / VIRTUAL NETWORK INJECTION	SERVICE ENDPOINTS
Storage	Azure Table storage	Yes		
Storage	Azure Disk storage	Yes		

* Public preview in Azure Government, May 2018.

Azure Security and Compliance Blueprint: Data Analytics for UK NHS

9/12/2018 • 14 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides a reference architecture and guidance for a data analytics solution suitable for the collection, storage, analysis, and retrieval of healthcare data. This solution demonstrates ways in which customers can comply with guidance provided in the [Cloud Security Good Practice Guide](#) published by [NHS Digital](#), a partner of the United Kingdom's (UK) Department of Health and Social Care (DHSC). The Cloud Security Good Practice Guide is based on the 14 [Cloud Security Principles](#) published by the UK National Cyber Security Centre (NCSC).

This reference architecture, implementation guide, and threat model are intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment without additional configuration. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides an analytics platform upon which customers can build their own analytics tools. The reference architecture outlines a generic use case where customers input data either through bulk data imports by the SQL/data administrator or through operational data updates via an operational user. Both work streams incorporate Azure Functions for importing data into Azure SQL Database and all external connections require TLSv1.2. Azure Functions must be configured by the customer through the Azure portal to handle the import tasks unique to the customer's analytics requirements.

Azure offers a variety of reporting and analytics services for the customer; however, this solution incorporates Azure Analysis Services in conjunction with Azure SQL Database to rapidly browse through data and deliver faster results through smarter modeling of customer data. Azure Analytics Services is a form of machine learning intended to increase query speeds by discovering new relationships between datasets. Once the data has been trained through several statistical functions, up to 7 additional query pools (8 total including the customer server) can be synchronized with the same tabular models to spread query workload and reduce response times.

For enhanced analytics and reporting, Azure SQL databases can be configured with columnstore indexes. Both Azure Analytics Services and Azure SQL databases can be scaled up or down or shut off completely in response to customer usage. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

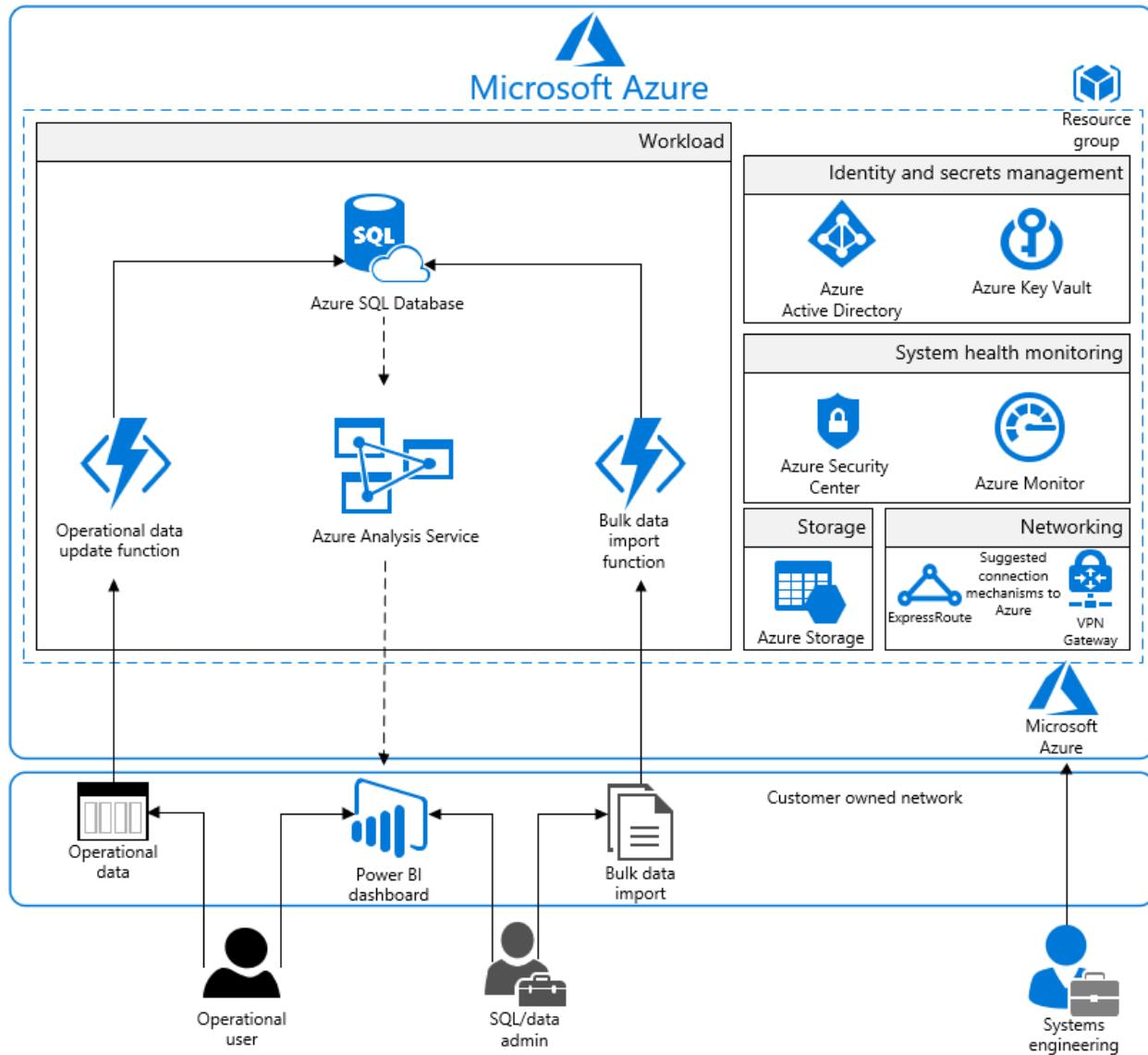
Once data is uploaded to the Azure SQL Database and trained by Azure Analysis Services, it is digested by both the operational user and SQL/data administrator with Power BI. Power BI displays data intuitively and pulls together information across multiple datasets to draw greater insight. Its high degree of adaptability and easy integration with Azure SQL Database ensures that customers can configure it to handle a wide array of scenarios as required by their business needs.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's chosen datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and again stored as three copies within that datacenter, preventing an adverse event at the

customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

Azure SQL Database is commonly managed through SQL Server Management Studio, which runs from a local machine configured to access the Azure SQL Database via a secure VPN or ExpressRoute connection. **Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture resource group.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment architecture](#) section.

- Azure Active Directory
- Azure Analysis Service
- Azure Automation
- Azure Data Catalog
- Azure Disk Encryption
- Azure Event Grid
- Azure Functions

- Azure Key Vault
- Azure Monitor
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Virtual Network
 - (1) /16 Network
 - (2) /24 Networks
 - (2) Network security groups
- Power BI Dashboard

Deployment architecture

The following section details the deployment and implementation elements.

Azure Functions: [Azure Functions](#) is a server-less compute service that enables users to run code on-demand without having to explicitly provision or manage infrastructure. Use Azure Functions to run a script or piece of code in response to a variety of events.

Azure Analysis Service: [Azure Analysis Service](#) provides enterprise data modeling and integration with Azure data platform services. Azure Analysis Service speeds up browsing through massive amounts of data by combining data from multiple sources into a single data model.

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual virtual machine level. The following network security groups exist:

- 1 network security group for Active Directory
- 1 network security group for the workload

Each of the network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [network security group's diagnostic logs](#)

Subnets: Each subnet is associated with its corresponding network security group.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by NHS Digital.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware

security module protected 2048-bit RSA Key.

- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

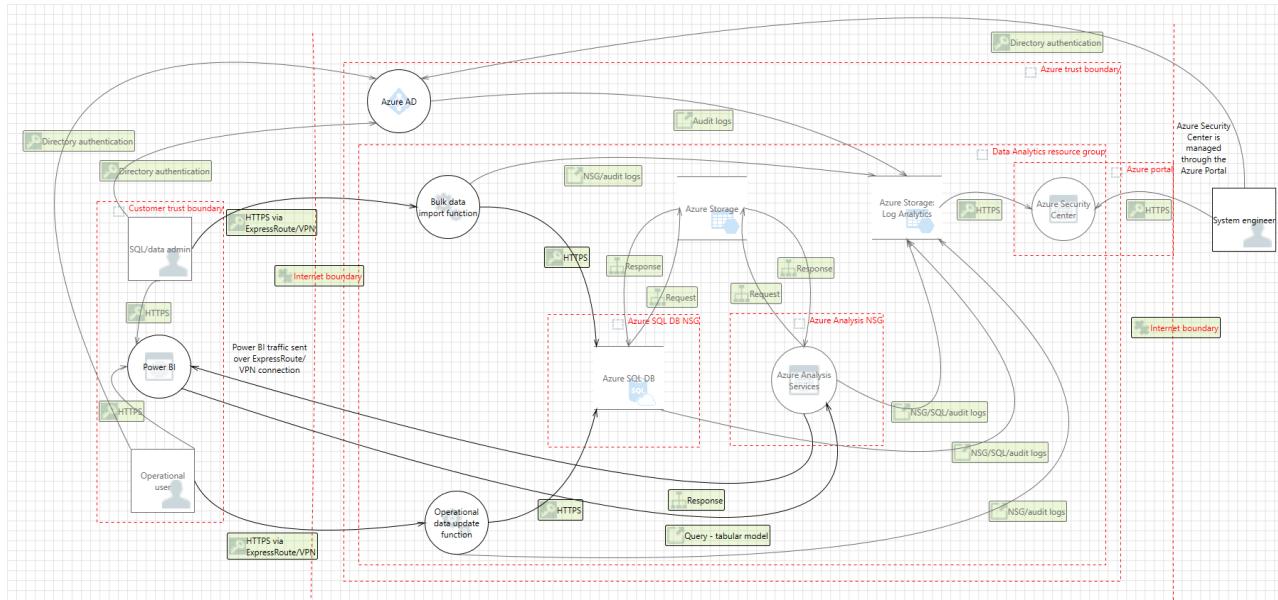
Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling

organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – UK NHS Customer Responsibility Matrix](#) lists all security principles required by UK NHS. This matrix details whether the implementation of each principle is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – UK NHS Data Analytics Implementation Matrix](#) provides information on which UK NHS requirements are addressed by the data analytics architecture, including detailed descriptions of how the implementation meets the requirements of each covered principle.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this data analytics reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are

available.

Extract-Transform-Load process

[PolyBase](#) can load data into Azure SQL Database without the need for a separate extract, transform, load or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing Azure Active Directory by making the deployed Active Directory infrastructure a subdomain of an Azure Active Directory forest.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: Data Warehouse for UK NHS

7/2/2018 • 17 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides a reference architecture and guidance for a data warehouse solution suitable for securely ingesting, staging, storing, and interacting with sensitive healthcare data. This solution demonstrates ways in which customers can comply with guidance provided in the [Cloud Security Good Practice Guide](#) published by [NHS Digital](#), a partner of the United Kingdom's (UK) Department of Health and Social Care (DHSC). The Cloud Security Good Practice Guide is based on the 14 [Cloud Security Principles](#) published by the UK National Cyber Security Centre (NCSC).

This reference architecture, implementation guide, and threat model are intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment without additional configuration. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

Architecture diagram and components

This solution provides a reference architecture which implements a high-performance and secure cloud data warehouse. There are two separate data tiers in this architecture: one where data is imported, stored, and staged within a clustered SQL environment, and another for the Azure SQL Data Warehouse where the data is loaded using an extract, transform, load tool (e.g. [PolyBase T-SQL queries](#)) for processing. Once data is stored in Azure SQL Data Warehouse, analytics can run at a massive scale.

Azure offers a variety of reporting and analytics services for the customer. This solution includes SQL Server Reporting Services for quick creation of reports from the Azure SQL Data Warehouse. All SQL traffic is encrypted with SSL through the inclusion of self-signed certificates. As a best practice, Azure recommends the use of a trusted certificate authority for enhanced security.

Data in the Azure SQL Data Warehouse is stored in relational tables with columnar storage, a format that significantly reduces the data storage costs while improving query performance. Depending on usage requirements, Azure SQL Data Warehouse compute resources can be scaled up or down or shut off completely if there are no active processes requiring compute resources.

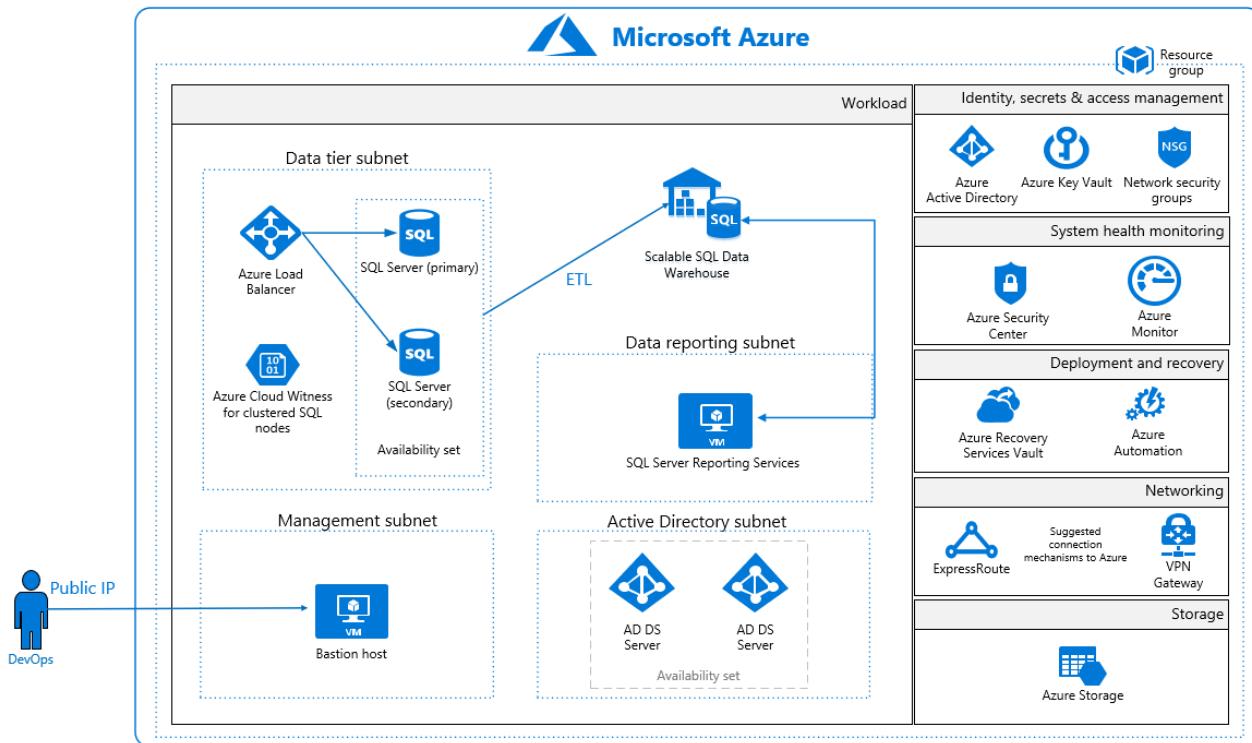
A SQL load balancer manages SQL traffic, ensuring high performance. All virtual machines in this reference architecture deploy in an availability set with SQL Server instances configured in an Always On availability group for high-availability and disaster-recovery capabilities.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources and keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard.

This data warehouse reference architecture also includes an Active Directory tier for management of resources within the architecture. The Active Directory subnet enables easy adoption under a larger Active Directory forest structure, allowing for continuous operation of the environment even when access to the larger forest is unavailable. All virtual machines are domain-joined to the Active Directory tier and use Active Directory group

policies to enforce security and compliance configurations at the operating system level.

A virtual machine serves as a management bastion host, providing a secure connection for administrators to access deployed resources. The data loads into the staging area through this management bastion host. **Microsoft recommends configuring a VPN or Azure ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are in the [Deployment architecture](#) section.

- Availability Sets
 - Active Directory domain controllers
 - SQL Cluster Nodes and Witness
- Azure Active Directory
- Azure Data Catalog
- Azure Key Vault
- Azure Monitor
- Azure Security Center
- Azure Load Balancer
- Azure Storage
- Azure Automation
- Azure Virtual Machines
 - (1) Bastion host
 - (2) Active Directory domain controller
 - (2) SQL Server Cluster Node
 - (1) SQL Server Witness
- Azure Virtual Network
 - (1) /16 Network
 - (4) /24 Networks
 - (4) Network security groups
- Recovery Services Vault

- SQL Data Warehouse
- SQL Server Reporting Services

Deployment architecture

The following section details the deployment and implementation elements.

SQL Data Warehouse: [SQL Data Warehouse](#) is an Enterprise Data Warehouse that leverages massively parallel processing to quickly run complex queries across petabytes of data, allowing users to efficiently identify healthcare data. Users can use simple PolyBase T-SQL queries to import big data into the SQL Data Warehouse and utilize the power of massively parallel processing to run high-performance analytics.

SQL Server Reporting Services: [SQL Server Reporting Services](#) provides quick creation of reports with tables, charts, maps, gauges, matrixes, and more for Azure SQL Data Warehouse.

Data Catalog: [Data Catalog](#) makes data sources easily discoverable and understandable by the users who manage the data. Common data sources can be registered, tagged, and searched for health-related data. The data remains in its existing location, but a copy of its metadata is added to Data Catalog, along with a reference to the data source location. The metadata is also indexed to make each data source easily discoverable via search and understandable to the users who discover it.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

Virtual network

This reference architecture defines a private virtual network with an address space of 10.0.0.0/16.

Network security groups: [Network security groups](#) contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual virtual machine level. The following Network security groups exist:

- A network security group for the Data Tier (SQL Server Clusters, SQL Server Witness, and SQL load balancer)
- A network security group for the management bastion host
- A network security group for Active Directory
- A network security group for SQL Server Reporting Services

Each of the Network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- [Diagnostic logs and events](#) are enabled and stored in a storage account
- Log Analytics is connected to the [network security group's diagnostic logs](#)

Subnets: Each subnet is associated with its corresponding network security group.

Data at rest

The architecture protects data at rest through multiple measures, including encryption and database auditing.

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by NHS Digital.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security module protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for virtual machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Furthermore, this reference architecture utilizes the [vulnerability assessment](#) in Azure Security Center. Once configured, a partner agent (e.g. Qualys) reports vulnerability data to the partner's management platform. In turn, the partner's management platform provides vulnerability and health monitoring data back to Azure Security Center, allowing customers to quickly identify vulnerable virtual machines.

Business continuity

High availability: Server workloads are grouped in an [Availability Set](#) to help ensure high availability of virtual machines in Azure. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure virtual machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS virtual machine without restoring the entire virtual machine, enabling faster restore times.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity

logs can help determine an operation's initiator, time of occurrence, and status.

- **Diagnostic logs:** Diagnostic logs include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

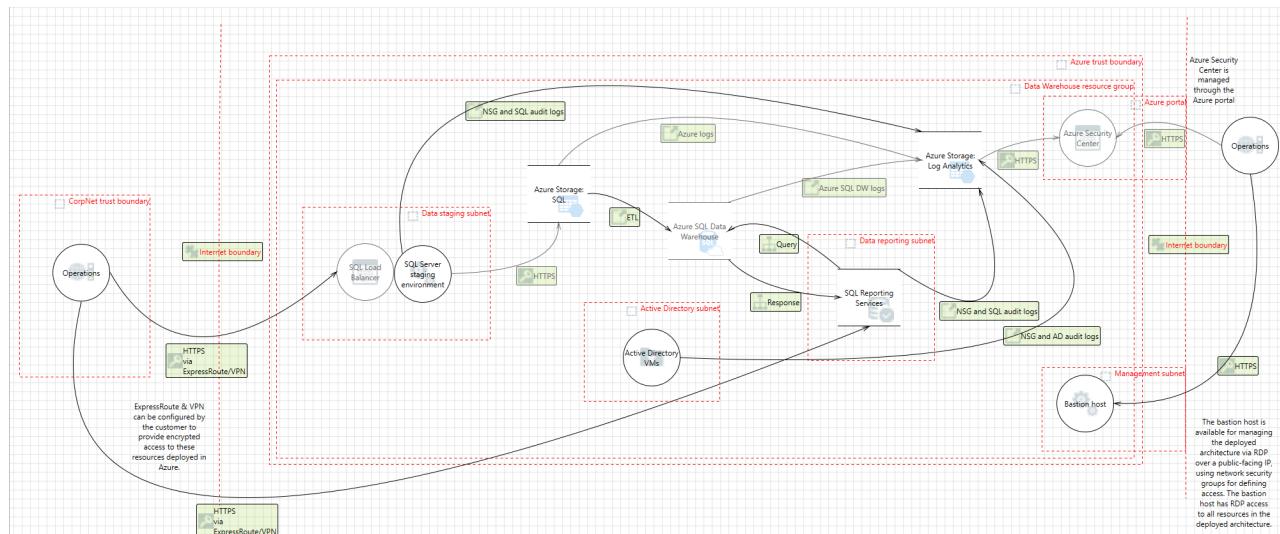
- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – UK NHS Customer Responsibility Matrix](#) lists all UK NHS requirements. This matrix details whether the implementation of each principle is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – UK NHS Data Warehouse Implementation Matrix](#) provides information on which UK NHS requirements are addressed by the data warehouse architecture, including detailed descriptions of how the implementation meets the requirements of each covered principle.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this PaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Extract-Transform-Load process

[PolyBase](#) can load data into Azure SQL Data Warehouse without the need for a separate extract, transform, load or import tool. PolyBase allows access to data through T-SQL queries. Microsoft's business intelligence and analysis stack, as well as third-party tools compatible with SQL Server, can be used with PolyBase.

Azure Active Directory setup

[Azure Active Directory](#) is essential to managing the deployment and provisioning access to personnel interacting with the environment. An existing Windows Server Active Directory can be integrated with Azure Active Directory in [four clicks](#). Customers can also tie the deployed Active Directory infrastructure (domain controllers) to an existing Azure Active Directory by making the deployed Active Directory infrastructure a subdomain of an Azure Active Directory forest.

Optional services

Azure offers a variety of services to assist with the storage and staging of formatted and unformatted data. The following services can be added to this reference architecture depending on customer requirements:

- [Azure Data Factory](#) is a managed cloud service that is built for complex hybrid extract-transform-load, and data integration projects. Azure Data Factory has capabilities to help trace and locate health-related data, including visualization and monitoring tools to identify when data arrived and where it came from. Using Azure Data Factory, customers can create and schedule data-driven workflows called pipelines that ingest data from disparate data stores. These pipelines allow customers to ingest data from both internal and external sources. Customers can then process and transform the data for output into data stores such as Azure SQL Data Warehouse.
- Customers can stage unstructured data in [Azure Data Lake Store](#), which enables the capture of data of any size,

type, and ingestion speed in a single place for operational and exploratory analytics. Azure Data Lake has capabilities that enable the extraction and conversion of data. Azure Data Lake Store is compatible with most open source components in the Hadoop ecosystem and integrates nicely with other Azure services such as Azure SQL Data Warehouse.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: IaaS Web Application for UK NHS

7/2/2018 • 15 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides a reference architecture and guidance for a infrastructure as a service (IaaS) web application suitable for the collection, storage, and retrieval of healthcare data. This solution demonstrates ways in which customers can comply with guidance provided in the [Cloud Security Good Practice Guide](#) published by [NHS Digital](#), a partner of the United Kingdom's (UK) Department of Health and Social Care (DHSC). The Cloud Security Good Practice Guide is based on the 14 [Cloud Security Principles](#) published by the UK National Cyber Security Centre (NCSC).

This reference architecture, implementation guide, and threat model are intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment without additional configuration. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

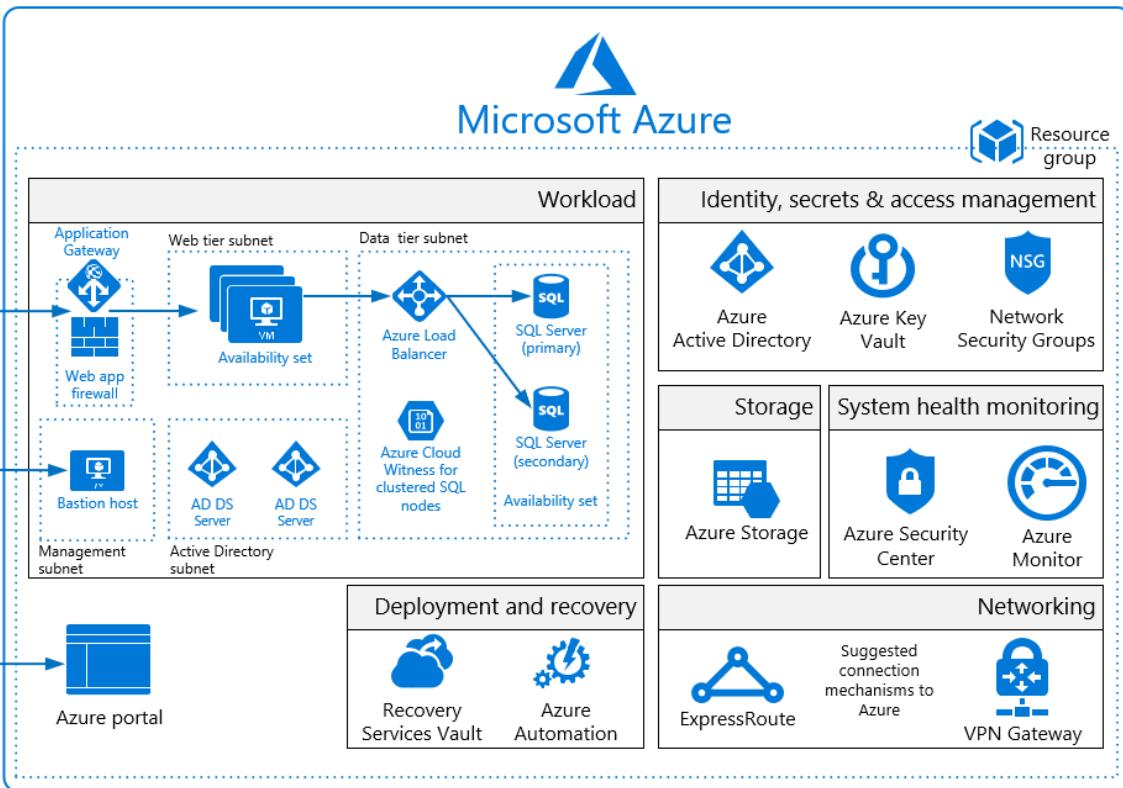
Architecture diagram and components

This solution deploys a reference architecture for an IaaS web application with a SQL Server backend. The architecture includes a web tier, data tier, Active Directory infrastructure, Application Gateway, and Load Balancer. Virtual machines deployed to the web and data tiers are configured in an availability set, and SQL Server instances are configured in an Always On availability group for high availability. Virtual machines are domain-joined, and Active Directory group policies are used to enforce security and compliance configurations at the operating system level.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's chosen datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and again stored as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources, including their keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard. Azure Application Gateway is configured as a firewall in prevention mode and disallows traffic that is not TLSv1.2.

A management bastion host provides a secure connection for administrators to access deployed resources. **Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.**



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

- Azure Virtual Machines
 - (1) management/bastion (Windows Server 2016 Datacenter)
 - (2) Active Directory domain controller (Windows Server 2016 Datacenter)
 - (2) SQL Server cluster node (SQL Server 2017 on Windows Server 2016)
 - (2) Web/IIS (Windows Server 2016 Datacenter)
- Availability Sets
 - (1) Active Directory domain controllers
 - (1) SQL cluster nodes
 - (1) Web/IIS
- Azure Virtual Network
 - (1) /16 Network
 - (5) /24 Networks
 - (5) Network security group
 - Recovery Services Vault
- Azure Application Gateway
 - (1) Web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP 3.0
 - Listener port: 443
- Azure Active Directory
- Azure Cloud Witness
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager

- Azure Security Center
- Azure Automation
- Azure Storage
 - (7) Geo-redundant storage accounts

Deployment architecture

The following section details the deployment and implementation elements.

Bastion host: The bastion host is the single point of entry that allows users to access the deployed resources in this environment. The bastion host provides a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the network security group.

This solution creates a virtual machine as a domain-joined bastion host with the following configurations:

- [Antimalware extension](#)
- [Azure Diagnostics extension](#)
- [Azure Disk Encryption](#) using Azure Key Vault
- An [auto-shutdown policy](#) to reduce consumption of virtual machine resources when not in use
- [Windows Defender Credential Guard](#) enabled so that credentials and other secrets run in a protected environment that is isolated from the running operating system

Virtual network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: This solution deploys resources in an architecture with a separate web subnet, database subnet, Active Directory subnet, and management subnet inside of a virtual network. Subnets are logically separated by network security group rules applied to the individual subnets to restrict traffic between subnets to only that necessary for system and management functionality.

See the configuration for [network security groups](#) deployed with this solution. Organizations can configure network security groups by editing the file above using [this documentation](#) as a guide.

Each of the subnets has a dedicated Network Security Group:

- 1 network security group for Application Gateway (LBNSG)
- 1 network security group for bastion host (MGTNSG)
- 1 network security group for primary and backup domain controllers (ADNSG)
- 1 network security group for SQL Servers and Cloud Witness (SQLNSG)
- 1 network security group for web tier (WEBNSG)

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. Additionally, all transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by NHS Digital.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-

encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.

- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security module protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.
- The solution is integrated with Azure Key Vault to manage IaaS virtual machine disk-encryption keys and secrets.

Patch management: Windows virtual machines deployed as part of this reference architecture are configured by default to receive automatic updates from Windows Update Service. This solution also includes the [Azure Automation](#) service through which updated deployments can be created to patch virtual machines when needed.

Malware protection: [Microsoft Antimalware](#) for Virtual Machines provides real-time protection capability that helps identify and remove viruses, spyware, and other malicious software, with configurable alerts when known malicious or unwanted software attempts to install or run on protected virtual machines.

Azure Security Center: With [Azure Security Center](#), customers can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler for customers to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Furthermore, this reference architecture utilizes the [vulnerability assessment](#) in Azure Security Center. Once configured, a partner agent (e.g. Qualys) reports vulnerability data to the partner's management platform. In turn, the partner's management platform provides vulnerability and health monitoring data back to Azure Security Center, allowing customers to quickly identify vulnerable virtual machines.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities using an Azure Application Gateway with a web application firewall configured, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Business continuity

High availability: The solution deploys all virtual machines in an [Availability Set](#). Availability sets ensure that the virtual machines are distributed across multiple isolated hardware clusters to improve availability. At least one virtual machine is available during a planned or unplanned maintenance event, meeting the 99.95% Azure SLA.

Recovery Services Vault: The [Recovery Services Vault](#) houses backup data and protects all configurations of Azure Virtual Machines in this architecture. With a Recovery Services Vault, customers can restore files and folders from an IaaS virtual machine without restoring the entire virtual machine, enabling faster restore times.

Cloud Witness: [Cloud Witness](#) is a type of Failover Cluster quorum witness in Windows Server 2016 that leverages Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations, but it uses the standard publicly available Azure Blob Storage. This eliminates the extra maintenance overhead of virtual machiness hosted in a public cloud.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting. Once collected, the data is organized into separate tables for each data type, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

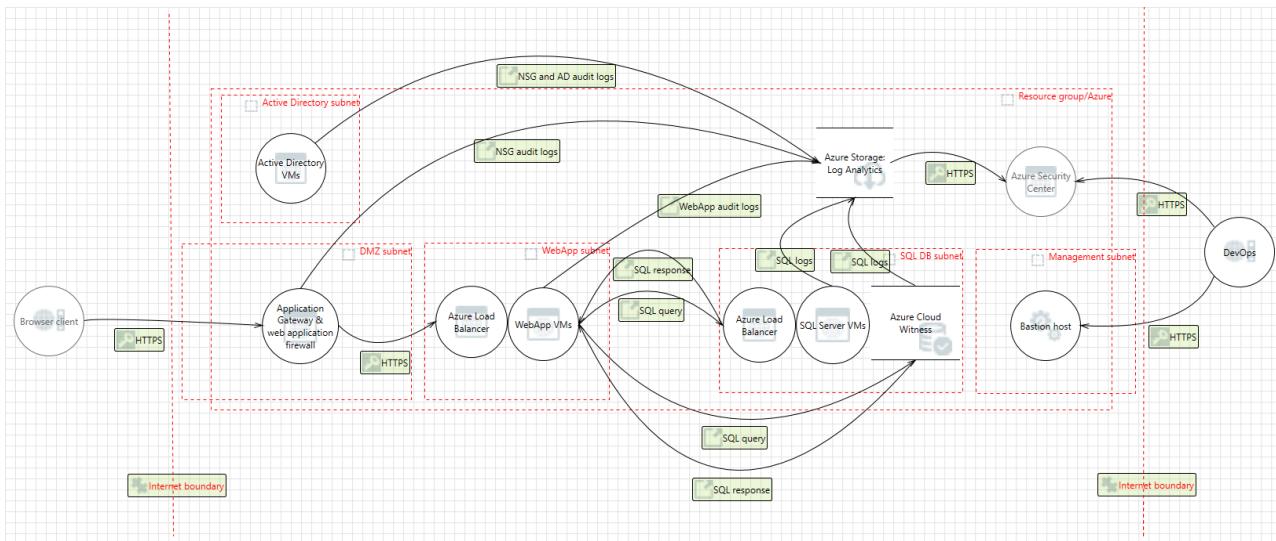
- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – UK NHS Customer Responsibility Matrix](#) lists all UK NHS requirements. This matrix details whether the implementation of each principle is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – UK NHS IaaS Web Application Implementation Matrix](#) provides information on which UK NHS requirements are addressed by the IaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered principle.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this IaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.

- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: PaaS Web Application for UK NHS

9/10/2018 • 15 minutes to read • [Edit Online](#)

Overview

This Azure Security and Compliance Blueprint provides a reference architecture and guidance for a platform as a service (PaaS) solution suitable for the collection, storage, and retrieval of healthcare data. This solution demonstrates ways in which customers can comply with guidance provided in the [Cloud Security Good Practice Guide](#) published by [NHS Digital](#), a partner of the United Kingdom's (UK) Department of Health and Social Care (DHSC). The Cloud Security Good Practice Guide is based on the 14 [Cloud Security Principles](#) published by the UK National Cyber Security Centre (NCSC).

This reference architecture, implementation guide, and threat model are intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment without additional configuration. Customers are responsible for conducting appropriate security and compliance assessments of any solution built using this architecture, as requirements may vary based on the specifics of each customer's implementation.

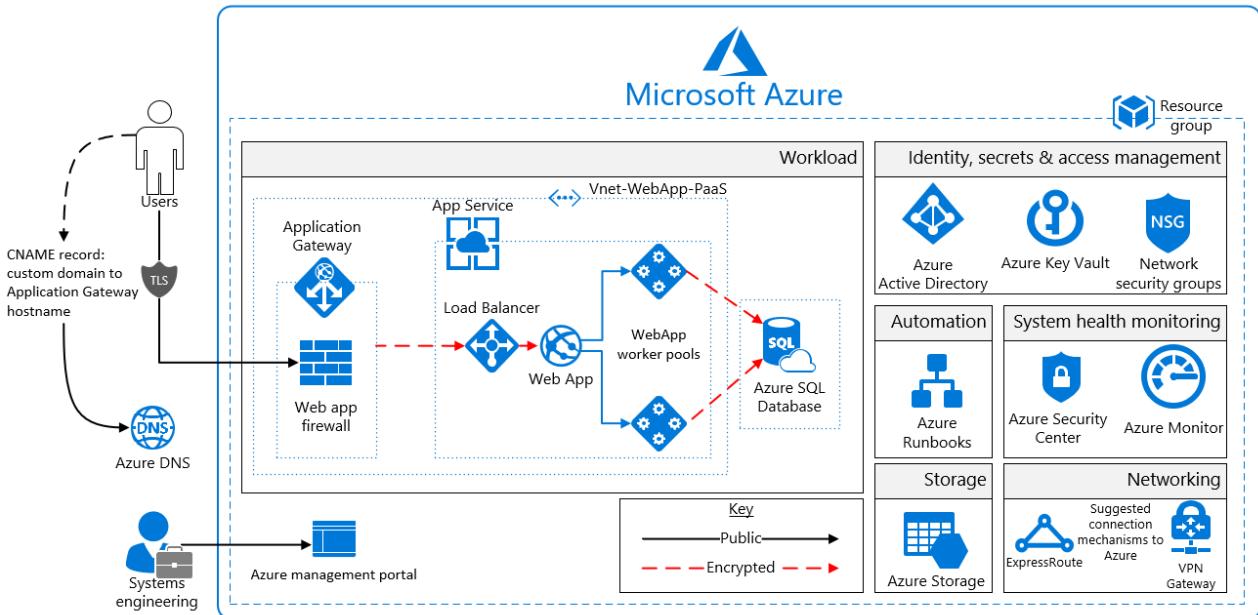
Architecture diagram and components

This solution provides a reference architecture for a PaaS web application with an Azure SQL Database backend. The web application is hosted in an isolated Azure App Service Environment, which is a private, dedicated environment in an Azure datacenter. The environment load balances traffic for the web application across virtual machines managed by Azure. All external connections require TLSv1.2. This architecture also includes network security groups, an Application Gateway, Azure DNS, and Load Balancer.

The solution uses Azure Storage accounts, which customers can configure to use Storage Service Encryption to maintain confidentiality of data at rest. Azure stores three copies of data within a customer's selected datacenter for resiliency. Geographic redundant storage ensures that data will be replicated to a secondary datacenter hundreds of miles away and stored again as three copies within that datacenter, preventing an adverse event at the customer's primary data center from resulting in a loss of data.

For enhanced security, all resources in this solution are managed as a resource group through Azure Resource Manager. Azure Active Directory role-based access control is used for controlling access to deployed resources and keys in Azure Key Vault. System health is monitored through Azure Security Center and Azure Monitor. Customers configure both monitoring services to capture logs and display system health in a single, easily navigable dashboard. Azure Application Gateway is configured as a firewall in prevention mode and disallows traffic that is not TLSv1.2. The solution utilizes Azure Application Service Environment v2 to isolate the web tier in a non-multi-tenant environment.

Microsoft recommends configuring a VPN or ExpressRoute connection for management and data import into the reference architecture subnet.



This solution uses the following Azure services. Details of the deployment architecture are in the [deployment architecture](#) section.

- Application Gateway
 - Web application firewall
 - Firewall mode: prevention
 - Rule set: OWASP
 - Listener port: 443
- Azure Active Directory
- Azure Application Service Environment v2
- Azure Automation
- Azure DNS
- Azure Key Vault
- Azure Load Balancer
- Azure Monitor
- Azure Resource Manager
- Azure Security Center
- Azure SQL Database
- Azure Storage
- Azure Virtual Network
 - Network security groups
- Azure Web App

Deployment architecture

The following section details the deployment and implementation elements.

Azure Resource Manager: [Azure Resource Manager](#) enables customers to work with the resources in the solution as a group. Customers can deploy, update, or delete all the resources for the solution in a single, coordinated operation. Customers use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help customers manage their resources after deployment.

App Service Environment v2: The Azure App Service Environment is an App Service feature that provides a fully isolated and dedicated environment for securely running App Service applications at a high scale.

The App Service Environment is isolated to only run a single application and is always deployed into a virtual network. This isolation feature enables the reference architecture to have complete tenant isolation, removing it from Azure's multi-tenant environment. This isolation feature is required to meet the requirements of UK NHS principle 3. Customers have fine-grained control over both inbound and outbound application network traffic, and applications can establish high-speed secure connections over virtual networks to on-premises corporate resources. Customers can "auto-scale" with App Service Environment based on load metrics, available budget, or a defined schedule.

Use of App Service Environment for this architecture allows for the following controls/configurations:

- Host inside a secured Azure virtual network and network security rules
- Self-signed internal load balancer certificate for HTTPS communication. As a best practice, Microsoft recommends the use of a trusted certificate authority for enhanced security.
- Internal load balancing mode
- Disable TLS 1.0
- Change TLS cipher
- Control inbound traffic N/W ports
- Web application firewall – restrict data
- Allow Azure SQL Database traffic

Azure Web App: Azure Web Apps enables customers to build and host web applications in the programming language of their choice without managing infrastructure. It offers auto-scaling and high availability, supports both Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repo.

Virtual Network

The architecture defines a private virtual network with an address space of 10.200.0.0/16.

Network security groups: Network security groups contain access control lists that allow or deny traffic within a virtual network. Network security groups can be used to secure traffic at a subnet or individual virtual machine level. The following Network security groups exist:

- 1 network security group for Application Gateway
- 1 network security group for App Service Environment
- 1 network security group for Azure SQL Database

Each of the Network security groups have specific ports and protocols open so that the solution can work securely and correctly. In addition, the following configurations are enabled for each network security group:

- Diagnostic logs and events are enabled and stored in a storage account
- Log Analytics is connected to the network security group's diagnostics logs

Subnets: Each subnet is associated with its corresponding network security group.

Azure DNS: The Domain Name System, or DNS, is responsible for translating (or resolving) a website or service name to its IP address. Azure DNS is a hosting service for DNS domains that provides name resolution using Azure infrastructure. By hosting domains in Azure, users can manage DNS records using the same credentials, APIs, tools, and billing as other Azure services. Azure DNS also supports private DNS domains.

Azure Load Balancer: Azure Load Balancer allows customers to scale their applications and create high availability for services. Load Balancer supports inbound as well as outbound scenarios, and provides low latency, high throughput, and scales up to millions of flows for all TCP and UDP applications.

Data in transit

Azure encrypts all communications to and from Azure datacenters by default. All transactions to Azure Storage through the Azure portal occur via HTTPS.

Data at rest

The architecture protects data at rest through encryption, database auditing, and other measures.

Azure Storage: To meet encrypted data at rest requirements, all [Azure Storage](#) uses [Storage Service Encryption](#). This helps protect and safeguard data in support of organizational security commitments and compliance requirements defined by NHS Digital.

Azure Disk Encryption: [Azure Disk Encryption](#) leverages the BitLocker feature of Windows to provide volume encryption for data disks. The solution integrates with Azure Key Vault to help control and manage the disk-encryption keys.

Azure SQL Database: The Azure SQL Database instance uses the following database security measures:

- [Active Directory authentication and authorization](#) enables identity management of database users and other Microsoft services in one central location.
- [SQL database auditing](#) tracks database events and writes them to an audit log in an Azure storage account.
- Azure SQL Database is configured to use [transparent data encryption](#), which performs real-time encryption and decryption of the database, associated backups, and transaction log files to protect information at rest. Transparent data encryption provides assurance that stored data has not been subject to unauthorized access.
- [Firewall rules](#) prevent all access to database servers until proper permissions are granted. The firewall grants access to databases based on the originating IP address of each request.
- [SQL Threat Detection](#) enables the detection and response to potential threats as they occur by providing security alerts for suspicious database activities, potential vulnerabilities, SQL injection attacks, and anomalous database access patterns.
- [Encrypted Columns](#) ensure that sensitive data never appears as plaintext inside the database system. After enabling data encryption, only client applications or application servers with access to the keys can access plaintext data.
- [SQL Database dynamic data masking](#) limits sensitive data exposure by masking the data to non-privileged users or applications. Dynamic data masking can automatically discover potentially sensitive data and suggest the appropriate masks to be applied. This helps to identify and reduce access to data such that it does not exit the database via unauthorized access. Customers are responsible for adjusting dynamic data masking settings to adhere to their database schema.

Identity management

The following technologies provide capabilities to manage access to data in the Azure environment:

- [Azure Active Directory](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for this solution are created in Azure Active Directory, including users accessing the Azure SQL Database.
- Authentication to the application is performed using Azure Active Directory. For more information, see [integrating applications with Azure Active Directory](#). Additionally, the database column encryption uses Azure Active Directory to authenticate the application to Azure SQL Database. For more information, see how to [protect sensitive data in Azure SQL Database](#).
- [Azure role-based access control](#) enables administrators to define fine-grained access permissions to grant only the amount of access that users need to perform their jobs. Instead of giving every user unrestricted permission for Azure resources, administrators can allow only certain actions for accessing data. Subscription access is limited to the subscription administrator.
- [Azure Active Directory Privileged Identity Management](#) enables customers to minimize the number of users who have access to certain information. Administrators can use Azure Active Directory Privileged Identity Management to discover, restrict, and monitor privileged identities and their access to resources. This functionality can also be used to enforce on-demand, just-in-time administrative access when needed.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities affecting an organization's identities, configures automated responses to detected suspicious actions related to an organization's identities, and investigates suspicious incidents to take appropriate action to resolve them.

Security

Secrets management: The solution uses [Azure Key Vault](#) for the management of keys and secrets. Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. The following Azure Key Vault capabilities help customers protect and access such data:

- Advanced access policies are configured on a need basis.
- Key Vault access policies are defined with minimum required permissions to keys and secrets.
- All keys and secrets in Key Vault have expiration dates.
- All keys in Key Vault are protected by specialized hardware security modules. The key type is a hardware security module protected 2048-bit RSA Key.
- All users and identities are granted minimum required permissions using role-based access control.
- Diagnostics logs for Key Vault are enabled with a retention period of at least 365 days.
- Permitted cryptographic operations for keys are restricted to the ones required.

Azure Security Center: With [Azure Security Center](#), this solution can centrally apply and manage security policies across workloads, limit exposure to threats, and detect and respond to attacks. Additionally, Azure Security Center accesses existing configurations of Azure services to provide configuration and service recommendations to help improve security posture and protect data.

Azure Security Center uses a variety of detection capabilities to alert customers of potential attacks targeting their environments. These alerts contain valuable information about what triggered the alert, the resources targeted, and the source of the attack. Azure Security Center has a set of [predefined security alerts](#), which are triggered when a threat, or suspicious activity takes place. [Custom alert rules](#) in Azure Security Center allow customers to define new security alerts based on data that is already collected from their environment.

Azure Security Center provides prioritized security alerts and incidents, making it simpler to discover and address potential security issues. A [threat intelligence report](#) is generated for each detected threat to assist incident response teams in investigating and remediating threats.

Azure Application Gateway: The architecture reduces the risk of security vulnerabilities using an Azure Application Gateway with a web application firewall configured, and the OWASP ruleset enabled. Additional capabilities include:

- [End-to-end-SSL](#)
- Enable [SSL Offload](#)
- Disable [TLS v1.0 and v1.1](#)
- [Web application firewall](#) (prevention mode)
- [Prevention mode](#) with OWASP 3.0 ruleset
- Enable [diagnostics logging](#)
- [Custom health probes](#)
- [Azure Security Center](#) and [Azure Advisor](#) provide additional protection and notifications. Azure Security Center also provides a reputation system.

Logging and auditing

Azure services extensively log system and user activity, as well as system health:

- **Activity logs:** [Activity logs](#) provide insight into operations performed on resources in a subscription. Activity logs can help determine an operation's initiator, time of occurrence, and status.
- **Diagnostic logs:** [Diagnostic logs](#) include all logs emitted by every resource. These logs include Windows event system logs, Azure Storage logs, Key Vault audit logs, and Application Gateway access and firewall logs. All diagnostic logs write to a centralized and encrypted Azure storage account for archival. The retention is user-configurable, up to 730 days, to meet organization-specific retention requirements.

Log Analytics: These logs are consolidated in [Log Analytics](#) for processing, storing, and dashboard reporting.

Once collected, the data is organized into separate tables for each data type, which allows all data to be analyzed together regardless of its original source. Furthermore, Azure Security Center integrates with Log Analytics allowing customers to use Log Analytics queries to access their security event data and combine it with data from other services.

The following Log Analytics [management solutions](#) are included as a part of this architecture:

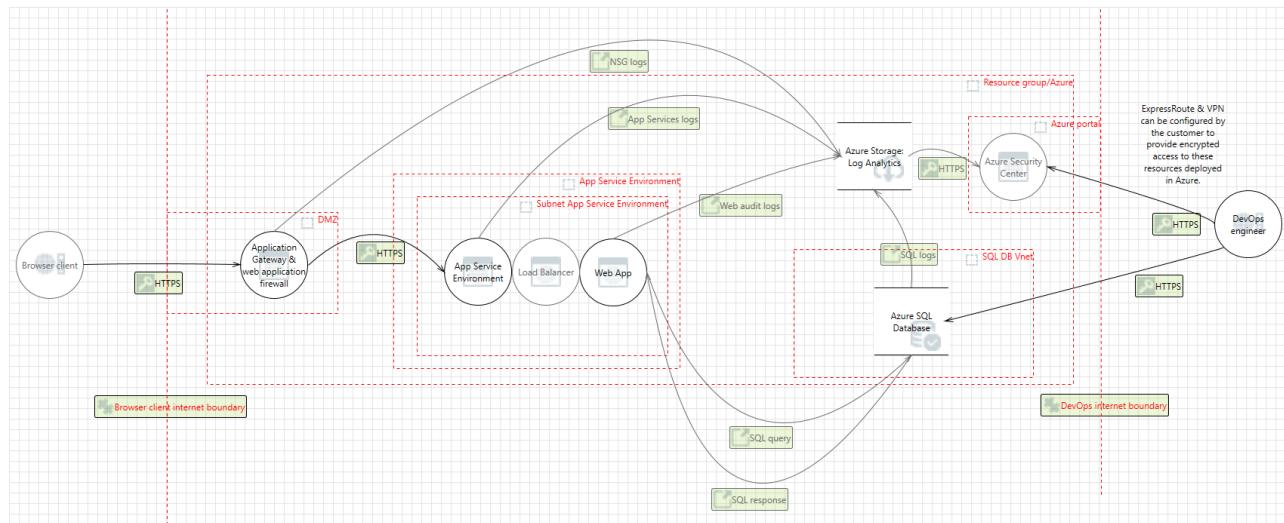
- [Active Directory Assessment](#): The Active Directory Health Check solution assesses the risk and health of server environments on a regular interval and provides a prioritized list of recommendations specific to the deployed server infrastructure.
- [SQL Assessment](#): The SQL Health Check solution assesses the risk and health of server environments on a regular interval and provides customers with a prioritized list of recommendations specific to the deployed server infrastructure.
- [Agent Health](#): The Agent Health solution reports how many agents are deployed and their geographic distribution, as well as how many agents which are unresponsive and the number of agents which are submitting operational data.
- [Activity Log Analytics](#): The Activity Log Analytics solution assists with analysis of the Azure activity logs across all Azure subscriptions for a customer.

Azure Automation: [Azure Automation](#) stores, runs, and manages runbooks. In this solution, runbooks help collect logs from Azure SQL Database. The Automation [Change Tracking](#) solution enables customers to easily identify changes in the environment.

Azure Monitor: [Azure Monitor](#) helps users track performance, maintain security, and identify trends by enabling organizations to audit, create alerts, and archive data, including tracking API calls in their Azure resources.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



Compliance documentation

The [Azure Security and Compliance Blueprint – UK NHS Customer Responsibility Matrix](#) lists all UK NHS requirements. This matrix details whether the implementation of each principle is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint – UK NHS PaaS Web Application Implementation Matrix](#) provides information on which UK NHS requirements are addressed by the PaaS web application architecture, including detailed descriptions of how the implementation meets the requirements of each covered principle.

Guidance and recommendations

VPN and ExpressRoute

A secure VPN tunnel or [ExpressRoute](#) needs to be configured to securely establish a connection to the resources deployed as a part of this PaaS web application reference architecture. By appropriately setting up a VPN or ExpressRoute, customers can add a layer of protection for data in transit.

By implementing a secure VPN tunnel with Azure, a virtual private connection between an on-premises network and an Azure virtual network can be created. This connection takes place over the Internet and allows customers to securely "tunnel" information inside an encrypted link between the customer's network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. The [IPsec tunnel mode](#) is used in this option as an encryption mechanism.

Because traffic within the VPN tunnel does traverse the Internet with a site-to-site VPN, Microsoft offers another, even more secure connection option. Azure ExpressRoute is a dedicated WAN link between Azure and an on-premises location or an Exchange hosting provider. As ExpressRoute connections do not go over the Internet, these connections offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet. Furthermore, because this is a direct connection of customer's telecommunication provider, the data does not travel over the Internet and therefore is not exposed to it.

Best practices for implementing a secure hybrid network that extends an on-premises network to Azure are [available](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint - Three-Tier IaaS Web Application for UK OFFICIAL

7/13/2018 • 12 minutes to read • [Edit Online](#)

Overview

This article provides guidance and automation scripts to deliver a Microsoft Azure three-tier web based architecture appropriate for handling many workloads classified as OFFICIAL in the United Kingdom.

Using an Infrastructure as Code approach, the set of [Azure Resource Manager](#) templates deploy an environment that aligns to the UK National Cyber Security Centre (NCSC) 14 [Cloud Security Principles](#) and the Center for Internet Security (CIS) [Critical Security Controls](#).

The NCSC recommend their Cloud Security Principles be used by customers to evaluate the security properties of the service, and to help understand the division of responsibility between the customer and supplier. We've provided information against each of these principles to help you understand the split of responsibilities.

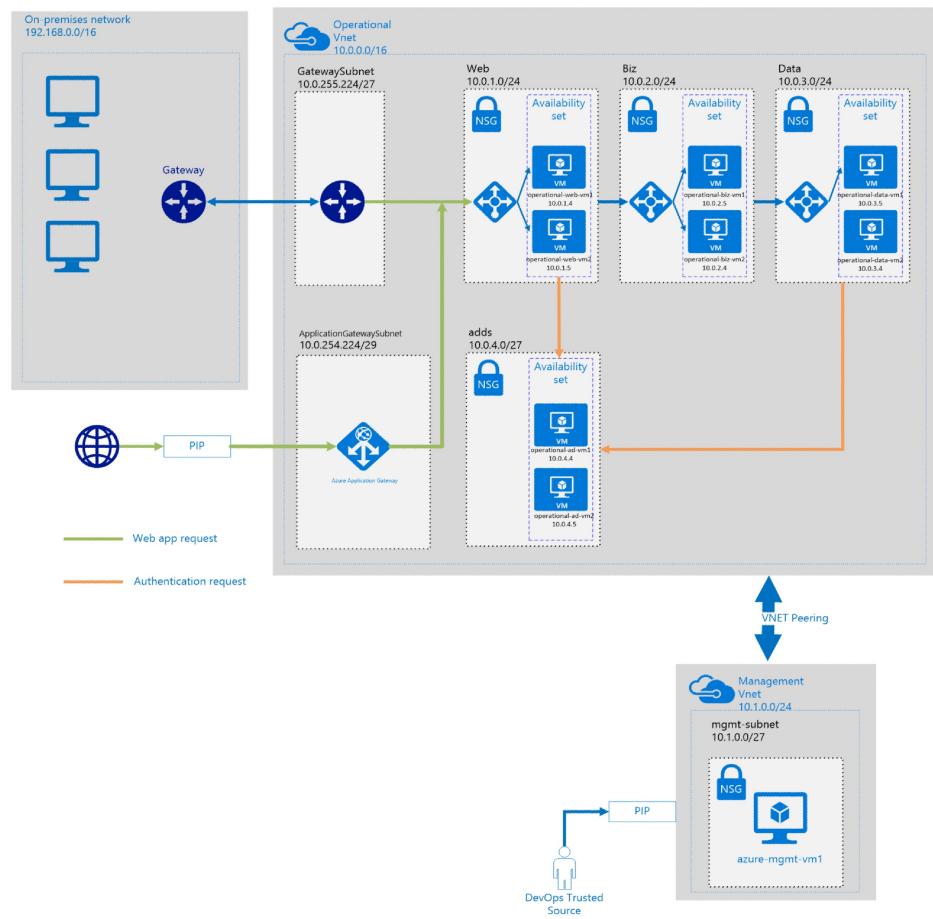
This architecture and corresponding Azure Resource Manager templates are supported by the Microsoft whitepaper, [14 Cloud Security Controls for UK cloud Using Microsoft Azure](#). This paper catalogues how Azure services align with the UK NCSC 14 Cloud Security Principles, thereby enabling organisations to fast-track their ability to meet their compliance obligations using cloud-based services globally and in the UK on the Microsoft Azure cloud.

This template deploys the infrastructure for the workload. Application code and supporting business tier and data tier software must be installed and configured. Detailed deployment instructions are available [here](#).

If you do not have an Azure subscription then you can sign up quickly and easily - [Get Started with Azure](#).

Architecture Diagram and Components

The Azure templates deliver a three-tier web application architecture in an Azure cloud environment that supports UK OFFICIAL workloads. The architecture delivers a secure hybrid environment that extends an on-premises network to Azure allowing web based workloads to be accessed securely by corporate users or from the internet.



This solution uses the following Azure services. Details of the deployment architecture are located in the [deployment architecture](#) section.

(1) /16 Virtual Network - Operational VNet

- (3) /24 subnets - 3-tier (Web, Biz, Data)
- (1) /27 subnet - ADDS
- (1) /27 subnet - Gateway Subnet
- (1) /29 subnet - Application Gateway Subnet
- Uses Default (Azure-Provided) DNS
- Peering enabled to Management VNet
- Network Security Group (NSG) for managing traffic flow

(1) /24 Virtual Network - Management VNet

- (1) /27 subnet
- Uses (2) ADDS DNS and (1) Azure DNS entries
- Peering enabled to Operational VNet
- Network Security Group (NSG) for managing traffic flow

(1) Application Gateway

- WAF - enabled
- WAF Mode - Prevention
- Rule set: OWASP 3.0
- HTTP Listener on Port 80
- Connectivity/Traffic regulated through NSG
- Public IP address endpoint defined (Azure)

(1) VPN - Route-based, Site-2-Site IPSec VPN tunnel

- Public IP address endpoint defined (Azure)
- Connectivity/Traffic regulated through NSG
- (1) local network gateway (on-premises endpoint)
- (1) Azure network gateway (Azure endpoint)

(9) Virtual Machines - All VMs are deployed with Azure IaaS Antimalware DSC settings

- (2) Active Directory Domain Services Domain Controllers (Windows Server 2012 R2)
 - (2) DNS Server Roles - 1 per VM
 - (2) NICs connected to Operational VNet - 1 per VM
 - Both are domain-joined to the domain defined in the template
 - Domain created as a part of the deployment
- (1) Jumpbox (Bastion Host) Management VM
 - 1 NIC on the Management VNet with Public IP address
 - NSG is used for limiting traffic (in/out) to specific sources
 - Not domain-joined
- (2) Web Tier VMs
 - (2) IIS Server Roles - 1 per VM
 - (2) NICs connected to Operational VNet - 1 per VM
 - Not domain-joined
- (2) Biz Tier VMs
 - (2) NICs connected to Operational VNet - 1 per VM
 - Not domain-joined
- (2) Data Tier VMs
 - (2) NICs connected to Operational VNet - 1 per VM
 - Not domain-joined

Availability Sets

- (1) Active Directory Domain Controller VM set - 2 VMs
- (1) Web Tier VM set - 2 VMs
- (1) Biz Tier VM set - 2 VMs
- (1) Data Tier VM set - 2 VMs

Load Balancer

- (1) Web Tier Load Balancer
- (1) Biz Tier Load Balancer
- (1) Data Tier Load Balancer

Storage

- (14) Total Storage Accounts
 - Active Directory Domain Controller Availability Set
 - (2) Primary Locally Redundant Storage (LRS) accounts - 1 for each VM
 - (1) Diagnostic Locally Redundant Storage (LRS) account for the ADDS Availability Set
 - Management Jumpbox VM

- (1) Primary Locally Redundant Storage (LRS) account for the Jumpbox VM
- (1) Diagnostic Locally Redundant Storage (LRS) account for the Jumpbox VM
- Web Tier VMs
 - (2) Primary Locally Redundant Storage (LRS) accounts - 1 for each VM
 - (1) Diagnostic Locally Redundant Storage (LRS) account for the Web Tier Availability Set
- Biz Tier VMs
 - (2) Primary Locally Redundant Storage (LRS) accounts - 1 for each VM
 - (1) Diagnostic Locally Redundant Storage (LRS) account for the Biz Tier Availability Set
- Data Tier VMs
 - (2) Primary Locally Redundant Storage (LRS) accounts - 1 for each VM
 - (1) Diagnostic Locally Redundant Storage (LRS) account for the Data Tier Availability Set

Deployment Architecture:

On-Premises Network: A private local-area network implemented in an organisation.

Production VNet: The Production [VNet](#) (Virtual Network) hosts the application and other operational resources running in Azure. Each VNet may contain several subnets which are used for isolating and managing network traffic.

Web Tier: Handles incoming HTTP requests. Responses are returned through this tier.

Business Tier: Implements business processes and other functional logic for the system.

Database Tier: Provides persistent data storage, using [SQL Server Always On Availability Groups](#) for high availability. Customers may use [Azure SQL Database](#) as a PaaS alternative.

Gateway: The [VPN Gateway](#) provides connectivity between the routers in the on-premises network and the production VNet.

Internet Gateway and Public IP Address: The internet gateway exposes application services to users through the internet. Traffic accessing these services is secured using an [Application Gateway](#) offering Layer 7 routing and load balancing capabilities with web application firewall (WAF) protection.

Management VNet: This [VNet](#) contains resources that implement management and monitoring capabilities for the workloads running in the production VNet.

Jumpbox: Also called a [bastion host](#), which is a secure VM on the network that administrators use to connect to VMs in the production VNet. The jumpbox has an NSG that allows remote traffic only from public IP addresses on a safe list. To permit remote desktop (RDP) traffic, the source of the traffic needs to be defined in the NSG. Management of production resources is via RDP using a secured Jumpbox VM.

User Defined Routes: [User defined routes](#) are used to define the flow of IP traffic within Azure VNets.

Network Peered VNets: The Production and Management VNets are connected using [VNet peering](#). These VNets are still managed as separate resources, but appear as one for all connectivity purposes for these virtual machines. These networks communicate with each other directly by using private IP addresses. VNet peering is subject to the VNets being in the same Azure Region.

Network Security Groups: [NSGs](#) contain Access Control Lists that allow or deny traffic within a VNet. NSGs can be used to secure traffic at a subnet or individual VM level.

Active Directory Domain Services (AD DS): This architecture provides a dedicated [Active Directory Domain Services](#) deployment.

Logging and Audit: [Azure Activity Log](#) captures operations taken on the resources in your subscription such as who initiated the operation, when the operation occurred, the status of the operation and the values of other properties that might help you research the operation. Azure Activity Log is an Azure platform service that

captures all actions on a subscription. Logs can be archived or exported if required.

Network Monitoring and Alerting: [Azure Network Watcher](#) is a platform service provides network packet capture, flow logging, topology tools and diagnostics for network traffics within your VNets.

Guidance and Recommendations

Business Continuity

High Availability: Server workloads are grouped in a [Availability Set](#) to help ensure high availability of virtual machines in Azure. This configuration helps ensure that during a planned or unplanned maintenance event at least one virtual machine will be available and meet the 99.95% Azure SLA.

Logging and Audit

Monitoring: [Azure Monitor](#) is the platform service that provides a single source for monitoring the activity log, metrics, and diagnostic logs of all your Azure resources. Azure Monitor can be configured to visualize, query, route, archive, and act on the metrics and logs coming from resources in Azure. It is recommended that Resource Based Access Control is used to secure the audit trail to help ensure that users don't have the ability to modify the logs.

Activity Logs: Configure [Azure Activity Logs](#) to provide insight into the operations that were performed on resources in your subscription.

Diagnostic Logs: [Diagnostic Logs](#) are all logs emitted by a resource. These logs could include Windows event system logs, blob, table, and queue logs.

Firewall Logs: Application Gateway provides full diagnostics and access logs. Firewall logs are available for application gateway resources that have WAF enabled.

Log Archiving: Log data storage can be configured to write to a centralised Azure storage account for archival and a defined retention period. Logs can be processed using Azure Log Analytics or by third party SIEM systems.

Identity

Active Directory Domain Services: This architecture delivers an Active Directory Domain Services deployment in Azure. For specific recommendations on implementing Active Directory in Azure, see the following articles:

[Extending Active Directory Domain Services \(AD DS\) to Azure.](#)

[Guidelines for Deploying Windows Server Active Directory on Azure Virtual Machines.](#)

Active Directory Integration: As an alternative to a dedicated AD DS architecture, customers may wish to use [Azure Active Directory](#) integration or [Active Directory in Azure joined to an on-premises forest](#).

Security

Management Security: This blueprint allows administrators to connect to the management VNet and Jumpbox using RDP from a trusted source. Network traffic for the management VNet is controlled using NSGs. Access to port 3389 is restricted to traffic from a trusted IP range that can access the subnet containing the Jumpbox.

Customers may also consider using an [enhanced security administrative model](#) to secure the environment when connecting to the management VNet and Jumpbox. It is suggested that for enhanced security customers use a [Privileged Access Workstation](#) and RDGateway configuration. The use of network virtual appliances and public/private DMZs will offer further security enhancements.

Securing the Network: [Network Security Groups](#) (NSGs) are recommended for each subnet to provide a second level of protection against inbound traffic bypassing an incorrectly configured or disabled gateway. Example - [Resource Manager template for deploying an NSG](#).

Securing Public Endpoints: The internet gateway exposes application services to users through the internet. Traffic accessing these services is secured using an [Application Gateway](#), which provides a Web Application

Firewall and HTTPS protocol management.

IP Ranges: The IP ranges in the architecture are suggested ranges. Customers are advised to consider their own environment and use appropriate ranges.

Hybrid Connectivity: The cloud based workloads are connected to the on-premises datacenter through IPSEC VPN using the Azure VPN Gateway. Customers should ensure that they are using an appropriate VPN Gateway to connect to Azure. Example - [VPN Gateway Resource Manager template](#). Customers running large-scale, mission critical workloads with big data requirements may wish to consider a hybrid network architecture using [ExpressRoute](#) for private network connectivity to Microsoft cloud services.

Separation of Concerns: This reference architecture separates the VNets for management operations and business operations. Separate VNets and subnets allow traffic management, including traffic ingress and egress restrictions, by using NSGs between network segments following [Microsoft cloud services and network security](#) best practices.

Resource Management: Azure resources such as VMs, VNets, and load balancers are managed by grouping them together into [Azure Resource Groups](#). Resource Based Access Control roles can then be assigned to each resource group to restrict access to only authorized users.

Access Control Restrictions: Use [Role-Based Access Control](#) (RBAC) to manage the resources in your application using [custom roles](#) RBAC can be used to restrict the operations that DevOps can perform on each tier. When granting permissions, use the [principle of least privilege](#). Log all administrative operations and perform regular audits to ensure any configuration changes were planned.

Internet Access: This reference architecture utilises [Azure Application Gateway](#) as the internet facing gateway and load balancer. Some customers may also consider using third party network virtual appliances for additional layers of networking security as an alternative to the [Azure Application Gateway](#).

Azure Security Center: The [Azure Security Center](#) provides a central view of the security status of resources in the subscription, and provides recommendations that help prevent compromised resources. It can also be used to enable more granular policies. For example, policies can be applied to specific resource groups, which allows the enterprise to tailor its posture to risk. It is recommended that customers enable Azure Security Center in their Azure Subscription.

NCSC Cloud Security Principles Compliance Documentation

The Crown Commercial Service (an agency that works to improve commercial and procurement activity by the government) renewed the classification of Microsoft in-scope enterprise cloud services to G-Cloud v6, covering all its offerings at the OFFICIAL level. Details of Azure and G-Cloud can be found in the [Azure UK G-Cloud security assessment summary](#).

This blueprint aligns to the 14 cloud security principles that are documented in the NCSC [Cloud Security Principles](#) to help ensure an environment that supports workloads classified as UK-OFFICIAL.

The [Customer Responsibility Matrix](#) (Excel Workbook) lists all 14 cloud security principles, and the matrix denotes, for each principle (or principle subpart), whether the principle implementation is the responsibility of Microsoft, the customer, or shared between the two.

The [Principle Implementation Matrix](#) (Excel Workbook) lists all 14 cloud security principles, and the matrix denotes, for each principle (or principle subpart) that is designated a customer responsibility in the Customer Responsibilities Matrix, 1) if the blueprint automation implements the principle, and 2) a description of how the implementation aligns with the principle requirement(s).

Furthermore, the Cloud Security Alliance (CSA) published the Cloud Control Matrix to support customers in the evaluation of cloud providers and to identify questions that should be answered before moving to cloud services. In response, Microsoft Azure answered the CSA Consensus Assessment Initiative Questionnaire ([CSA CAIQ](#)),

which describes how Microsoft addresses the suggested principles.

Deploy the Solution

There are two methods that deployment users may use to deploy this blueprint automation. The first method uses a PowerShell script, whereas the second method utilises Azure portal to deploy the reference architecture. Detailed deployment instructions are available [here](#).

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security and Compliance Blueprint: PaaS Web Application Hosting for UK OFFICIAL Workloads

10/15/2018 • 16 minutes to read • [Edit Online](#)

Azure Security and Compliance Blueprints

Azure Blueprints consist of guidance documents and automation templates that deploy cloud-based architectures to offer solutions to scenarios that have accreditation or compliance requirements. Azure Blueprints are guidance and automation template collections that allow Microsoft Azure customers to accelerate delivery of their business goals through provisioning a foundation architecture that can be extended to meet any further requirements.

Overview

This Azure Security and Compliance Blueprint provides guidance and automation scripts to deliver a Microsoft Azure [platform as a service \(PaaS\)](#) hosted web application architecture appropriate for handling workloads classified as [UK OFFICIAL](#). This security classification encompasses the majority of information created or processed by the public sector. This includes routine business operations and services, which if lost, stolen or published in the media, some of which could have damaging consequences. The typical threat profile for the OFFICIAL classification is much the same as a private business who provides valuable information and services. UK OFFICIAL anticipates the need to defend UK Government data or services against threat or compromise by attackers with bounded capabilities and resources such as (but is not limited to) hactivists, single-issue pressure groups, investigative journalists, competent individual hackers, and the majority of criminal individuals and groups.

This blueprint has been reviewed by the UK National Cyber Security Centre (NCSC) and aligns to the NCSC 14 Cloud Security Principles.

The architecture uses Azure [platform as a service](#) components to deliver an environment that allows customers to avoid the expense and complexity of buying software licenses, of managing the underlying application infrastructure and middleware or the development tools, and other resources. Customers manage the applications and services that they develop, focusing on delivering business value, whilst Microsoft Azure manages the other Azure resources such as virtual machines, storage and networking, putting more of the [division of responsibility](#) for infrastructure management on to the Azure platform. [Azure App Services](#) offers auto-scaling, high availability, supports Windows and Linux, and enables automated deployments from GitHub, Azure DevOps, or any Git repository as default services. Through using App Services, developers can concentrate on delivering business value without the overhead of managing infrastructure. It is possible to build greenfield new Java, PHP, Node.js, Python, HTML or C# web applications or also to migrate existing cloud or on premises web applications to Azure App Services (although thorough due diligence and testing to confirm performance is required).

This blueprint focuses on the provisioning of a secure foundation [platform as a service](#) web-based interface for public and also back office users. This blueprint design scenario considers the use of Azure hosted web-based services where a public user can securely submit, view, and manage sensitive data; also that a back office or government operator can securely process the sensitive data that the public user has submitted. Use cases for this scenario could include:

- A user submitting a tax return, with a government operator processing the submission;
- A user requesting a service through a web-based application, with a back office user validating and delivering the service; or
- A user seeking and viewing public domain help information concerning a government service.

Using [Azure Resource Manager](#) templates and Azure Command Line Interface scripts, the blueprint deploys an

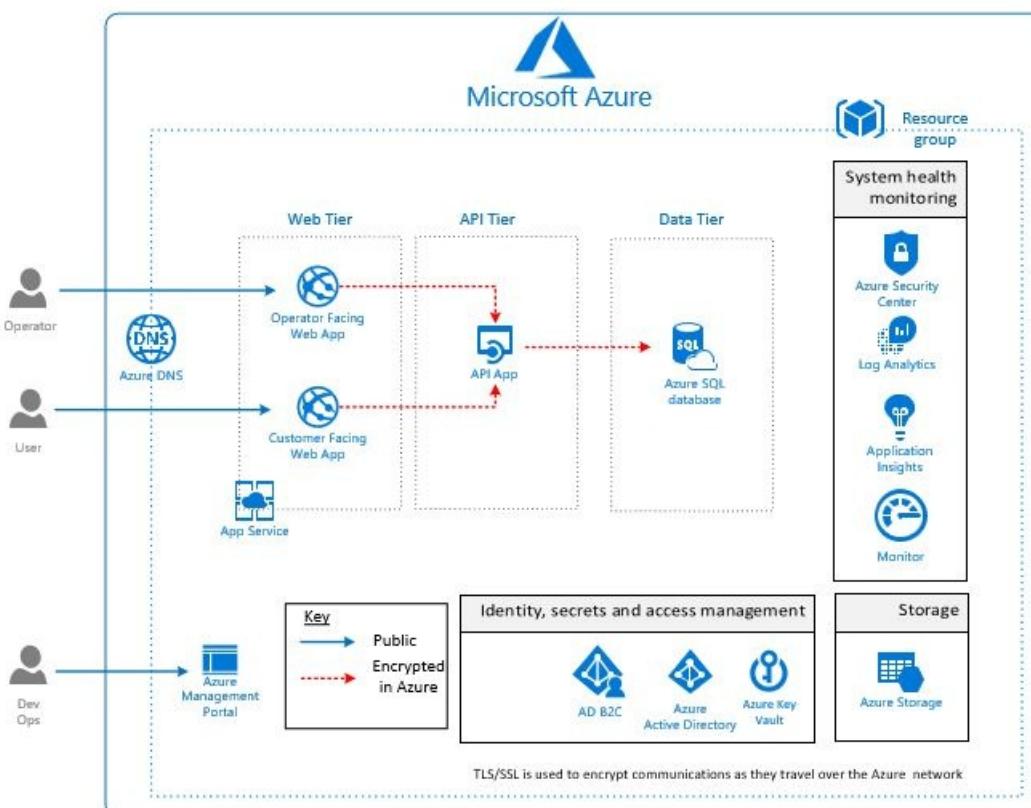
environment that aligns to the UK National Cyber Security Centre (NCSC) 14 [Cloud Security Principles](#) and the Center for Internet Security (CIS) [Critical Security Controls](#). The NCSC recommends their Cloud Security Principles be used by customers to evaluate the security properties of the service and to help understand the division of responsibility between the customer and supplier. Microsoft has provided information against each of these principles to help better understand the split of responsibilities. This architecture and corresponding Azure Resource Manager templates are supported by the Microsoft whitepaper, [14 Cloud Security Controls for UK cloud Using Microsoft Azure](#). This architecture has been reviewed by the NCSC, and aligns with the UK NCSC 14 Cloud Security Principles, thus enabling public sector organizations to fast-track their ability to meet compliance obligations using cloud-based services globally and in the UK on the Microsoft Azure cloud. This template deploys the infrastructure for the workload. Application code and supporting business tier and data tier software must be installed and configured by customers. Detailed deployment instructions are available [here](#).

This blueprint is a foundation architecture. Our customers can use this blueprint as a foundation for their OFFICIAL classification web-based workloads and expand on the templates and resources with their own requirements. This blueprint builds on the principles of the [UK-OFFICIAL Three-Tier IaaS Web Applications blueprint](#) to offer our customers both [infrastructure as a service \(IaaS\)](#) and PaaS implementation choices for hosting web-based workloads.

To deploy this blueprint, an Azure subscription is required. If you do not have an Azure subscription, you can sign up quickly and easily at no charge: Get Started with Azure. Click [here](#) for deployment instructions.

Architecture and components

This blueprint delivers a web application hosting solution in an Azure cloud environment that supports UK OFFICIAL workloads. The architecture delivers a secure environment that leverages Azure platform as a service capabilities. Within the environment, two App Service web apps are deployed (one for public users and one for back office users), with an API App tier to provide the business services for the web front end. An Azure SQL Database is deployed as a managed relational data store for the application. Connectivity to these components from outside the platform and between all these components is encrypted through TLS 1.2 to ensure data in transport privacy, with access authenticated by Azure Active Directory.



As part of the deployment architecture, secure storage provision, monitoring & logging, unified security

management & advanced threat protection, and management capabilities are also deployed to ensure that customers have all the tools required to secure and monitor their environment for this solution.

This solution uses the following Azure services. Details of the deployment architecture are in the [deployment architecture](#) section.

- Azure Active Directory
- App Service
- Web App
- API App
- Azure DNS
- Key Vault
- Azure Monitor
- Application Insights
- Log Analytics
- Azure Resource Manager
- Azure Security Center
- Azure SQL Database
- Azure Storage

Deployment architecture

The following section details the deployment and implementation elements.

Security

Identity and authentication

This blueprint ensures that access to resources is protected through directory and identity management services. This architecture makes full use of [identity as the security perimeter](#).

The following technologies provide identity management capabilities in the Azure environment:

- [Azure Active Directory \(Azure AD\)](#) is Microsoft's multi-tenant cloud-based directory and identity management service. All users for the solution were created in Azure Active Directory, including users accessing the SQL Database.
- Authentication to the operator facing web application and access for the administration of the Azure resources is performed using Azure AD. For more information, see [Integrating applications with Azure Active Directory](#).
- Database column encryption uses Azure AD to authenticate the application to Azure SQL Database. For more information, see [Always Encrypted: Protect sensitive data in SQL Database](#).
- The citizen facing web application is configured for public access. To allow for account creation and authentication through active directory or social network identity providers [Azure Active Directory B2C](#) can be integrated if required.
- [Azure Active Directory Identity Protection](#) detects potential vulnerabilities and risky accounts provides recommendations to enhance the security posture of your organization's identities, configures automated responses to detected suspicious actions related to your organization's identities, and investigates suspicious incidents and takes appropriate action to resolve them.
- [Azure Role-based Access Control \(RBAC\)](#) enables precisely focused access management for Azure. Subscription access is limited to the subscription administrator, and Azure Key Vault access is restricted only to users who require key management access.
- Through leveraging [Azure Active Directory Conditional Access](#) customers can enforce additional security controls on access to apps or users in their environment based on specific conditions such as location, device, state and sign in risk.
- [Azure DDoS Protection](#) combined with application design best practices, provides defense against DDoS

attacks, with always-on traffic monitoring, and real-time mitigation of common network-level attacks. With a PaaS architecture, platform level DDoS protection is transparent to the customer and incorporated into the platform but it is important to note that the application security design responsibility lies with the customer.

Data in transit

Data in transit from outside and between Azure components is protected using [Transport Layer Security/Secure Sockets Layer \(TLS/SSL\)](#), which uses symmetric cryptography based on a shared secret to encrypt communications as they travel over the network. By default, network traffic is secured using TLS 1.2.

Security and malware protection

[Azure Security Center](#) provides a centralized view of the security state of all your Azure resources. At a glance, you can verify that the appropriate security controls are in place and configured correctly, and you can quickly identify any resources that require attention.

[Azure Advisor](#) is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, high availability, and security of your Azure resources.

[Microsoft Antimalware](#) is a real-time protection capability that helps identify and remove viruses, spyware, and other malicious software. This by default is installed on the underlying PaaS virtual machine infrastructure and is managed by the Azure fabric transparently to the customer, ensuring

PaaS services in this blueprint

Azure App Service

Azure Web Apps provides a fully managed web hosting environment for web application developed in Java, PHP, Nodejs Python, HTML and C# without having to manage infrastructure. It offers auto-scaling and high availability, supports both Windows and Linux, and enables automated deployments from [Azure DevOps](#) or any Git-based repo.

App Service is [ISO, SOC, and PCI compliant](#) and can authenticate users with [Azure Active Directory](#) or with social login ([Google](#), [Facebook](#), [Twitter](#), and [Microsoft authentication](#)).

Basic, Standard, and Premium plans are for production workloads and run on dedicated Virtual Machine instances. Each instance can support multiple applications and domains. App services also support [IP address restrictions](#) to secure traffic to trusted IP addresses if required and also [managed identities for Azure resources](#) for secure connection to other PaaS services such as [Key Vault](#) and [Azure SQL Database](#). Where additional security is required our Isolated plan hosts your apps in a private, dedicated Azure environment and is ideal for apps that require secure connections with your on-premises network, or additional performance and scale.

This template deploys the following App Service features:

- [Standard App Service Plan Tier](#)
- Multiple Web App [deployment slots](#): Dev, Preview, QA, UAT and of course Production (default slot).
- [Managed identities for Azure resources](#) to connect to [Azure Key Vault](#) (this could also be used to provide access to [Azure SQL Database](#))
- Integration with [Azure Application Insights](#) to monitor performance
- [Diagnostic Logs](#)
- Metric [alerts](#)
- [Azure API Apps](#)

Azure SQL Database

SQL Database is a general-purpose relational database managed service in Microsoft Azure that supports structures such as relational data, JSON, spatial, and XML. SQL Database offers managed single SQL databases, managed SQL databases in an [elastic pool](#), and SQL [Managed Instances](#) (in public preview). It delivers [dynamically scalable performance]<https://docs.microsoft.com/azure/sql-database/sql-database-service-tiers>) and provides options such as [columnstore indexes](#) for extreme analytic analysis and reporting, and [in-memory OLTP](#)

for extreme transactional processing. Microsoft handles all patching and updating of the SQL code base seamlessly and abstracts away all management of the underlying infrastructure.

Azure SQL Database in this blueprint

The Azure SQL Database instance uses the following database security measures:

- [Server-level and database-level firewall rules](#), or through [Virtual Network Service Endpoints](#) using [virtual network rules](#).
- [Transparent data encryption](#) helps protect Azure SQL Database and Azure Data Warehouse against the threat of malicious activity. It performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application.
- [Azure AD authentication](#), you can centrally manage the identities of database users and other Microsoft services in one central location. Central ID management provides a single place to manage database users and simplifies permission management.
- Use of Azure Active Directory for database administration
- [Audit logs](#) to storage accounts
- Metric [alerts](#) for failed DB connections
- [SQL Threat Detection](#)
- [Always Encrypted columns](#)

Azure Storage

Microsoft [Azure Storage](#) is a Microsoft-managed cloud service that provides storage that is highly available, secure, durable, scalable, and redundant. Azure Storage consists of Blob storage, File Storage, and Queue storage.

Azure Storage in this blueprint

This template uses the following Azure Storage components:

- [Storage Service Encryption](#)
- Only allow HTTPS connections

Data at rest

Through [Storage Service Encryption](#) all data written to Azure Storage is encrypted through 256-bit AES encryption, one of the strongest block ciphers available. You can use Microsoft-managed encryption keys with SSE or you can use [your own encryption keys](#).

Storage accounts can be secured via [Virtual Network Service Endpoints](#) using [virtual network rules](#).

Detailed information about securing Azure Storage can be found in the [security guide](#).

Secrets management

Azure Key Vault

[Key Vault](#) is used to secure application keys and secrets to ensure that they are not accessible by third parties. Key Vault is not intended to be used as a store for user passwords. It allows you to create multiple secure containers, called vaults. These vaults are backed by hardware security modules (HSMs). Vaults help reduce the chances of accidental loss of security information by centralizing the storage of application secrets. Key Vaults also control and log the access to anything stored in them. Azure Key Vault can handle requesting and renewing Transport Layer Security (TLS) certificates, providing the features required for a robust certificate lifecycle management solution.

Azure Key Vault in this blueprint

- Holds the Storage access key, with read access granted to the [managed identity](#) of the Customer facing web app
- Holds the SQL Server DBA Password (in a separate vault)
- Diagnostics logging

Monitoring, logging, and audit

Log Analytics

[Log Analytics](#) is a service in Azure that helps you collect and analyze data generated by resources in your cloud and on-premises environments.

Log Analytics in this blueprint

- SQL Assessment
 - Key Vault diagnostics
 - Application Insights connection
 - Azure Activity Log

Application Insights

[Application Insights](#) is an extensible Application Performance Management (APM) service for web developers on multiple platforms. Used to monitor live web applications it will automatically detect performance anomalies, analyze performance, diagnose issues and to understand how users interact with the app. Application Insights can be deployed on platforms including .NET, Node.js and J2EE, hosted on-premises or in the cloud. It integrates with your DevOps process, and has connection points to a variety of development tools.

Application Insights in this blueprint

This template uses the following Application Insights components:

- Application Insights dashboard per site (Operator, Customer and API)

Azure Activity Logs

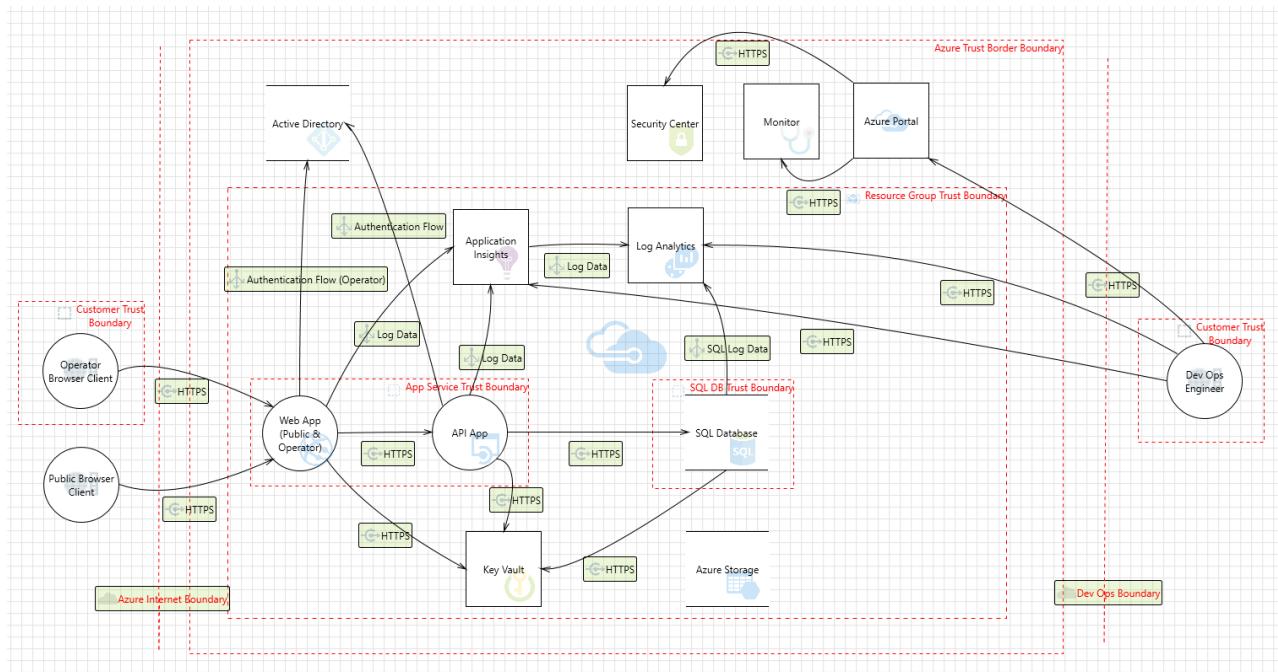
[Azure Activity Log](#) audits control-plane events for your subscriptions. Using the Activity Log, you can determine the 'what, who, and when' for any write operations (PUT, POST, DELETE) taken on the resources in your subscription. You can also understand the status of the operation and other relevant properties.

Azure Monitor

[Azure Monitor](#) enables core monitoring for Azure services by allowing the collection of metrics, activity logs, and diagnostic logs. Azure Monitor provides base-level infrastructure metrics and logs for most services in Microsoft Azure.

Threat model

The data flow diagram for this reference architecture is available for [download](#) or can be found below. This model can help customers understand the points of potential risk in the system infrastructure when making modifications.



The Crown Commercial Service (an agency that works to improve commercial and procurement activity by the government) renewed the classification of Microsoft in-scope enterprise cloud services to G-Cloud v6, covering all its offerings at the OFFICIAL level. Details of Azure and G-Cloud can be found in the [Azure UK G-Cloud security assessment summary](#).

This blueprint aligns to the 14 cloud security principles that are documented in the NCSC [Cloud Security Principles](#) to help ensure an environment that supports workloads classified as UK OFFICIAL.

The [Azure Security and Compliance Blueprint - UK OFFICIAL Customer Responsibility Matrix](#) (Excel Workbook) lists all 14 cloud security principles and denotes, for each principle (or principle subpart), whether the principle implementation is the responsibility of Microsoft, the customer, or shared between the two.

The [Azure Security and Compliance Blueprint - PaaS Web Application for UK OFFICIAL Principle Implementation Matrix](#) (Excel Workbook) lists all 14 cloud security principles and denotes, for each principle (or principle subpart) that is designated a customer responsibility in the Customer Responsibilities Matrix, 1) if the blueprint implements the principle, and 2) a description of how the implementation aligns with the principle requirement(s).

Furthermore, the Cloud Security Alliance (CSA) published the Cloud Control Matrix to support customers in the evaluation of cloud providers and to identify questions that should be answered before moving to cloud services. In response, Microsoft Azure answered the CSA Consensus Assessment Initiative Questionnaire ([CSA CAIQ](#)), which describes how Microsoft addresses the suggested principles.

Third-party assessment

This blueprint has been reviewed by the UK National Cyber Security Centre (NCSC) and aligns to the NCSC 14 Cloud Security Principles

The automation templates have been tested by the UK Customer Success Unit Azure Cloud Solution Architect team and by our Microsoft partner, [Ampliphae](#).

Deploy the solution

This Azure Security and Compliance Blueprint Automation is comprised of JSON configuration files and PowerShell scripts that are handled by Azure Resource Manager's API service to deploy resources within Azure. Detailed deployment instructions are available [here](#).

Three approaches have been provided for deployment; A simple "express" [Azure CLI 2](#) suitable for quickly building a test environment; a parameterized [Azure CLI 2](#) approach providing greater configuration for workload environments; and an Azure portal based deployment where the operator can specify the deployment parameters through the Azure portal.

1. Clone or download [this](#) GitHub repository to your local workstation.
2. Review [Method 1: Azure CLI 2 \(Express version\)](#) and execute the provided commands.
3. Review [Method 1a: Azure CLI 2 \(Configuring the deployment via script arguments\)](#) and execute the provided commands
4. Review [Method 2: Azure Portal Deployment Process](#) and execute the listed commands

Guidance and recommendations

API Management

[Azure API Management](#) could be used in front of the API App Service to provide additional layers of security, throttling and controls to expose, proxy and protect APIs.

Azure B2C

[Azure Active Directory B2C](#) may be implemented as a control to allow users to register, create an identity and

enable authorization and access control for the public web application.

Disclaimer

- This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. Customers reading this document bear the risk of using it.
- This document does not provide customers with any legal rights to any intellectual property in any Microsoft product or solutions.
- Customers may copy and use this document for internal reference purposes.
- Certain recommendations in this document may result in increased data, network, or compute resource usage in Azure, and may increase a customer's Azure license or subscription costs.
- This architecture is intended to serve as a foundation for customers to adjust to their specific requirements and should not be used as-is in a production environment.
- This document is developed as a reference and should not be used to define all means by which a customer can meet specific compliance requirements and regulations. Customers should seek legal support from their organization on approved customer implementations.

Azure Security white papers

12/12/2018 • 3 minutes to read • [Edit Online](#)

Introduction to Azure Security	Explains the collection of security controls implemented in Azure from both the customer's and Microsoft operations' perspectives. Provides a comprehensive look at the customer-facing security controls available with Azure.
Security best practices for Azure solutions	A collection of security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure.
Developing secure applications on Azure	A general guide to the security questions and controls you should consider at each phase of the software development lifecycle when developing applications for the cloud.
Advanced threat detection	Guides you through the Azure approaches towards threat vulnerability assessments, diagnostics, and analysis. Explains how Microsoft uses advanced threat detection mechanisms to secure the platform. Also explains how Microsoft includes these mechanisms in public facing features and services.
Azure data encryption-at-rest	Focuses on how data is protected at rest across Azure and the various components taking part in the data protection implementation. Reviews the pros and cons of the different key management protection approaches.
Azure logging and auditing	Provides an introduction for generating, collecting, and analyzing security logs from services hosted on Azure. These logs can help you gain security insights into your Azure deployments.
Azure network security	Introduces you to the wide range of network controls you can configure to enhance the security of the solutions you deploy in Azure. The focus is customer-facing network security controls.
Azure Functions and serverless platform security	This downloadable white paper covers the benefits of serverless computing while providing security considerations and mitigations in the context of Azure.
Container security in Microsoft Azure	Describes containers, container deployment and management, and native platform services. It also describes runtime security issues that arise with the use of containers on the Azure platform.
Azure operational security	Provides a comprehensive look at the customer-facing operational security technologies and services available with Azure.

Azure security technical capabilities	Focuses on the security features and functionality supporting Azure Storage, Azure SQL Database, the Azure virtual machine model, and the tools and infrastructure that manage it all.
Azure Storage security guide	Provides an overview of each of the security features that can be used with Azure Storage. Covers management plane security, data plane security, encryption at rest, encryption in flight, and storage analytics.
Data classification for cloud readiness	This downloadable paper introduces the fundamentals of data classification and its value in the context of cloud computing. Organizations assessing cloud computing for future use or organizations currently using cloud services and seeking ways to optimize data management will benefit most from this paper.
Governance in Azure	Explains the security and governance features built into Azure. The main governance issues discussed are: policies, processes, and procedures implementation for your organization goals; security and continuous compliance with organization standards; alerting and monitoring.
Isolation in the Azure public cloud	Outlines how Azure provides isolation against both malicious and non-malicious users. Serves as a guide for architecting cloud solutions by offering various isolation choices to architects. Primary focus is on the customer-facing security controls, and does not attempt to address SLAs, pricing models, and DevOps practice considerations.
Overview of Azure compliance	This downloadable paper discusses Azure compliance offerings, including formal certifications, attestations, validations, authorizations, and assessments produced by independent third-party auditing firms, as well as contractual amendments, self-assessments, and customer guidance documents produced by Microsoft. Each offering description states which Azure customer-facing services are in scope for the assessment, and provides links to downloadable resources to assist customers with their own compliance obligations.
Security management in Azure	Discusses issues in the remote access of Azure resources. The nature of the cloud demands remote access administration and, therefore, security is paramount. Covers general security guidelines, client configuration, best practices, and operational principles and procedures.

Security services and technologies available on Azure

10/17/2018 • 3 minutes to read • [Edit Online](#)

In our discussions with current and future Azure customers, we're often asked "do you have a list of all the security-related services and technologies that Azure has to offer?"

When you evaluate cloud service provider options, it's helpful to have this information. So we have provided this list to get you started.

Over time, this list will change and grow, just as Azure does. Make sure to check this page on a regular basis to stay up-to-date on our security-related services and technologies.

General Azure security

Service	Description
Azure Security Center	A cloud workload protection solution that provides security management and advanced threat protection across hybrid cloud workloads.
Azure Key Vault	A secure secrets store for the passwords, connection strings, and other information you need to keep your apps working.
Log Analytics	A monitoring service that collects telemetry and other data, and provides a query language and analytics engine to deliver operational insights for your apps and resources. Can be used alone or with other services such as Security Center.
Azure Dev/Test Labs	A service that helps developers and testers quickly create environments in Azure while minimizing waste and controlling cost.

Storage security

Service	Description
Azure Storage Service Encryption	A security feature that automatically encrypts your data in Azure storage.
StorSimple Encrypted Hybrid Storage	An integrated storage solution that manages storage tasks between on-premises devices and Azure cloud storage.
Azure Client-Side Encryption	A client-side encryption solution that encrypts data inside client applications before uploading to Azure Storage; also decrypts the data while downloading.
Azure Storage Shared Access Signatures	A shared access signature provides delegated access to resources in your storage account.
Azure Storage Account Keys	An access control method for Azure storage that is used for authentication when the storage account is accessed.

Service	Description
Azure File shares with SMB 3.0 Encryption	A network security technology that enables automatic network encryption for the Server Message Block (SMB) file sharing protocol.
Azure Storage Analytics	A logging and metrics-generating technology for data in your storage account.

Database security

Service	Description
Azure SQL Firewall	A network access control feature that protects against network-based attacks to database.
Azure SQL Cell Level Encryption	A database security technology that provides encryption at a granular level.
Azure SQL Connection Encryption	To provide security, SQL Database controls access with firewall rules limiting connectivity by IP address, authentication mechanisms requiring users to prove their identity, and authorization mechanisms limiting users to specific actions and data.
Azure SQL Always Encryption	Protects sensitive data, such as credit card numbers or national identification numbers (for example, U.S. social security numbers), stored in Azure SQL Database or SQL Server databases.
Azure SQL Transparent Data Encryption	A database security feature that encrypts the storage of an entire database.
Azure SQL Database Auditing	A database auditing feature that tracks database events and writes them to an audit log in your Azure storage account.

Identity and access management

Service	Description
Azure Role Based Access Control	An access control feature designed to allow users to access only the resources they are required to access based on their roles within the organization.
Azure Active Directory	A cloud-based authentication repository that supports a multi-tenant, cloud-based directory and multiple identity management services within Azure.
Azure Active Directory B2C	An identity management service that enables control over how customers sign-up, sign-in, and manage their profiles when using Azure-based applications.
Azure Active Directory Domain Services	A cloud-based and managed version of Active Directory Domain Services.

SERVICE	DESCRIPTION
Azure Multi-Factor Authentication	A security provision that employs several different forms of authentication and verification before allowing access to secured information.

Backup and disaster recovery

SERVICE	DESCRIPTION
Azure Backup	An Azure-based service used to back up and restore data in the Azure cloud.
Azure Site Recovery	An online service that replicates workloads running on physical and virtual machines (VMs) from a primary site to a secondary location to enable recovery of services after a failure.

Networking

SERVICE	DESCRIPTION
Network Security Groups	A network-based access control feature using a 5-tuple to make allow or deny decisions.
Azure VPN Gateway	A network device used as a VPN endpoint to allow cross-premises access to Azure Virtual Networks.
Azure Application Gateway	An advanced web application load balancer that can route based on URL and perform SSL-offloading.
Azure Load Balancer	A TCP/UDP application network load balancer.
Azure ExpressRoute	A dedicated WAN link between on-premises networks and Azure Virtual Networks.
Azure Traffic Manager	A global DNS load balancer.
Azure Application Proxy	An authenticating front-end used to secure remote access for web applications hosted on-premises.

Azure Security technical overviews

9/20/2018 • 2 minutes to read • [Edit Online](#)

- [Azure database security overview](#)
- [Azure encryption overview](#)
- [Azure identity management security overview](#)
- [Azure network security overview](#)
- [Azure operational security overview](#)
- [Azure security management and monitoring overview](#)
- [Azure Service Fabric security overview](#)
- [Azure Storage security overview](#)
- [Azure virtual machines security overview](#)
- [Internet of Things security architecture](#)
- [Introduction to Azure security](#)
- [Introduction to Azure log integration](#)
- [Secure your Internet of Things in Azure](#)
- [Securing PaaS deployments](#)
- [What is Azure Security Center?](#)

Azure security best practices and patterns

9/10/2018 • 2 minutes to read • [Edit Online](#)

The articles below contain security best practices to use when you're designing, deploying, and managing your cloud solutions by using Azure. These best practices come from our experience with Azure security and the experiences of customers like you.

The best practices are intended to be a resource for IT pros. This might include designers, architects, developers, and testers who build and deploy secure Azure solutions.

- [Azure boundary security best practices](#)
- [Azure database security best practices](#)
- [Azure data security and encryption best practices](#)
- [Azure identity management and access control security best practices](#)
- [Azure network security best practices](#)
- [Azure operational security best practices](#)
- [Azure PaaS Best Practices](#)
- [Azure Service Fabric security best practices](#)
- [Best practices for Azure VM security](#)
- [Implementing a secure hybrid network architecture in Azure](#)
- [Internet of Things security best practices](#)
- [Securing PaaS databases in Azure](#)
- [Securing PaaS web and mobile applications using Azure App Service](#)
- [Securing PaaS web and mobile applications using Azure Storage](#)
- [Security best practices for IaaS workloads in Azure](#)

The white paper [Security best practices for Azure solutions](#) is a collection of the security best practices found in the articles listed above.

[Download the white paper](#)

Azure Security MVP Program

4/26/2018 • 2 minutes to read • [Edit Online](#)

Microsoft Most Valuable Professionals (MVPs) are community leaders who have demonstrated an exemplary commitment to helping others get the most out of their experience with Microsoft technologies. They share their passion, real-world knowledge, and technical expertise with the community and with Microsoft.

Microsoft Azure now recognizes community experts with special expertise in Azure security. Microsoft MVPs can be awarded the MVP in Microsoft Azure in the Azure Security contribution area.

Award Category	Microsoft Azure	Windows Development	Office Development	Visual Studio and Development Technologies	Data Platform
Contribution Areas	<ul style="list-style-type: none">• Azure App Service• Azure Media Service & CDN• IoT on Azure and Azure Messaging (Event Hub and Service Bus)• Azure Cloud Service• Azure Service Fabric• Application Integration• Azure Virtual Machines (IaaS) and Batch• Azure Storage• Azure Networking• Azure Backup & Recovery• Azure Security• Linux and Docker on Azure• DevOps on Azure (Chef, Puppet, Salt, Ansible, Dev/Test Lab)• SDK support on Azure (.NET, Node.js, Java, PHP, Python, GO, Ruby)	<ul style="list-style-type: none">• Windows App Development• Classic Windows Development• Windows Bridges• Windows On Devices (IoT /Embedded)• Windows Hardware Engineering• Emerging Experiences (More Personal Computing)	<ul style="list-style-type: none">• Office Add-in Development• O365 API Development• SharePoint Add-in Development• Office Development for iOS• Office Development for Android• Office Development with PHP• Office Development with Node.js• Office Development with Angularjs	<ul style="list-style-type: none">• ASP.NET/IIS• .NET• Visual C++• Visual Studio ALM• Developer Security• Visual Studio Extensibility• Front End Web Dev• Node.js• PHP• Python• Java• Unity• Xamarin• Cordova• JavaScript/TypeScript• Grunt/Gulp• CSS3• Clang/LLVM	<ul style="list-style-type: none">• Analytics Platform System• Azure Data Lake• Azure DocumentDB• Azure HDInsight and Hadoop, Spark, & Storm on Azure• Azure Machine Learning• Azure Search• Azure SQL Data Warehouse• Azure SQL Database• Azure Stream Analytics• Cortana Analytics Suite• Information Management (ADF, SSIS, & Data Sync)• Power BI• SQL Server• SQL Server Reporting Services & Analysis Services

While there is no benchmark for becoming an MVP, in part because it varies by technology and its life-cycle, some of the criteria includes the impact of a nominee's contributions to online forums such as Microsoft Answers, TechNet and MSDN; wikis and online content; conferences and user groups; podcasts, Web sites, blogs and social media; and articles and books.

Are you an expert in Azure security? Do you know someone who is? Then [Nominate yourself or someone else](#) to become an Azure security MVP today!

Microsoft Services in Cybersecurity

11/7/2018 • 2 minutes to read • [Edit Online](#)

Microsoft Services provides a comprehensive approach to security, identity and cybersecurity. Microsoft Services provides an array of Security and Identity services across strategy, planning, implementation, and ongoing support. These services can help Enterprise customers implement holistic security solutions that align with their strategic goals.

With direct access to product development teams, we can create solutions that integrate, and enhance the latest security and identity capabilities of our products to help protect our customer's business and drive innovation.

Entrusted with helping protect and enable the world's largest organizations, our diverse group of technical professionals consists of highly trained experts who offer a wealth of security and identity experience.

Learn more about services provided by Microsoft Services:

- [Security Risk Assessment](#)
- [Dynamic Identity Framework Assessment](#)
- [Offline Assessment for Active Directory Services](#)
- [Enhanced Security Administration Environment](#)
- [Azure AD Implementation Services](#)
- [Securing Against Lateral Account Movement](#)
- [Microsoft Threat Detection Services](#)
- [Incident Response and Recovery](#)

[Learn more](#) about Microsoft Services Security consulting services.

How to Log a Security Event Support Ticket

12/4/2017 • 2 minutes to read • [Edit Online](#)

1. Navigate to [Publisher Support](#) and sign in with your Microsoft credentials.
2. Select "Security Event" as the Problem Type and choose between the "Security Incident" and "Vulnerability" categories.

Event Type	Definition
Security Incident	Security incidents include, but are not limited to: <ul style="list-style-type: none">• Attempts to gain unauthorized access to a system or its data• Denial of service attacks• Misuse or abuse of a system or data in violation of security policy• Malware incidents that disrupt services, steal or attempt to steal sensitive information, gain access or attempt to gain access to private computer systems, etc.
Vulnerability	Vulnerabilities are defects that make an incident possible but that have not yet been exploited.

3. After you select the Problem Type and Category, click the '**Start request**' button. Provide the following information to help us better understand the issue.
 - i. What is the problem and/or vulnerability?
 - ii. For vulnerabilities, please provide the CVE (mitre.org) or the filled out CVSS3 v3 calculator (<https://www.first.org/cvss/calculator/3.0>).
 - iii. Is there a resolution or mitigation? If yes, then please provide the remediation steps.
 - iv. Do you have a message that you want to send to customers? We will work with you to craft an appropriate message if applicable.
4. Submission confirmation - Once you have submitted your issue, we will acknowledge receipt within one business day and assign your issue a priority and severity.
 - If you need to communicate with us about your issue, use the confirmation number in all correspondence.
 - You can view progress on your issue at any time.
5. What happens next? Depending on the issue and severity, the following steps may be taken:
 - We will communicate the outcome of our assessment to you. Depending on the outcome, we may remove or request that you modify your offering. In this event, we will work with you to ensure that disruption to impacted customers is minimized.
 - We will work with you to help mitigate the impact of the incident/vulnerability for our mutual customers.

Pen Testing

8/13/2018 • 2 minutes to read • [Edit Online](#)

One of the benefits of using Azure for application testing and deployment is that you can quickly get environments created. You don't have to worry about requisitioning, acquiring, and "racking and stacking" your own on-premises hardware.

This is great – but you still need to make sure you perform your normal security due diligence. One of the things you need to do is penetration test the applications you deploy in Azure.

You might already know that Microsoft performs [penetration testing of our Azure environment](#). This helps drive Azure improvements.

We don't pen test your application for you, but we do understand that you will want and need to perform pen testing on your own applications. That's a good thing, because when you enhance the security of your applications, you help make the entire Azure ecosystem more secure.

What to do?

As of June 15, 2017, Microsoft no longer requires pre-approval to conduct a penetration tests against Azure resources. Customers who wish to formally document upcoming penetration testing engagements against Microsoft Azure are encouraged to fill out the [Azure Service Penetration Testing Notification form](#). This process is only related to Microsoft Azure, and not applicable to any other Microsoft Cloud Service.

IMPORTANT

While notifying Microsoft of pen testing activities is no longer required customers must still comply with the [Microsoft Cloud Unified Penetration Testing Rules of Engagement](#).

Standard tests you can perform include:

- Tests on your endpoints to uncover the [Open Web Application Security Project \(OWASP\) top 10 vulnerabilities](#)
- [Fuzz testing](#) of your endpoints
- [Port scanning](#) of your endpoints

One type of test that you can't perform is any kind of [Denial of Service \(DoS\)](#) attack. This includes initiating a DoS attack itself, or performing related tests that might determine, demonstrate or simulate any type of DoS attack.

Next steps

- Are you ready to get started with pen testing your applications hosted in Microsoft Azure? If so, then head on over to the [Penetration Testing Rules of Engagement](#) and fill out the testing notification form.

Microsoft Threat Modeling Tool

8/30/2018 • 2 minutes to read • [Edit Online](#)

The Threat Modeling Tool is a core element of the Microsoft Security Development Lifecycle (SDL). It allows software architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve. As a result, it greatly reduces the total cost of development. Also, we designed the tool with non-security experts in mind, making threat modeling easier for all developers by providing clear guidance on creating and analyzing threat models.

The tool enables anyone to:

- Communicate about the security design of their systems
- Analyze those designs for potential security issues using a proven methodology
- Suggest and manage mitigations for security issues

Here are some tooling capabilities and innovations, just to name a few:

- **Automation:** Guidance and feedback in drawing a model
- **STRIDE per Element:** Guided analysis of threats and mitigations
- **Reporting:** Security activities and testing in the verification phase
- **Unique Methodology:** Enables users to better visualize and understand threats
- **Designed for Developers and Centered on Software:** many approaches are centered on assets or attackers. We are centered on software. We build on activities that all software developers and architects are familiar with -- such as drawing pictures for their software architecture
- **Focused on Design Analysis:** The term "threat modeling" can refer to either a requirements or a design analysis technique. Sometimes, it refers to a complex blend of the two. The Microsoft SDL approach to threat modeling is a focused design analysis technique

Next steps

The table below contains important links to get you started with the Threat Modeling Tool:

STEP	DESCRIPTION
1	Download the Threat Modeling Tool
2	Read Our getting started guide
3	Get familiar with the features
4	Learn about generated threat categories
5	Find mitigations to generated threats

Resources

Here are a few older articles still relevant to threat modeling today:

- [Article on the Importance of Threat Modeling](#)
- [Training Published by Trustworthy Computing](#)

Check out what a few Threat Modeling Tool experts have done:

- [Threats Manager](#)
- [Simone Curzi Security Blog](#)

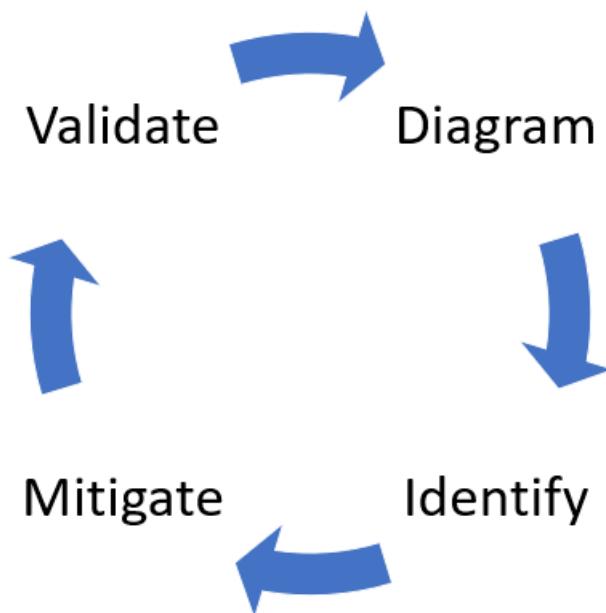
Getting started with the Threat Modeling Tool

6/13/2018 • 7 minutes to read • [Edit Online](#)

The Cloud and Enterprise Security Tools team released the Threat Modeling Tool Preview earlier this year as a free [click-to-download](#). The change in delivery mechanism allows us to push the latest improvements and bug fixes to customers each time they open the tool, making it easier to maintain and use. This article takes you through the process of getting started with the Microsoft SDL threat modeling approach and shows you how to use the tool to develop great threat models as a backbone of your security process.

This article builds on existing knowledge of the SDL threat modeling approach. For a quick review, refer to [Threat Modeling Web Applications](#) and an archived version of [Uncover Security Flaws Using the STRIDE Approach](#) MSDN article published in 2006.

To quickly summarize, the approach involves creating a diagram, identifying threats, mitigating them and validating each mitigation. Here's a diagram that highlights this process:



Starting the threat modeling process

When you launch the Threat Modeling Tool, you'll notice a few things, as seen in the picture:

MICROSOFT MICROSOFT THREAT MODELING TOOL (PREVIEW)

Threat Model:

[Feedback, Suggestions and Issues](#)

Create A Model
Model your system by drawing diagram(s). Make sure you capture important details.

Open A Model
Open an existing model and analyze threats against your system; do not worry, the tool will help you identify them.

Getting Started Guide
A step-by-step guide to help you get up and running now.

Template For New Models
Recently Opened Models
Threat Modeling Workflow

Azure Threat Model Template(1.0.0.20)

[Basic Web App NEW.tm7](#)
[New Threat Model.tm7](#)
[Library Sample.tm7](#)
[Basic Web App Sample.tm7](#)
[QPP_complete19_filtered.tm7](#)
[CloudMobileThreatModel_April2017.tm7](#)

Template:

Create New Template
Define stencils, threat types and custom threat properties for your threat model from scratch.

Open Template
Open an existing Template and make modifications to better suit your specific threat analysis.

Template Workflow
Use templates to define threats that applications should look for.
1. Define stencils
2. Define categories
3. Define threat properties
4. Define threat
5. Share your template

Threat model section

COMPONENT	DETAILS
Feedback, Suggestions and Issues Button	Takes you to the MSDN Forum for all things SDL. It gives you an opportunity to read through what other users are doing, along with workarounds and recommendations. If you still can't find what you're looking for, email tmtextsupport@microsoft.com for our support team to help you
Create a Model	Opens a blank canvas for you to draw your diagram. Make sure to select which template you'd like to use for your model
Template for New Models	You must select which template to use before creating a model. Our main template is the Azure Threat Model Template, which contains Azure-specific stencils, threats and mitigations. For generic models, select the SDL TM Knowledge Base from the drop-down menu. Want to create your own template or submit a new one for all users? Check out our Template Repository GitHub Page to learn more
Open a Model	Opens previously saved threat models. The Recently Opened Models feature is great if you need to open your most recent files. When you hover over the selection, you'll find 2 ways to open models: <ul style="list-style-type: none"> • Open From this Computer – classic way of opening a file using local storage • Open from OneDrive – teams can use folders in OneDrive to save and share all their threat models in a single location to help increase productivity and collaboration

COMPONENT	DETAILS
Getting Started Guide	Opens the Microsoft Threat Modeling Tool main page

Template section

COMPONENT	DETAILS
Create New Template	Opens a blank template for you to build on. Unless you have extensive knowledge in building templates from scratch, we recommend you to build from existing ones
Open Template	Opens existing templates for you to make changes to

The Threat Modeling Tool team is constantly working to improve tool functionality and experience. A few minor changes might take place over the course of the year, but all major changes require rewrites in the guide. Refer to it often to ensure you get the latest announcements.

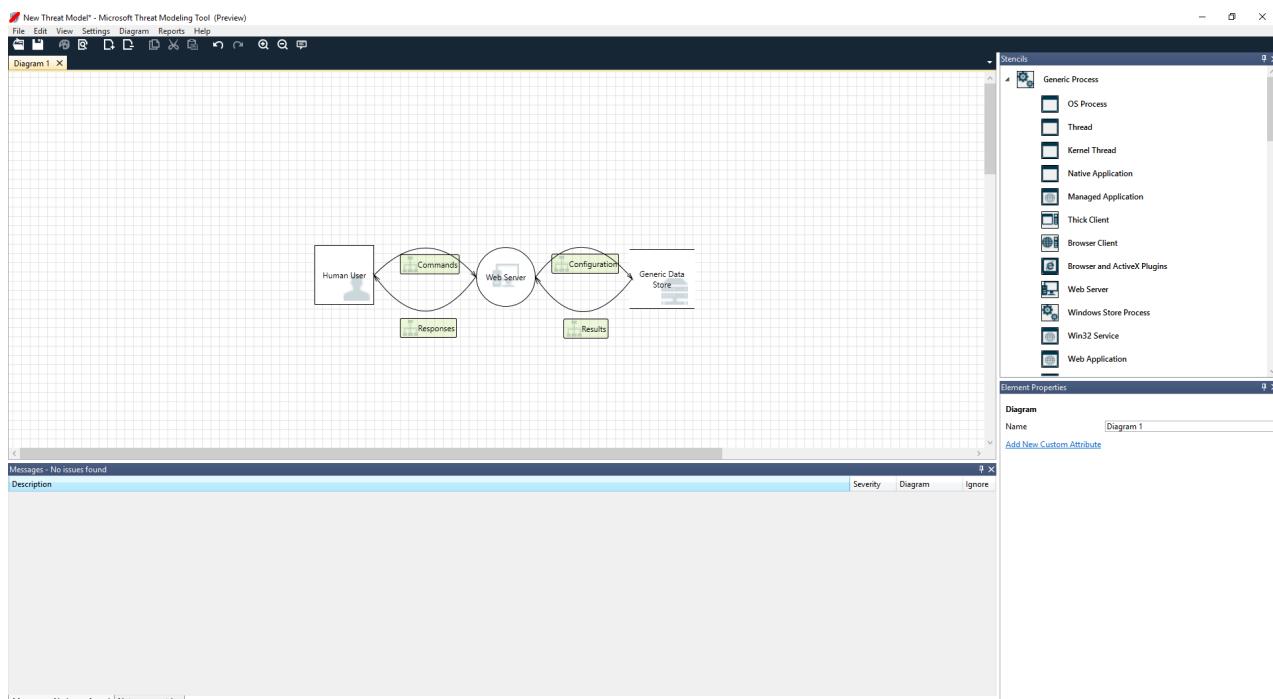
Building a model

In this section, we follow:

- Cristina (a developer)
- Ricardo (a program manager) and
- Ashish (a tester)

They are going through the process of developing their first threat model.

Ricardo: Hi Cristina, I worked on the threat model diagram and wanted to make sure we got the details right. Can you help me look it over? Cristina: Absolutely. Let's take a look. Ricardo opens the tool and shares his screen with Cristina.



Cristina: Ok, looks straightforward, but can you walk me through it? Ricardo: Sure! Here is the breakdown:

- Our human user is drawn as an outside entity—a square

- They're sending commands to our Web server—the circle
- The Web server is consulting a database (two parallel lines)

What Ricardo just showed Cristina is a DFD, short for **Data Flow Diagram**. The Threat Modeling Tool allows users to specify trust boundaries, indicated by the red dotted lines, to show where different entities are in control. For example, IT administrators require an Active Directory system for authentication purposes, so the Active Directory is outside of their control.

Cristina: Looks right to me. What about the threats? Ricardo: Let me show you.

Analyzing threats

Once he clicks on the analysis view from the icon menu selection (file with magnifying glass), he is taken to a list of generated threats the Threat Modeling Tool found based on the default template, which uses the SDL approach called **STRIDE (Spoofing, Tampering, Info Disclosure, Repudiation, Denial of Service and Elevation of Privilege)**. The idea is that software comes under a predictable set of threats, which can be found using these 6 categories.

This approach is like securing your house by ensuring each door and window has a locking mechanism in place before adding an alarm system or chasing after the thief.

The screenshot shows the Microsoft Threat Modeling Tool interface. At the top, there's a toolbar with various icons for file operations, settings, and help. Below the toolbar is a navigation bar with tabs like 'Diagram' and 'Threat List'. The main area contains a 'Diagram 1' section with a Data Flow Diagram (DFD) and a 'Threat List' section below it.

Data Flow Diagram:

```

graph LR
    HumanUser[Human User] -- Commands --> WebServer[Web Server]
    WebServer -- Configuration --> GenericDataStore[Generic Data Store]
    HumanUser -- Responses --> Results[Results]
    
```

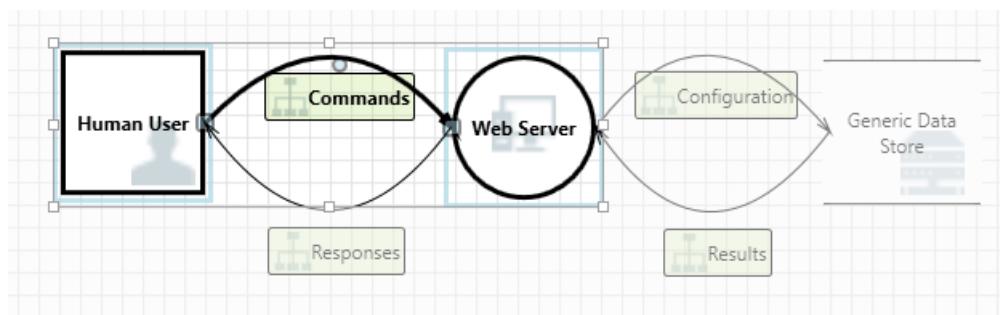
Threat List:

ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
0	Diagram 1	Generated	Not Started	Spoofing the...	Spoofing	Human User...	Human User...	Commands	High	
1	Diagram 1	Generated	Not Started	Cross Site Scr...	Tampering	The web serv...	The web serv...	Commands	High	
2	Diagram 1	Generated	Not Started	Elevation Us...	Elevation Of...	Web Server...	Web Server...	Commands	High	
3	Diagram 1	Generated	Not Started	Spoofing of D...	Spoofing	Generic Data...	Generic Data...	Configuration	High	
4	Diagram 1	Generated	Not Started	Potential Deny...	Deny Of Ser...	Does Web Se...	Generic Data...	Configuration	High	
5	Diagram 1	Generated	Not Started	Spoofing of S...	Spoofing	Generic Data...	Generic Data...	Results	High	
6	Diagram 1	Generated	Not Started	Cross Site Scr...	Tampering	The web serv...	The web serv...	Results	High	
7	Diagram 1	Generated	Not Started	Persistent Cr...	Tampering	The web serv...	The web serv...	Results	High	
8	Diagram 1	Generated	Not Started	Weak Access...	Information...	Improper dat...	Improper dat...	Results	High	

Below the Threat List, a message says '9 Threats Displayed, 9 Total'. A separate 'Threat Properties' window is open, showing a message 'No threats are selected'.

Ricardo begins by selecting the first item on the list. Here's what happens:

First, the interaction between the two stencils is enhanced



Second, additional information about the threat appears in the Threat Properties window

Threat Properties	
ID: 0	Diagram: Diagram 1
Status: Not Started	
Title:	Spoofing the Human User External Entity
Category:	Spoofing
Description:	Human User may be spoofed by an attacker and this may lead to unauthorized access to Web Server. Consider using a standard authentication mechanism to identify the external entity.
Justification:	
Interaction:	Commands
Priority:	High

The generated threat helps him understand potential design flaws. The STRIDE categorization gives him an idea on potential attack vectors, while the additional description tells him exactly what's wrong, along with potential ways to mitigate it. He can use editable fields to write notes in the justification details or change priority ratings depending on his organization's bug bar.

Azure templates have additional details to help users understand not only what's wrong, but also how to fix it by adding descriptions, examples and hyperlinks to Azure-specific documentation.

The description made him realize the importance of adding an authentication mechanism to prevent users from being spoofed, revealing the first threat to be worked on. A few minutes into the discussion with Cristina, they understood the importance of implementing access control and roles. Ricardo filled in some quick notes to make sure these were implemented.

As Ricardo went into the threats under Information Disclosure, he realized the access control plan required some read-only accounts for audit and report generation. He wondered whether this should be a new threat, but the mitigations were the same, so he noted the threat accordingly. He also thought about information disclosure a bit more and realized that the backup tapes were going to need encryption, a job for the operations team.

Threats not applicable to the design due to existing mitigations or security guarantees can be changed to "Not Applicable" from the Status drop-down. There are three other choices: Not Started – default selection, Needs Investigation – used to follow up on items and Mitigated – once it's fully worked on.

Reports & sharing

Once Ricardo goes through the list with Cristina and adds important notes, mitigations/justifications, priority and status changes, he selects Reports -> Create Full Report -> Save Report, which prints out a nice report for him to go through with colleagues to ensure the proper security work is implemented.

Threat Modeling Report

Created on 7/31/2017 12:35:42 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0

Diagram: Diagram 1

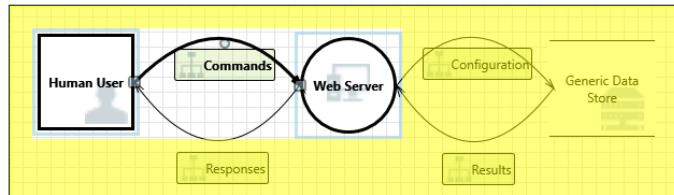
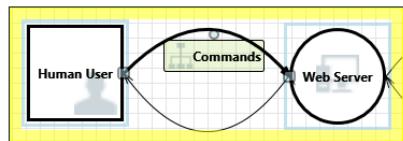


Diagram 1 Diagram Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0

Interaction: Commands



1. Spoofing the Human User External Entity [State: Not Started] [Priority: High]

Category:	Spoofing
Description:	Human User may be spoofed by an attacker and this may lead to unauthorized access to Web Server. Consider using a standard authentication mechanism to identify the external entity.
Justification:	<no mitigation provided>
Possible Mitigation(s):	
SDL Phase:	Design

2. Cross Site Scripting [State: Not Started] [Priority: High]

Category:	Tampering
Description:	The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification:	<no mitigation provided>
Possible Mitigation(s):	
SDL Phase:	Design

If Ricardo wants to share the file instead, he can easily do so by saving in his organization's OneDrive account. Once he does that, he can copy the document link and share it with his colleagues.

Threat modeling meetings

When Ricardo sent his threat model to his colleague using OneDrive, Ashish, the tester, was underwhelmed. Seemed like Ricardo and Cristina missed quite a few important corner cases, which could be easily compromised. His skepticism is a complement to threat models.

In this scenario, after Ashish took over the threat model, he called for two threat modeling meetings: one meeting to synchronize on the process and walk through the diagrams and then a second meeting for threat review and sign-off.

In the first meeting, Ashish spent 10 minutes walking everyone through the SDL threat modeling process. He then pulled up the threat model diagram and started explaining it in detail. Within five minutes, an important missing component had been identified.

A few minutes later, Ashish and Ricardo got into an extended discussion of how the Web server was built. It was not the ideal way for a meeting to proceed, but everyone eventually agreed that discovering the discrepancy early was going to save them time in the future.

In the second meeting, the team walked through the threats, discussed some ways to address them, and signed off on the threat model. They checked the document into source control and continued with development.

Thinking about assets

Some readers who have threat modeled may notice that we haven't talked about assets at all. We've discovered that many software engineers understand their software better than they understand the concept of assets and what assets an attacker may be interested in.

If you're going to threat model a house, you might start by thinking about your family, irreplaceable photos or valuable artwork. Perhaps you might start by thinking about who might break in and the current security system. Or you might start by considering the physical features, like the pool or the front porch. These are analogous to thinking about assets, attackers, or software design. Any of these three approaches work.

The approach to threat modeling we've presented here is substantially simpler than what Microsoft has done in the past. We found that the software design approach works well for many teams. We hope that include yours.

Next Steps

Send your questions, comments and concerns to tmtextsupport@microsoft.com. [Download](#) the Threat Modeling Tool to get started.

Threat Modeling Tool feature overview

12/18/2017 • 5 minutes to read • [Edit Online](#)

The Threat Modeling Tool can help you with your threat modeling needs. For a basic introduction to the tool, see [Get started with the Threat Modeling Tool](#).

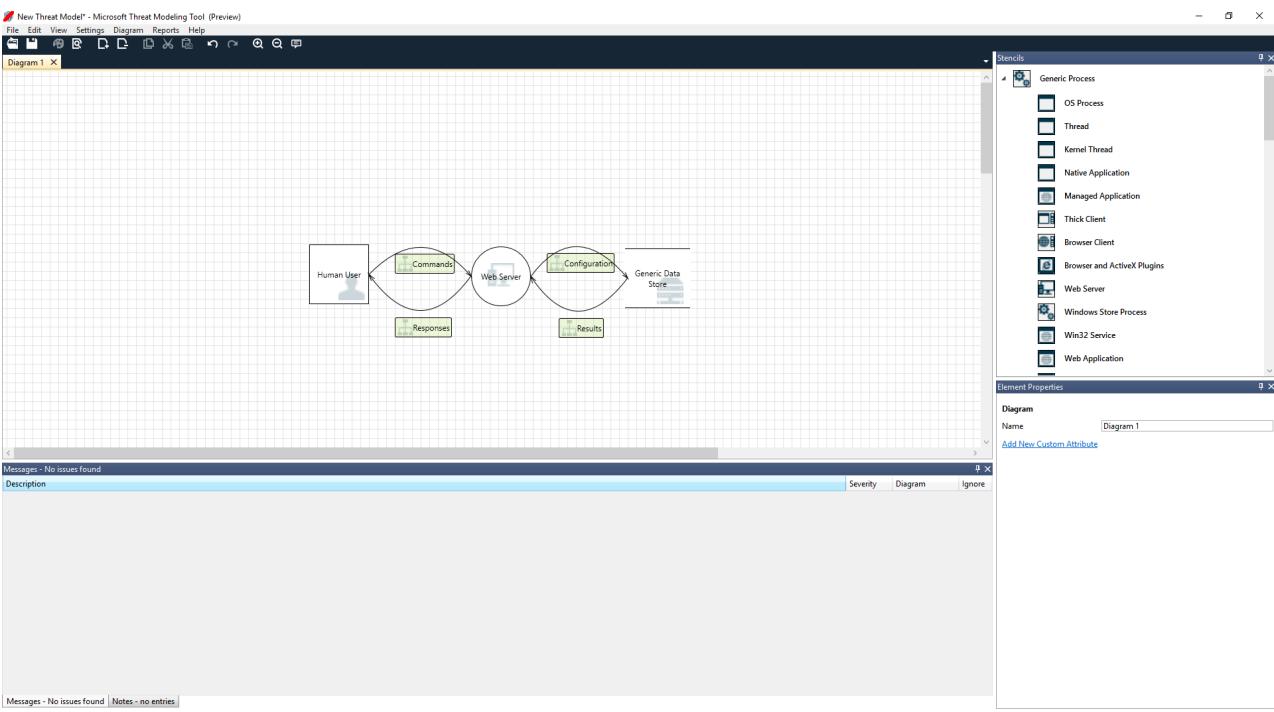
NOTE

The Threat Modeling Tool is updated frequently, so check this guide often to see our latest features and improvements.

To open a blank page, select **Create A Model**.

The screenshot shows the Microsoft Threat Modeling Tool (Preview) interface. At the top, there's a header bar with the title "MICROSOFT THREAT MODELING TOOL (PREVIEW)". Below the header, there are two main sections: "Threat Model:" and "Template:". The "Threat Model:" section contains three cards: "Create A Model" (described as drawing a diagram to model the system), "Open A Model" (described as opening an existing model to analyze threats), and "Getting Started Guide" (a step-by-step guide to help get up and running). The "Template:" section also contains three cards: "Template For New Models" (with a dropdown menu showing "Azure Threat Model Template(1.0.0.20)" and a "Browse..." button), "Recently Opened Models" (listing files like "Basic Web App NEW.tm7", "New Threat Model.tm7", etc.), and "Threat Modeling Workflow" (a 4-step process: 1. Select your template, 2. Create your data flow diagram model, 3. Analyze the model for potential threats, 4. Determine mitigations). Below these sections, there are two more cards: "Create New Template" (for defining stencils, threat types, and custom threat properties) and "Open Template" (for opening an existing template to make modifications). To the right of the "Open Template" card is a "Template Workflow" section with steps: 1. Use templates to define threats that applications should look for, 2. Define stencils, 3. Define categories, 4. Define threat properties, 5. Share your template.

To see the features currently available in the tool, use the threat model created by our team in the [Get started example](#).



Navigation

Before we discuss the built-in features, let's review the main components found in the tool.

Menu items

The experience is similar to other Microsoft products. Let's review the top-level menu items.



LABEL	DETAILS
File	<ul style="list-style-type: none"> Open, save, and close files Sign in and sign out of OneDrive accounts. Share links (view and edit). View file information. Apply a new template to existing models.
Edit	Undo and redo actions, as well as copy, paste, and delete.
View	<ul style="list-style-type: none"> Switch between Analysis and Design views. Open closed windows (for example, stencils, element properties, and messages). Reset layout to default settings.
Diagram	Add and delete diagrams, and move through tabs of diagrams.
Reports	Create HTML reports to share with others.
Help	Find guides to help you use the tool.

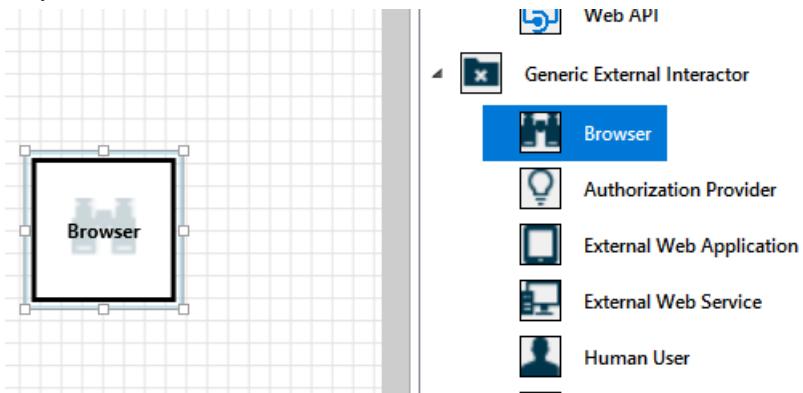
The symbols are shortcuts for the top-level menus:

SYMBOL	DETAILS
Open	Opens a new file.
Save	Saves the current file.
Design	Opens the Design view, where you can create models.
Analyze	Shows generated threats and their properties.
Add diagram	Adds a new diagram (similar to new tabs in Excel).
Delete diagram	Deletes the current diagram.
Copy/Cut/Paste	Copies, cuts, and pastes elements.
Undo/Redo	Undoes and redoes actions.
Zoom in/Zoom out	Zooms in and out of the diagram for a better view.
Feedback	Opens the MSDN Forum.

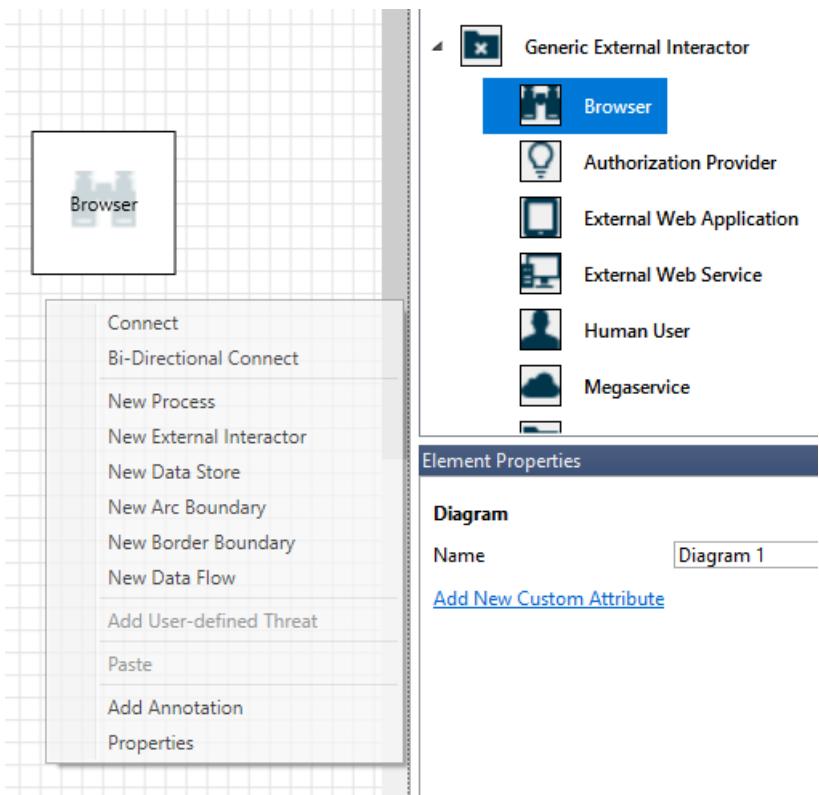
Canvas

The canvas is the space where you drag and drop elements. Drag and drop is the quickest and most efficient way to build models. You can also right-click and select items from the menu to add generic versions of elements, as shown:

Drop the stencil on the canvas



Select the stencil



Stencils

Based on the template you select, you can find all the stencils available to use. If you can't find the right elements, use another template. Or you can modify a template to fit your needs. Generally, you can find a combination of categories like these:

STENCIL NAME	DETAILS
Process	Applications, browser plug-ins, threads, virtual machines
External interactor	Authentication providers, browsers, users, web applications
Data store	Cache, storage, configuration files, databases, registry
Data flow	Binary, ALPC, HTTP, HTTPS/TLS/SSL, IOCTL, IPSec, named pipe, RPC/DCOM, SMB, UDP
Trust line/Border boundary	Corporate networks, internet, machine, sandbox, user/kernel mode

Notes/messages

COMPONENT	DETAILS
Messages	Internal tool logic that alerts users whenever there's an error, such as no data flows between elements.
Notes	Manual notes are added to the file by engineering teams throughout the design and review process.

Element properties

Element properties vary by the elements you select. Apart from trust boundaries, all other elements contain three general selections:

ELEMENT PROPERTY	DETAILS
Name	Useful for naming your processes, stores, interactors, and flows so that they're easily recognized.
Out of scope	If selected, the element is taken out of the threat-generation matrix (not recommended).
Reason for out of scope	Justification field to let users know why out of scope was selected.

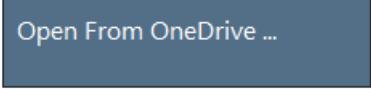
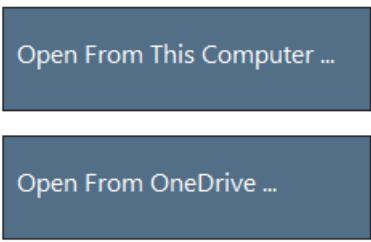
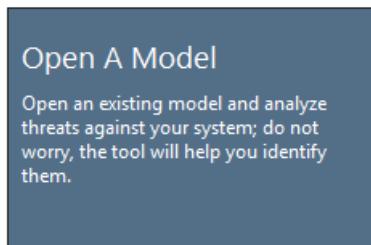
Properties are changed under each element category. Select each element to inspect the available options. Or you can open the template to learn more. Let's review the features.

Welcome screen

When you open the app, you see the **Welcome** screen.

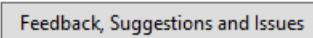
Open a model

Hover over **Open A Model** to reveal two options: **Open From This Computer** and **Open From OneDrive**. The first option opens the **File Open** screen. The second option takes you through the sign-in process for OneDrive. After successful authentication, you can select folders and files.



Feedback, suggestions, and issues

When you select **Feedback, Suggestions and Issues**, you go to the MSDN Forum for SDL Tools. You can read what other people are saying about the tool, including workarounds and new ideas.



Design view

When you open or create a new model, the **Design** view opens.

Add elements

You can add elements on the grid in two ways:

- **Drag and drop:** Drag the desired element to the grid. Then use the element properties to provide additional information.
- **Right-click:** Right-click anywhere on the grid, and select items from the drop-down menu. A generic

representation of the element you select appears on the screen.

Connect elements

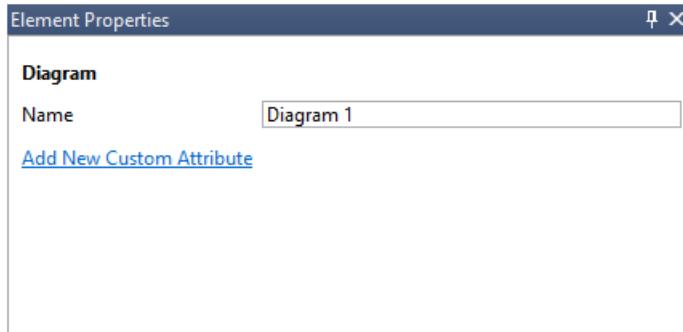
You can connect elements in two ways:

- **Drag and drop:** Drag the desired dataflow to the grid, and connect both ends to the appropriate elements.
- **Click + Shift:** Click the first element (sending data), press and hold the Shift key, and then select the second element (receiving data). Right-click, and select **Connect**. If you use a bi-directional data flow, the order is not as important.

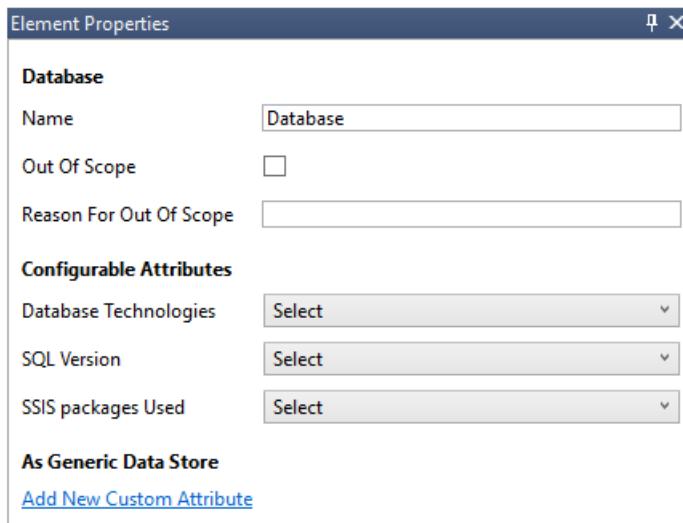
Properties

To see the properties that can be modified on the stencils, select the stencil and the information populates accordingly. The following example shows before and after a **Database** stencil is dragged onto the diagram:

Before

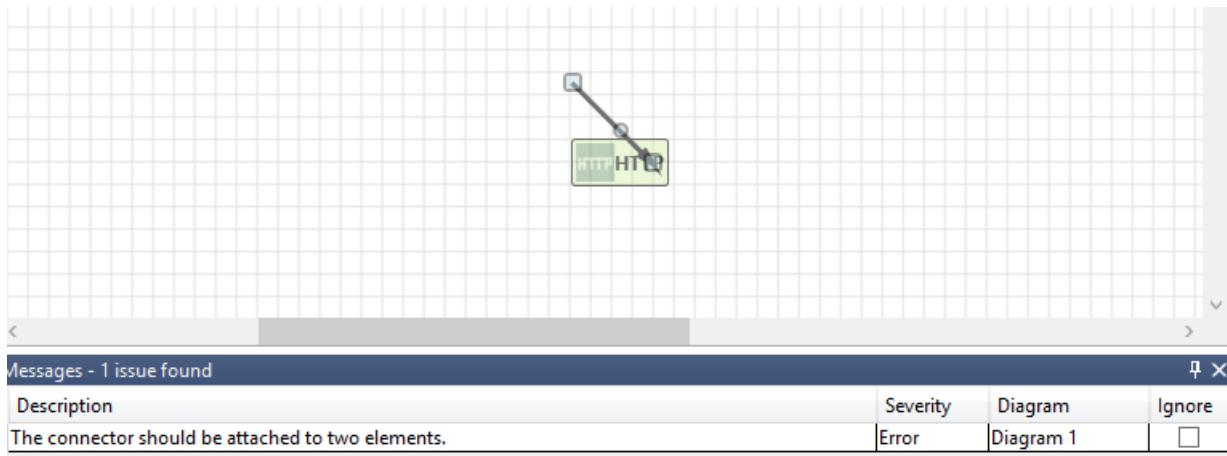


After



Messages

If you create a threat model and forget to connect data flows to elements, you get a notification. You can ignore the message, or you can follow the instructions to fix the issue.



Notes

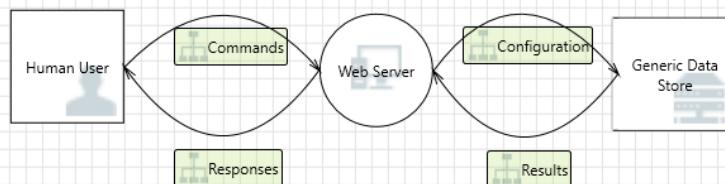
To add notes to your diagram, switch from the **Messages** tab to the **Notes** tab.

Analysis view

After you build your diagram, select the **Analysis** symbol (the magnifying glass) on the shortcuts toolbar to switch to the **Analysis** view.



Diagram 1 X



Threat List

ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
1	Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Commands	High
2	Diagram 1		Generated	Not Started	Elevation Usi...	Elevation Of...	Web Server...		Commands	High
3	Diagram 1		Generated	Not Started	Spoofing of D...	Spoofing	Generic Data...		Configuration	High
4	Diagram 1		Generated	Not Started	Potential Exc...	Denial Of Ser...	Does Web Se...		Configuration	High
5	Diagram 1		Generated	Not Started	Spoofing of S...	Spoofing	Generic Data...		Results	High
6	Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Results	High
7	Diagram 1		Generated	Not Started	Persistent Cr...	Tampering	The web serv...		Results	High
8	Diagram 1		Generated	Not Started	Weak Access...	Information...	Improper dat...		Results	High
9	Diagram 1		Generated	Not Started	Spoofing the...	Spoofing	Human User...		Commands	High

9 Threats Displayed, 9 Total

Threat Properties

No threats are selected

Generated threat selection

When you select a threat, you can use three distinct functions:

FEATURE

Read indicator

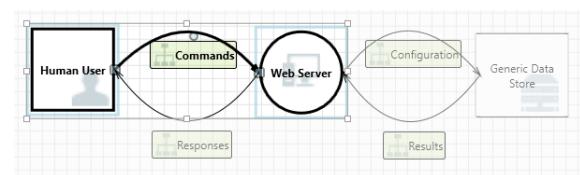
INFORMATION

The threat is marked as read, which helps you keep track of the items you reviewed.

ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
1	Diagram 1		Generated	Not Started	Cross Site Scr...	Tampering	The web serv...		Commands	High
2	Diagram 1		Generated	Not Started	Elevation Usi...	Elevation Of...	Web Server...		Commands	High
3	Diagram 1		Generated	Not Started	Spoofing of D...	Spoofing	Generic Data...		Configuration	High

Interaction focus

Interaction in the diagram that belongs to a threat is highlighted.



FEATURE	INFORMATION
Threat properties	Additional information about the threat appears in the Threat Properties window. 

Priority change

You can change the priority level of each generated threat. Different colors make it easy to identify high-, medium-, and low-priority threats.

ID	Diagram	Changed By	Last Modified	Status	Title	Category	Description	Justification	Interaction	Priority
1	Diagram 1	REDMOND\ro...	8/16/2017 3:13...	Not Started	Cross Site Scri...	Tampering	The web server...		Commands	Medium
2	Diagram 1	REDMOND\ro...	8/16/2017 3:13...	Not Started	Elevation Usin...	Elevation Of Pr...	Web Server ma...		Commands	Low
3	Diagram 1	REDMOND\ro...	8/16/2017 3:13...	Not Started	Spoofing of De...	Spoofing	Generic Data S...		Configuration	Medium
4	Diagram 1	REDMOND\ro...	8/16/2017 3:13...	Not Started	Potential Exces...	Denial Of Servi...	Does Web Serv...		Configuration	Medium
5	Diagram 1		Generated	Not Started	Spoofing of So...	Spoofing	Generic Data S...		Results	High
6	Diagram 1		Generated	Not Started	Cross Site Scri...	Tampering	The web server...		Results	High
7	Diagram 1	REDMOND\ro...	8/16/2017 3:13...	Not Started	Persistent Cros...	Tampering	The web server...		Results	Low
8	Diagram 1		Generated	Not Started	Weak Access...	Information...	Improper dat...		Results	High
9	Diagram 1		Generated	Not Started	Spoofing the...	Spoofing	Human User...		Commands	High

Threat properties editable fields

As seen in the preceding image, you can change the information generated by the tool. You can also add information to certain fields, such as justification. These fields are generated by the template. If you need more information for each threat, you can make modifications.

Threat Properties	
ID: 2	Diagram: Diagram 1
Status: Not Started	
Last Modified: Generated	
Title: Elevation Using Impersonation	
Category: Elevation Of Privilege	
Description: Web Server may be able to impersonate the context of Human User in order to gain additional privilege.	
Justification:	
Interaction: Commands	
Priority: High	

Reports

After you finish changing priorities and updating the status of each generated threat, you can save the file and/or print out a report. Go to **Report > Create Full Report**. Name the report, and you should see something similar to the following image:

Threat Modeling Report

Created on 7/31/2017 12:35:42 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0

Diagram: Diagram 1

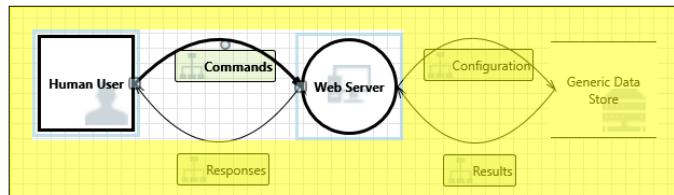
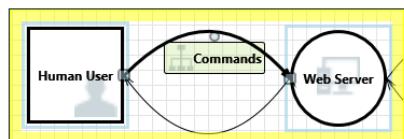


Diagram 1 Diagram Summary:

Not Started	9
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	9
Total Migrated	0

Interaction: Commands



1. Spoofing the Human User External Entity [State: Not Started] [Priority: High]

Category: Spoofing
Description: Human User may be spoofed by an attacker and this may lead to unauthorized access to Web Server. Consider using a standard authentication mechanism to identify the external entity.
Justification: <no mitigation provided>
Possible Mitigation(s):
SDL Phase: Design

2. Cross Site Scripting [State: Not Started] [Priority: High]

Category: Tampering
Description: The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification: <no mitigation provided>
Possible Mitigation(s):
SDL Phase: Design

Next steps

- To contribute a template for the community, go to our [GitHub](#) page.
- To get started with the tool, go to the [Download](#) page.

Microsoft Threat Modeling Tool threats

8/25/2017 • 2 minutes to read • [Edit Online](#)

The Threat Modeling Tool is a core element of the Microsoft Security Development Lifecycle (SDL). It allows software architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve. As a result, it greatly reduces the total cost of development. Also, we designed the tool with non-security experts in mind, making threat modeling easier for all developers by providing clear guidance on creating and analyzing threat models.

Visit the [Threat Modeling Tool](#) to get started today!

The Threat Modeling Tool helps you answer certain questions, such as the ones below:

- How can an attacker change the authentication data?
- What is the impact if an attacker can read the user profile data?
- What happens if access is denied to the user profile database?

STRIDE model

To better help you formulate these kinds of pointed questions, Microsoft uses the STRIDE model, which categorizes different types of threats and simplifies the overall security conversations.

CATEGORY	DESCRIPTION
Spoofing	Involves illegally accessing and then using another user's authentication information, such as username and password
Tampering	Involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet
Repudiation	Associated with users who deny performing an action without other parties having any way to prove otherwise—for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. Non-Repudiation refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package
Information Disclosure	Involves the exposure of information to individuals who are not supposed to have access to it—for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers
Denial of Service	Denial of service (DoS) attacks deny service to valid users—for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability

CATEGORY	DESCRIPTION
Elevation of Privilege	An unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed

Next steps

Proceed to [Threat Modeling Tool Mitigations](#) to learn the different ways you can mitigate these threats with Azure.

Microsoft Threat Modeling Tool mitigations

8/25/2017 • 2 minutes to read • [Edit Online](#)

The Threat Modeling Tool is a core element of the Microsoft Security Development Lifecycle (SDL). It allows software architects to identify and mitigate potential security issues early, when they are relatively easy and cost-effective to resolve. As a result, it greatly reduces the total cost of development. Also, we designed the tool with non-security experts in mind, making threat modeling easier for all developers by providing clear guidance on creating and analyzing threat models.

Visit the [Threat Modeling Tool](#) to get started today!

Mitigation categories

The Threat Modeling Tool mitigations are categorized according to the Web Application Security Frame, which consists of the following:

CATEGORY	DESCRIPTION
Auditing and Logging	Who did what and when? Auditing and logging refer to how your application records security-related events
Authentication	Who are you? Authentication is the process where an entity proves the identity of another entity, typically through credentials, such as a user name and password
Authorization	What can you do? Authorization is how your application provides access controls for resources and operations
Communication Security	Who are you talking to? Communication Security ensures all communication done is as secure as possible
Configuration Management	Who does your application run as? Which databases does it connect to? How is your application administered? How are these settings secured? Configuration management refers to how your application handles these operational issues
Cryptography	How are you keeping secrets (confidentiality)? How are you tamper-proofing your data or libraries (integrity)? How are you providing seeds for random values that must be cryptographically strong? Cryptography refers to how your application enforces confidentiality and integrity
Exception Management	When a method call in your application fails, what does your application do? How much do you reveal? Do you return friendly error information to end users? Do you pass valuable exception information back to the caller? Does your application fail gracefully?

CATEGORY	DESCRIPTION
Input Validation	How do you know that the input your application receives is valid and safe? Input validation refers to how your application filters, scrubs, or rejects input before additional processing. Consider constraining input through entry points and encoding output through exit points. Do you trust data from sources such as databases and file shares?
Sensitive Data	How does your application handle sensitive data? Sensitive data refers to how your application handles any data that must be protected either in memory, over the network, or in persistent stores
Session Management	How does your application handle and protect user sessions? A session refers to a series of related interactions between a user and your Web application

This helps you identify:

- Where are the most common mistakes made
- Where are the most actionable improvements

As a result, you use these categories to focus and prioritize your security work, so that if you know the most prevalent security issues occur in the input validation, authentication and authorization categories, you can start there. For more information visit [this patent link](#)

Next steps

Visit [Threat Modeling Tool Threats](#) to learn more about the threat categories the tool uses to generate possible design threats.

Security Frame: Auditing and Logging | Mitigations

8/30/2018 • 8 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Dynamics CRM	<ul style="list-style-type: none">• Identify sensitive entities in your solution and implement change auditing
Web Application	<ul style="list-style-type: none">• Ensure that auditing and logging is enforced on the application• Ensure that log rotation and separation are in place• Ensure that the application does not log sensitive user data• Ensure that Audit and Log Files have Restricted Access• Ensure that User Management Events are Logged• Ensure that the system has inbuilt defenses against misuse• Enable diagnostics logging for web apps in Azure App Service
Database	<ul style="list-style-type: none">• Ensure that login auditing is enabled on SQL Server• Enable Threat detection on Azure SQL
Azure Storage	<ul style="list-style-type: none">• Use Azure Storage Analytics to audit access of Azure Storage
WCF	<ul style="list-style-type: none">• Implement sufficient Logging• Implement sufficient Audit Failure Handling
Web API	<ul style="list-style-type: none">• Ensure that auditing and logging is enforced on Web API
IoT Field Gateway	<ul style="list-style-type: none">• Ensure that appropriate auditing and logging is enforced on Field Gateway
IoT Cloud Gateway	<ul style="list-style-type: none">• Ensure that appropriate auditing and logging is enforced on Cloud Gateway

Identify sensitive entities in your solution and implement change auditing

TITLE	DETAILS
Component	Dynamics CRM

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Identify entities in your solution containing sensitive data and implement change auditing on those entities and fields

Ensure that auditing and logging is enforced on the application

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Enable auditing and logging on all components. Audit logs should capture user context. Identify all important events and log those events. Implement centralized logging

Ensure that log rotation and separation are in place

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>Log rotation is an automated process used in system administration in which dated log files are archived. Servers which run large applications often log every request: in the face of bulky logs, log rotation is a way to limit the total size of the logs while still allowing analysis of recent events.</p> <p>Log separation basically means that you have to store your log files on a different partition as where your OS/application is running on in order to avert a Denial of service attack or the downgrading of your application its performance</p>

Ensure that the application does not log sensitive user data

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>Check that you do not log any sensitive data that a user submits to your site. Check for intentional logging as well as side effects caused by design issues. Examples of sensitive data include:</p> <ul style="list-style-type: none"> • User Credentials • Social Security number or other identifying information • Credit card numbers or other financial information • Health information • Private keys or other data that could be used to decrypt encrypted information • System or application information that can be used to more effectively attack the application

Ensure that Audit and Log Files have Restricted Access

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	N/A
Steps	<p>Check to ensure access rights to log files are appropriately set. Application accounts should have write-only access and operators and support personnel should have read-only access as needed.</p> <p>Administrators accounts are the only accounts which should have full access. Check Windows ACL on log files to ensure they are properly restricted:</p> <ul style="list-style-type: none"> • Application accounts should have write-only access • Operators and support personnel should have read-only access as needed • Administrators are the only accounts that should have full access

Ensure that User Management Events are Logged

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that the application monitors user management events such as successful and failed user logins, password resets, password changes, account lockout, user registration. Doing this helps to detect and react to potentially suspicious behavior. It also enables to gather operations data; for example, to track who is accessing the application

Ensure that the system has inbuilt defenses against misuse

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	N/A
Steps	<p>Controls should be in place which throw security exception in case of application misuse. E.g., If input validation is in place and an attacker attempts to inject malicious code that does not match the regex, a security exception can be thrown which can be an indicative of system misuse</p> <p>For example, it is recommended to have security exceptions logged and actions taken for the following issues:</p> <ul style="list-style-type: none"> • Input validation • CSRF violations • Brute force (upper limit for number of requests per user per resource) • File upload violations

Enable diagnostics logging for web apps in Azure App Service

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	EnvironmentType - Azure
References	N/A
Steps	<p>Azure provides built-in diagnostics to assist with debugging an App Service web app. It also applies to API apps and mobile apps. App Service web apps provide diagnostic functionality for logging information from both the web server and the web application.</p> <p>These are logically separated into web server diagnostics and application diagnostics</p>

Ensure that login auditing is enabled on SQL Server

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic

TITLE	DETAILS
Attributes	N/A
References	Configure Login Auditing
Steps	Database Server login auditing must be enabled to detect/confirm password guessing attacks. It is important to capture failed login attempts. Capturing both successful and failed login attempts provides additional benefit during forensic investigations

Enable Threat detection on Azure SQL

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	SQL Azure
Attributes	SQL Version - V12
References	Get Started with SQL Database Threat Detection
Steps	<p>Threat Detection detects anomalous database activities indicating potential security threats to the database. It provides a new layer of security, which enables customers to detect and respond to potential threats as they occur by providing security alerts on anomalous activities.</p> <p>Users can explore the suspicious events using Azure SQL Database Auditing to determine if they result from an attempt to access, breach or exploit data in the database.</p> <p>Threat Detection makes it simple to address potential threats to the database without the need to be a security expert or manage advanced security monitoring systems</p>

Use Azure Storage Analytics to audit access of Azure Storage

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	Using Storage Analytics to monitor authorization type

TITLE	DETAILS
Steps	<p>For each storage account, one can enable Azure Storage Analytics to perform logging and store metrics data. The storage analytics logs provide important information such as authentication method used by someone when they access storage.</p> <p>This can be really helpful if you are tightly guarding access to storage. For example, in Blob Storage you can set all of the containers to private and implement the use of an SAS service throughout your applications. Then you can check the logs regularly to see if your blobs are accessed using the storage account keys, which may indicate a breach of security, or if the blobs are public but they shouldn't be.</p>

Implement sufficient Logging

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework
Attributes	N/A
References	MSDN , Fortify Kingdom
Steps	<p>The lack of a proper audit trail after a security incident can hamper forensic efforts. Windows Communication Foundation (WCF) offers the ability to log successful and/or failed authentication attempts.</p> <p>Logging failed authentication attempts can warn administrators of potential brute-force attacks. Similarly, logging successful authentication events can provide a useful audit trail when a legitimate account is compromised. Enable WCF's service security audit feature</p>

Example

The following is an example configuration with auditing enabled

```
<system.serviceModel>
    <behaviors>
        <serviceBehaviors>
            <behavior name="NewBehavior">
                <serviceSecurityAudit auditLogLocation="Default"
                    suppressAuditFailure="false"
                    serviceAuthorizationAuditLevel="SuccessAndFailure"
                    messageAuthenticationAuditLevel="SuccessAndFailure" />
                ...
            </behavior>
        </serviceBehaviors>
    </behaviors>
</system.serviceModel>
```

Implement sufficient Audit Failure Handling

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework
Attributes	N/A
References	MSDN , Fortify Kingdom
Steps	Developed solution is configured not to generate an exception when it fails to write to an audit log. If WCF is configured not to throw an exception when it is unable to write to an audit log, the program will not be notified of the failure and auditing of critical security events may not occur.

Example

The `<behavior/>` element of the WCF configuration file below instructs WCF to not notify the application when WCF fails to write to an audit log.

```
<behaviors>
    <serviceBehaviors>
        <behavior name="NewBehavior">
            <serviceSecurityAudit auditLogLocation="Application"
                suppressAuditFailure="true"
                serviceAuthorizationAuditLevel="Success"
                messageAuthenticationAuditLevel="Success" />
        </behavior>
    </serviceBehaviors>
</behaviors>
```

Configure WCF to notify the program whenever it is unable to write to an audit log. The program should have an alternative notification scheme in place to alert the organization that audit trails are not being maintained.

Ensure that auditing and logging is enforced on Web API

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	Enable auditing and logging on Web APIs. Audit logs should capture user context. Identify all important events and log those events. Implement centralized logging

Ensure that appropriate auditing and logging is enforced on Field Gateway

TITLE	DETAILS
Component	IoT Field Gateway
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>When multiple devices connect to a Field Gateway, ensure that connection attempts and authentication status (success or failure) for individual devices are logged and maintained on the Field Gateway.</p> <p>Also, in cases where Field Gateway is maintaining the IoT Hub credentials for individual devices, ensure that auditing is performed when these credentials are retrieved. Develop a process to periodically upload the logs to Azure IoT Hub/storage for long term retention.</p>

Ensure that appropriate auditing and logging is enforced on Cloud Gateway

TITLE	DETAILS
Component	IoT Cloud Gateway
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Introduction to IoT Hub operations monitoring

TITLE	DETAILS
Steps	<p>Design for collecting and storing audit data gathered through IoT Hub Operations Monitoring. Enable the following monitoring categories:</p> <ul style="list-style-type: none">• Device identity operations• Device-to-cloud communications• Cloud-to-device communications• Connections• File uploads

Security Frame: Authentication | Mitigations

12/6/2018 • 25 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Web Application	<ul style="list-style-type: none">• Consider using a standard authentication mechanism to authenticate to Web Application• Applications must handle failed authentication scenarios securely• Enable step up or adaptive authentication• Ensure that administrative interfaces are appropriately locked down• Implement forgot password functionalities securely• Ensure that password and account policy are implemented• Implement controls to prevent username enumeration
Database	<ul style="list-style-type: none">• When possible, use Windows Authentication for connecting to SQL Server• When possible use Azure Active Directory Authentication for Connecting to SQL Database• When SQL authentication mode is used, ensure that account and password policy are enforced on SQL server• Do not use SQL Authentication in contained databases
Azure Event Hub	<ul style="list-style-type: none">• Use per device authentication credentials using SaaS tokens
Azure Trust Boundary	<ul style="list-style-type: none">• Enable Azure Multi-Factor Authentication for Azure Administrators
Service Fabric Trust Boundary	<ul style="list-style-type: none">• Restrict anonymous access to Service Fabric Cluster• Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate• Use AAD to authenticate clients to service fabric clusters• Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA)
Identity Server	<ul style="list-style-type: none">• Use standard authentication scenarios supported by Identity Server• Override the default Identity Server token cache with a scalable alternative
Machine Trust Boundary	<ul style="list-style-type: none">• Ensure that deployed application's binaries are digitally signed

PRODUCT/SERVICE	ARTICLE
WCF	<ul style="list-style-type: none"> Enable authentication when connecting to MSMQ queues in WCF WCF-Do not set Message clientCredentialType to none WCF-Do not set Transport clientCredentialType to none
Web API	<ul style="list-style-type: none"> Ensure that standard authentication techniques are used to secure Web APIs
Azure AD	<ul style="list-style-type: none"> Use standard authentication scenarios supported by Azure Active Directory Override the default ADAL token cache with a scalable alternative Ensure that TokenReplayCache is used to prevent the replay of ADAL authentication tokens Use ADAL libraries to manage token requests from OAuth2 clients to AAD (or on-premises AD)
IoT Field Gateway	<ul style="list-style-type: none"> Authenticate devices connecting to the Field Gateway
IoT Cloud Gateway	<ul style="list-style-type: none"> Ensure that devices connecting to Cloud gateway are authenticated Use per-device authentication credentials
Azure Storage	<ul style="list-style-type: none"> Ensure that only the required containers and blobs are given anonymous read access Grant limited access to objects in Azure storage using SAS or SAP

Consider using a standard authentication mechanism to authenticate to Web Application

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Details	<p>Authentication is the process where an entity proves its identity, typically through credentials, such as a user name and password. There are multiple authentication protocols available which may be considered. Some of them are listed below:</p> <ul style="list-style-type: none"> • Client certificates • Windows based • Forms based • Federation - ADFS • Federation - Azure AD • Federation - Identity Server <p>Consider using a standard authentication mechanism to identify the source process</p>

Applications must handle failed authentication scenarios securely

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Details	<p>Applications that explicitly authenticate users must handle failed authentication scenarios securely. The authentication mechanism must:</p> <ul style="list-style-type: none"> • Deny access to privileged resources when authentication fails • Display a generic error message after failed authentication and access denied occurs <p>Test for:</p> <ul style="list-style-type: none"> • Protection of privileged resources after failed logins • A generic error message is displayed on failed authentication and access denied event(s) • Accounts are disabled after an excessive number of failed attempts

Enable step up or adaptive authentication

TITLE	DETAILS
Component	Web Application
SDL Phase	Build

TITLE	DETAILS
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Details	<p>Verify the application has additional authorization (such as step up or adaptive authentication, via multi-factor authentication such as sending OTP in SMS, email etc. or prompting for re-authentication) so the user is challenged before being granted access to sensitive information. This rule also applies for making critical changes to an account or action</p> <p>This also means that the adaptation of authentication has to be implemented in such a manner that the application correctly enforces context-sensitive authorization so as to not allow unauthorized manipulation by means of for example, parameter tampering</p>

Ensure that administrative interfaces are appropriately locked down

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Details	The first solution is to grant access only from a certain source IP range to the administrative interface. If that solution would not be possible than it is always recommended to enforce a step-up or adaptive authentication for logging in into the administrative interface

Implement forgot password functionalities securely

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	N/A
Details	<p>The first thing is to verify that forgot password and other recovery paths send a link including a time-limited activation token rather than the password itself. Additional authentication based on soft-tokens (e.g. SMS token, native mobile applications, etc.) can be required as well before the link is sent over. Second, you should not lock out the users account whilst the process of getting a new password is in progress.</p> <p>This could lead to a Denial of service attack whenever an attacker decides to intentionally lock out the users with an automated attack. Third, whenever the new password request was set in progress, the message you display should be generalized in order to prevent username enumeration. Fourth, always disallow the use of old passwords and implement a strong password policy.</p>

Ensure that password and account policy are implemented

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Details	<p>Password and account policy in compliance with organizational policy and best practices should be implemented.</p> <p>To defend against brute-force and dictionary based guessing: Strong password policy must be implemented to ensure that users create complex password (e.g., 12 characters minimum length, alphanumeric and special characters).</p> <p>Account lockout policies may be implemented in the following manner:</p> <ul style="list-style-type: none"> • Soft lock-out: This can be a good option for protecting your users against brute force attacks. For example, whenever the user enters a wrong password three times the application could lock down the account for a minute in order to slow down the process of brute forcing his password making it less profitable for the attacker to proceed. If you were to implement hard lock-out countermeasures for this example you would achieve a "Dos" by permanently locking out accounts. Alternatively, application may generate an OTP (One Time Password) and send it out-of-band (through email, sms etc.) to the user. Another approach may be to implement CAPTCHA after a threshold number of failed attempts is reached. • Hard lock-out: This type of lockout should be applied whenever you detect a user attacking your application and counter him by means of permanently locking out his account until a response team had time to do their forensics. After this process you can decide to give the user back his account or take further legal actions against him. This type of approach prevents the attacker from further penetrating your application and infrastructure. <p>To defend against attacks on default and predictable accounts, verify that all keys and passwords are replaceable, and are generated or replaced after installation time.</p> <p>If the application has to auto-generate passwords, ensure that the generated passwords are random and have high entropy.</p>

Implement controls to prevent username enumeration

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	N/A
Steps	All error messages should be generalized in order to prevent username enumeration. Also sometimes you cannot avoid information leaking in functionalities such as a registration page. Here you need to use rate-limiting methods like CAPTCHA to prevent an automated attack by an attacker.

When possible, use Windows Authentication for connecting to SQL Server

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	OnPrem
Attributes	SQL Version - All
References	SQL Server - Choose an Authentication Mode
Steps	Windows Authentication uses Kerberos security protocol, provides password policy enforcement with regard to complexity validation for strong passwords, provides support for account lockout, and supports password expiration.

When possible use Azure Active Directory Authentication for Connecting to SQL Database

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	SQL Azure
Attributes	SQL Version - V12
References	Connecting to SQL Database By Using Azure Active Directory Authentication
Steps	Minimum version: Azure SQL Database V12 required to allow Azure SQL Database to use AAD Authentication against the Microsoft Directory

When SQL authentication mode is used, ensure that account and

password policy are enforced on SQL server

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	SQL Server password policy
Steps	When using SQL Server Authentication, logins are created in SQL Server that are not based on Windows user accounts. Both the user name and the password are created by using SQL Server and stored in SQL Server. SQL Server can use Windows password policy mechanisms. It can apply the same complexity and expiration policies used in Windows to passwords used inside SQL Server.

Do not use SQL Authentication in contained databases

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	OnPrem, SQL Azure
Attributes	SQL Version - MSSQL2012, SQL Version - V12
References	Security Best Practices with Contained Databases
Steps	The absence of an enforced password policy may increase the likelihood of a weak credential being established in a contained database. Leverage Windows Authentication.

Use per device authentication credentials using SaaS tokens

TITLE	DETAILS
Component	Azure Event Hub
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	Event Hubs authentication and security model overview
Steps	<p>The Event Hubs security model is based on a combination of Shared Access Signature (SAS) tokens and event publishers. The publisher name represents the DeviceID that receives the token. This would help associate the tokens generated with the respective devices.</p> <p>All messages are tagged with originator on service side allowing detection of in-payload origin spoofing attempts. When authenticating devices, generate a per device SaS token scoped to a unique publisher.</p>

Enable Azure Multi-Factor Authentication for Azure Administrators

TITLE	DETAILS
Component	Azure Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	What is Azure Multi-Factor Authentication?
Steps	<p>Multi-factor authentication (MFA) is a method of authentication that requires more than one verification method and adds a critical second layer of security to user sign-ins and transactions. It works by requiring any two or more of the following verification methods:</p> <ul style="list-style-type: none"> • Something you know (typically a password) • Something you have (a trusted device that is not easily duplicated, like a phone) • Something you are (biometrics)

Restrict anonymous access to Service Fabric Cluster

TITLE	DETAILS
Component	Service Fabric Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	Environment - Azure
References	Service Fabric cluster security scenarios

TITLE	DETAILS
Steps	<p>Clusters should always be secured to prevent unauthorized users from connecting to your cluster, especially when it has production workloads running on it.</p> <p>While creating a service fabric cluster, ensure that the security mode is set to "secure" and configure the required X.509 server certificate. Creating an "insecure" cluster will allow any anonymous user to connect to it if it exposes management endpoints to the public Internet.</p>

Ensure that Service Fabric client-to-node certificate is different from node-to-node certificate

TITLE	DETAILS
Component	Service Fabric Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	Environment - Azure, Environment - Stand alone
References	Service Fabric Client-to-node certificate security , Connect to a secure cluster using client certificate
Steps	<p>Client-to-node certificate security is configured while creating the cluster either through the Azure portal, Resource Manager templates or a standalone JSON template by specifying an admin client certificate and/or a user client certificate.</p> <p>The admin client and user client certificates you specify should be different than the primary and secondary certificates you specify for Node-to-node security.</p>

Use AAD to authenticate clients to service fabric clusters

TITLE	DETAILS
Component	Service Fabric Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	Environment - Azure
References	Cluster security scenarios - Security Recommendations

TITLE	DETAILS
Steps	Clusters running on Azure can also secure access to the management endpoints using Azure Active Directory (AAD), apart from client certificates. For Azure clusters, it is recommended that you use AAD security to authenticate clients and certificates for node-to-node security.

Ensure that service fabric certificates are obtained from an approved Certificate Authority (CA)

TITLE	DETAILS
Component	Service Fabric Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	Environment - Azure
References	X.509 certificates and Service Fabric
Steps	<p>Service Fabric uses X.509 server certificates for authenticating nodes and clients.</p> <p>Some important things to consider while using certificates in service fabrics:</p> <ul style="list-style-type: none"> • Certificates used in clusters running production workloads should be created by using a correctly configured Windows Server certificate service or obtained from an approved Certificate Authority (CA). The CA can be an approved external CA or a properly managed internal Public Key Infrastructure (PKI) • Never use any temporary or test certificates in production that are created with tools such as MakeCert.exe • You can use a self-signed certificate, but should only do so for test clusters and not in production

Use standard authentication scenarios supported by Identity Server

TITLE	DETAILS
Component	Identity Server
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	IdentityServer3 - The Big Picture

TITLE	DETAILS
Steps	<p>Below are the typical interactions supported by Identity Server:</p> <ul style="list-style-type: none"> • Browsers communicate with web applications • Web applications communicate with web APIs (sometimes on their own, sometimes on behalf of a user) • Browser-based applications communicate with web APIs • Native applications communicate with web APIs • Server-based applications communicate with web APIs • Web APIs communicate with web APIs (sometimes on their own, sometimes on behalf of a user)

Override the default Identity Server token cache with a scalable alternative

TITLE	DETAILS
Component	Identity Server
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	Identity Server Deployment - Caching
Steps	<p>IdentityServer has a simple built-in in-memory cache. While this is good for small scale native apps, it does not scale for mid tier and backend applications for the following reasons:</p> <ul style="list-style-type: none"> • These applications are accessed by many users at once. Saving all access tokens in the same store creates isolation issues and presents challenges when operating at scale: many users, each with as many tokens as the resources the app accesses on their behalf, can mean huge numbers and very expensive lookup operations • These applications are typically deployed on distributed topologies, where multiple nodes must have access to the same cache • Cached tokens must survive process recycles and deactivations • For all the above reasons, while implementing web apps, it is recommended to override the default Identity Server's token cache with a scalable alternative such as Azure Cache for Redis

Ensure that deployed application's binaries are digitally signed

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that deployed application's binaries are digitally signed so that the integrity of the binaries can be verified

Enable authentication when connecting to MSMQ queues in WCF

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	N/A
References	MSDN
Steps	Program fails to enable authentication when connecting to MSMQ queues, an attacker can anonymously submit messages to the queue for processing. If authentication is not used to connect to an MSMQ queue used to deliver a message to another program, an attacker could submit an anonymous message that is malicious.

Example

The `<netMsmqBinding/>` element of the WCF configuration file below instructs WCF to disable authentication when connecting to an MSMQ queue for message delivery.

```

<bindings>
    <netMsmqBinding>
        <binding>
            <security>
                <transport msmqAuthenticationMode=""None"" />
            </security>
        </binding>
    </netMsmqBinding>
</bindings>

```

Configure MSMQ to require Windows Domain or Certificate authentication at all times for any incoming or outgoing messages.

Example

The `<netMsmqBinding/>` element of the WCF configuration file below instructs WCF to enable certificate authentication when connecting to an MSMQ queue. The client is authenticated using X.509 certificates. The client certificate must be present in the certificate store of the server.

```

<bindings>
  <netMsmqBinding>
    <binding>
      <security>
        <transport msmqAuthenticationMode=""Certificate"" />
      </security>
    </binding>
  </netMsmqBinding>
</bindings>

```

WCF-Do not set Message clientCredentialType to none

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework 3
Attributes	Client Credential Type - None
References	MSDN , Fortify
Steps	The absence of authentication means everyone is able to access this service. A service that does not authenticate its clients allows access to all users. Configure the application to authenticate against client credentials. This can be done by setting the message clientCredentialType to Windows or Certificate.

Example

```
<message clientCredentialType=""Certificate""/>
```

WCF-Do not set Transport clientCredentialType to none

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	Client Credential Type - None
References	MSDN , Fortify

TITLE	DETAILS
Steps	The absence of authentication means everyone is able to access this service. A service that does not authenticate its clients allows all users to access its functionality. Configure the application to authenticate against client credentials. This can be done by setting the transport clientCredentialType to Windows or Certificate.

Example

```
<transport clientCredentialType="Certificate"/>
```

Ensure that standard authentication techniques are used to secure Web APIs

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Authentication and Authorization in ASP.NET Web API , External Authentication Services with ASP.NET Web API (C#)
Steps	<p>Authentication is the process where an entity proves its identity, typically through credentials, such as a user name and password. There are multiple authentication protocols available which may be considered. Some of them are listed below:</p> <ul style="list-style-type: none"> • Client certificates • Windows based • Forms based • Federation - ADFS • Federation - Azure AD • Federation - Identity Server <p>Links in the references section provide low-level details on how each of the authentication schemes can be implemented to secure a Web API.</p>

Use standard authentication scenarios supported by Azure Active Directory

TITLE	DETAILS
Component	Azure AD

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Authentication Scenarios for Azure AD, Azure Active Directory Code Samples, Azure Active Directory developer's guide
Steps	<p>Azure Active Directory (Azure AD) simplifies authentication for developers by providing identity as a service, with support for industry-standard protocols such as OAuth 2.0 and OpenID Connect. Below are the five primary application scenarios supported by Azure AD:</p> <ul style="list-style-type: none"> • Web Browser to Web Application: A user needs to sign in to a web application that is secured by Azure AD • Single Page Application (SPA): A user needs to sign in to a single page application that is secured by Azure AD • Native Application to Web API: A native application that runs on a phone, tablet, or PC needs to authenticate a user to get resources from a web API that is secured by Azure AD • Web Application to Web API: A web application needs to get resources from a web API secured by Azure AD • Daemon or Server Application to Web API: A daemon application or a server application with no web user interface needs to get resources from a web API secured by Azure AD <p>Please refer to the links in the references section for low-level implementation details</p>

Override the default ADAL token cache with a scalable alternative

TITLE	DETAILS
Component	Azure AD
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Modern Authentication with Azure Active Directory for Web Applications, Using Redis as ADAL token cache

TITLE	DETAILS
Steps	<p>The default cache that ADAL (Active Directory Authentication Library) uses is an in-memory cache that relies on a static store, available process-wide. While this works for native applications, it does not scale for mid tier and backend applications for the following reasons:</p> <ul style="list-style-type: none"> • These applications are accessed by many users at once. Saving all access tokens in the same store creates isolation issues and presents challenges when operating at scale: many users, each with as many tokens as the resources the app accesses on their behalf, can mean huge numbers and very expensive lookup operations • These applications are typically deployed on distributed topologies, where multiple nodes must have access to the same cache • Cached tokens must survive process recycles and deactivations <p>For all the above reasons, while implementing web apps, it is recommended to override the default ADAL token cache with a scalable alternative such as Azure Cache for Redis.</p>

Ensure that TokenReplayCache is used to prevent the replay of ADAL authentication tokens

TITLE	DETAILS
Component	Azure AD
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Modern Authentication with Azure Active Directory for Web Applications

TITLE	DETAILS
Steps	<p>The TokenReplayCache property allows developers to define a token replay cache, a store that can be used for saving tokens for the purpose of verifying that no token can be used more than once.</p> <p>This is a measure against a common attack, the aptly called token replay attack: an attacker intercepting the token sent at sign-in might try to send it to the app again ("replay" it) for establishing a new session. E.g., In OIDC code-grant flow, after successful user authentication, a request to "/signin-oidc" endpoint of the relying party is made with "id_token", "code" and "state" parameters.</p> <p>The relying party validates this request and establishes a new session. If an adversary captures this request and replays it, he/she can establish a successful session and spoof the user. The presence of the nonce in OpenID Connect can limit but not fully eliminate the circumstances in which the attack can be successfully enacted. To protect their applications, developers can provide an implementation of ITokenReplayCache and assign an instance to TokenReplayCache.</p>

Example

```
// ITokenReplayCache defined in ADAL
public interface ITokenReplayCache
{
    bool TryAdd(string securityToken, DateTime expiresOn);
    bool TryFind(string securityToken);
}
```

Example

Here is a sample implementation of the ITokenReplayCache interface. (Please customize and implement your project-specific caching framework)

```
public class TokenReplayCache : ITokenReplayCache
{
    private readonly ICacheProvider cache; // Your project-specific cache provider
    public TokenReplayCache(ICacheProvider cache)
    {
        this.cache = cache;
    }
    public bool TryAdd(string securityToken, DateTime expiresOn)
    {
        if (this.cache.Get<string>(securityToken) == null)
        {
            this.cache.Set(securityToken, securityToken);
            return true;
        }
        return false;
    }
    public bool TryFind(string securityToken)
    {
        return this.cache.Get<string>(securityToken) != null;
    }
}
```

The implemented cache has to be referenced in OIDC options via the "TokenValidationParameters" property as

follows.

```
OpenIdConnectOptions openIdConnectOptions = new OpenIdConnectOptions
{
    AutomaticAuthenticate = true,
    ... // other configuration properties follow..
    TokenValidationParameters = new TokenValidationParameters
    {
        TokenReplayCache = new TokenReplayCache(/*Inject your cache provider*/);
    }
}
```

Please note that to test the effectiveness of this configuration, login into your local OIDC-protected application and capture the request to `"/signin-oidc"` endpoint in fiddler. When the protection is not in place, replaying this request in fiddler will set a new session cookie. When the request is replayed after the `TokenReplayCache` protection is added, the application will throw an exception as follows:

```
SecurityTokenReplayDetectedException: IDX10228: The securityToken has previously been validated, securityToken: 'eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ik1uQ19WVmNBVGZNNXBPWWIKSE1iYTlnb0VLWSIsImtpZCI6Ik1uQ1.....
```

Use ADAL libraries to manage token requests from OAuth2 clients to AAD (or on-premises AD)

TITLE	DETAILS
Component	Azure AD
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	ADAL
Steps	<p>The Azure AD authentication Library (ADAL) enables client application developers to easily authenticate users to cloud or on-premises Active Directory (AD), and then obtain access tokens for securing API calls.</p> <p>ADAL has many features that make authentication easier for developers, such as asynchronous support, a configurable token cache that stores access tokens and refresh tokens, automatic token refresh when an access token expires and a refresh token is available, and more.</p> <p>By handling most of the complexity, ADAL can help a developer focus on business logic in their application and easily secure resources without being an expert on security. Separate libraries are available for .NET, JavaScript (client and Node.js), iOS, Android and Java.</p>

Authenticate devices connecting to the Field Gateway

TITLE	DETAILS
Component	IoT Field Gateway

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that each device is authenticated by the Field Gateway before accepting data from them and before facilitating upstream communications with the Cloud Gateway. Also, ensure that devices connect with a per device credential so that individual devices can be uniquely identified.

Ensure that devices connecting to Cloud gateway are authenticated

TITLE	DETAILS
Component	IoT Cloud Gateway
SDL Phase	Build
Applicable Technologies	Generic, C#, Node.JS,
Attributes	N/A, Gateway choice - Azure IoT Hub
References	N/A, Azure IoT hub with .NET , Getting Started with IoT hub and Node JS , Securing IoT with SAS and certificates , Git repository
Steps	<ul style="list-style-type: none"> • Generic: Authenticate the device using Transport Layer Security (TLS) or IPSec. Infrastructure should support using pre-shared key (PSK) on those devices that cannot handle full asymmetric cryptography. Leverage Azure AD, OAuth. • C#: When creating a DeviceClient instance, by default, the Create method creates a DeviceClient instance that uses the AMQP protocol to communicate with IoT Hub. To use the HTTPS protocol, use the override of the Create method that enables you to specify the protocol. If you use the HTTPS protocol, you should also add the <code>Microsoft.AspNetCore.WebApi.Client</code> NuGet package to your project to include the <code>System.Net.Http.Formatting</code> namespace.

Example

```

static DeviceClient deviceClient;

static string deviceKey = "{device key}";
static string iotHubUri = "{iot hub hostname}";

var messageString = "{message in string format}";
var message = new Message(Encoding.ASCII.GetBytes(messageString));

deviceClient = DeviceClient.Create(iotHubUri, new
DeviceAuthenticationWithRegistrySymmetricKey("myFirstDevice", deviceKey));

await deviceClient.SendEventAsync(message);

```

Example

Node.JS: Authentication

Symmetric key

- Create a IoT hub on azure
- Create an entry in the device identity registry

```
javascript var device = new iothub.Device(null); device.deviceId = <DeviceId> registry.create(device,
function(err, deviceInfo, res) {})
```

- Create a simulated device

```
javascript var clientFromConnectionString = require('azure-iot-device-amqp').clientFromConnectionString; var
Message = require('azure-iot-device').Message; var connectionString = 'HostName=<HostName>DeviceId=
<DeviceId>SharedAccessKey=<SharedAccessKey>'; var client = clientFromConnectionString(connectionString);
```

SAS Token

- Gets internally generated when using symmetric key but we can generate and use it explicitly as well
- Define a protocol : `var Http = require('azure-iot-device-http').Http;`
- Create a sas token : ````javascript resourceUri =
encodeURIComponent(resourceUri.toLowerCase()).toLowerCase(); var deviceName = ""; var expires =
(Date.now() / 1000) + expiresInMins * 60; var toSign = resourceUri + '\n' + expires; // using crypto var
decodedPassword = new Buffer(signingKey, 'base64').toString('binary'); const hmac =
crypto.createHmac('sha256', decodedPassword); hmac.update(toSign); var base64signature =
hmac.digest('base64'); var base64UriEncoded = encodeURIComponent(base64signature); // construct
authorization string var token = "SharedAccessSignature sr=" + resourceUri +
"%2fdevices%2f" + deviceName + "&sig="
- `base64UriEncoded + "&se=" + expires; if (policyName) token += "&skn=" + policyName; return token;` ````

- Connect using sas token: `javascript Client.fromSharedAccessSignature(sas, Http);` #### Certificates
- Generate a self signed X509 certificate using any tool such as OpenSSL to generate a .cert and .key files to
store the certificate and the key respectively
- Provision a device that accepts secured connection using certificates.

```
javascript var connectionString = '<connectionString>'; var registry =
iothub.Registry.fromConnectionString(connectionString); var deviceJSON = {deviceId:<deviceId>,
authentication: { x509Thumbprint: { primaryThumbprint: "<PrimaryThumbprint>", secondaryThumbprint: "
<SecondaryThumbprint>" } }} var device = deviceJSON; registry.create(device, function (err) {});
```

- Connect a device using a certificate

```
javascript var Protocol = require('azure-iot-device-http').Http; var Client = require('azure-iot-
device').Client; var connectionString = 'HostName=<HostName>DeviceId=<DeviceId>x509=true'; var client =
Client.fromConnectionString(connectionString, Protocol); var options = { key: fs.readFileSync('./key.pem',
'utf8'), cert: fs.readFileSync('./server.crt', 'utf8') }; // Calling setOptions with the x509 certificate
and key (and optionally, passphrase) will configure the client //transport to use x509 when connecting to
IoT Hub client.setOptions(options); //call fn to execute after the connection is set up client.open(fn);
```

Use per-device authentication credentials

TITLE	DETAILS
Component	IoT Cloud Gateway
SDL Phase	Build
Applicable Technologies	Generic
Attributes	Gateway choice - Azure IoT Hub
References	Azure IoT Hub Security Tokens
Steps	Use per device authentication credentials using SAs tokens based on Device key or Client Certificate, instead of IoT Hub-level shared access policies. This prevents the reuse of authentication tokens of one device or field gateway by another

Ensure that only the required containers and blobs are given anonymous read access

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	StorageType - Blob
References	Manage anonymous read access to containers and blobs, Shared Access Signatures, Part 1: Understanding the SAS model

TITLE	DETAILS
Steps	<p>By default, a container and any blobs within it may be accessed only by the owner of the storage account. To give anonymous users read permissions to a container and its blobs, one can set the container permissions to allow public access. Anonymous users can read blobs within a publicly accessible container without authenticating the request.</p> <p>Containers provide the following options for managing container access:</p> <ul style="list-style-type: none"> • Full public read access: Container and blob data can be read via anonymous request. Clients can enumerate blobs within the container via anonymous request, but cannot enumerate containers within the storage account. • Public read access for blobs only: Blob data within this container can be read via anonymous request, but container data is not available. Clients cannot enumerate blobs within the container via anonymous request • No public read access: Container and blob data can be read by the account owner only <p>Anonymous access is best for scenarios where certain blobs should always be available for anonymous read access. For finer-grained control, one can create a shared access signature, which enables to delegate restricted access using different permissions and over a specified time interval. Ensure that containers and blobs, which may potentially contain sensitive data, are not given anonymous access accidentally</p>

Grant limited access to objects in Azure storage using SAS or SAP

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Shared Access Signatures, Part 1: Understanding the SAS model , Shared Access Signatures, Part 2: Create and use a SAS with Blob storage , How to delegate access to objects in your account using Shared Access Signatures and Stored Access Policies

TITLE	DETAILS
Steps	<p>Using a shared access signature (SAS) is a powerful way to grant limited access to objects in a storage account to other clients, without having to expose account access key. The SAS is a URI that encompasses in its query parameters all of the information necessary for authenticated access to a storage resource. To access storage resources with the SAS, the client only needs to pass in the SAS to the appropriate constructor or method.</p> <p>You can use a SAS when you want to provide access to resources in your storage account to a client that can't be trusted with the account key. Your storage account keys include both a primary and secondary key, both of which grant administrative access to your account and all of the resources in it. Exposing either of your account keys opens your account to the possibility of malicious or negligent use. Shared access signatures provide a safe alternative that allows other clients to read, write, and delete data in your storage account according to the permissions you've granted, and without need for the account key.</p> <p>If you have a logical set of parameters that are similar each time, using a Stored Access Policy (SAP) is a better idea. Because using a SAS derived from a Stored Access Policy gives you the ability to revoke that SAS immediately, it is the recommended best practice to always use Stored Access Policies when possible.</p>

Security Frame: Authorization | Mitigations

12/6/2018 • 17 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Machine Trust Boundary	<ul style="list-style-type: none">• Ensure that proper ACLs are configured to restrict unauthorized access to data on the device• Ensure that sensitive user-specific application content is stored in user-profile directory• Ensure that the deployed applications are run with least privileges
Web Application	<ul style="list-style-type: none">• Enforce sequential step order when processing business logic flows• Implement rate limiting mechanism to prevent enumeration• Ensure that proper authorization is in place and principle of least privileges is followed• Business logic and resource access authorization decisions should not be based on incoming request parameters• Ensure that content and resources are not enumerable or accessible via forceful browsing
Database	<ul style="list-style-type: none">• Ensure that least-privileged accounts are used to connect to Database server• Implement Row Level Security RLS to prevent tenants from accessing each other's data• Sysadmin role should only have valid necessary users
IoT Cloud Gateway	<ul style="list-style-type: none">• Connect to Cloud Gateway using least-privileged tokens
Azure Event Hub	<ul style="list-style-type: none">• Use a send-only permissions SAS Key for generating device tokens• Do not use access tokens that provide direct access to the Event Hub• Connect to Event Hub using SAS keys that have the minimum permissions required
Azure Document DB	<ul style="list-style-type: none">• Use resource tokens to connect to Azure Cosmos DB whenever possible
Azure Trust Boundary	<ul style="list-style-type: none">• Enable fine-grained access management to Azure Subscription using RBAC
Service Fabric Trust Boundary	<ul style="list-style-type: none">• Restrict client's access to cluster operations using RBAC

PRODUCT/SERVICE	ARTICLE
Dynamics CRM	<ul style="list-style-type: none"> • Perform security modeling and use Field Level Security where required
Dynamics CRM Portal	<ul style="list-style-type: none"> • Perform security modeling of portal accounts keeping in mind that the security model for the portal differs from the rest of CRM
Azure Storage	<ul style="list-style-type: none"> • Grant fine-grained permission on a range of entities in Azure Table Storage • Enable Role-Based Access Control (RBAC) to Azure storage account using Azure Resource Manager
Mobile Client	<ul style="list-style-type: none"> • Implement implicit jailbreak or rooting detection
WCF	<ul style="list-style-type: none"> • Weak Class Reference in WCF • WCF-Implement Authorization control
Web API	<ul style="list-style-type: none"> • Implement proper authorization mechanism in ASP.NET Web API
IoT Device	<ul style="list-style-type: none"> • Perform authorization checks in the device if it supports various actions that require different permission levels
IoT Field Gateway	<ul style="list-style-type: none"> • Perform authorization checks in the Field Gateway if it supports various actions that require different permission levels

Ensure that proper ACLs are configured to restrict unauthorized access to data on the device

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that proper ACLs are configured to restrict unauthorized access to data on the device

Ensure that sensitive user-specific application content is stored in user-profile directory

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that sensitive user-specific application content is stored in user-profile directory. This is to prevent multiple users of the machine from accessing each other's data.

Ensure that the deployed applications are run with least privileges

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that the deployed application is run with least privileges.

Enforce sequential step order when processing business logic flows

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	In order to verify that this stage was run through by a genuine user you want to enforce the application to only process business logic flows in sequential step order, with all steps being processed in realistic human time, and not process out of order, skipped steps, processed steps from another user, or too quickly submitted transactions.

Implement rate limiting mechanism to prevent enumeration

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that sensitive identifiers are random. Implement CAPTCHA control on anonymous pages. Ensure that error and exception should not reveal specific data

Ensure that proper authorization is in place and principle of least privileges is followed

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>The principle means giving a user account only those privileges which are essential to that user's work. For example, a backup user does not need to install software; hence, the backup user has rights only to run backup and backup-related applications. Any other privileges, such as installing new software, are blocked. The principle applies also to a personal computer user who usually does work in a normal user account, and opens a privileged, password protected account (that is, a superuser) only when the situation absolutely demands it.</p> <p>This principle can also be applied to your web-applications. Instead of solely depending on role-based authentication methods using sessions, we rather want to assign privileges to users by means of a Database-Based Authentication system. We still use sessions in order to identify if the user was logged in correctly, only now instead of assigning that user with a specific role we assign him with privileges to verify which actions he is privileged to perform on the system. Also a big pro of this method is, whenever a user has to be assigned fewer privileges your changes will be applied on the fly since the assigning does not depend on the session which otherwise had to expire first.</p>

Business logic and resource access authorization decisions should not be based on incoming request parameters

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Whenever you are checking whether a user is restricted to review certain data, the access restrictions should be processed server-side. The userID should be stored inside of a session variable on login and should be used to retrieve user data from the database

Example

```
SELECT data
FROM personaldata
WHERE userID=:id < - session var
```

Now an possible attacker can not tamper and change the application operation since the identifier for retrieving the data is handled server-side.

Ensure that content and resources are not enumerable or accessible via forceful browsing

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>Sensitive static and configuration files should not be kept in the web-root. For content not required to be public, either proper access controls should be applied or removal of the content itself.</p> <p>Also, forceful browsing is usually combined with Brute Force techniques to gather information by attempting to access as many URLs as possible to enumerate directories and files on a server. Attackers may check for all variations of commonly existing files. For example, a password file search would encompass files including psswd.txt, password.htm, password.dat, and other variations.</p> <p>To mitigate this, capabilities for detection of brute force attempts should be included.</p>

Ensure that least-privileged accounts are used to connect to Database server

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	SQL Database permissions hierarchy , SQL database securables
Steps	Least-privileged accounts should be used to connect to the database. Application login should be restricted in the database and should only execute selected stored procedures. Application's login should have no direct table access.

Implement Row Level Security RLS to prevent tenants from accessing each other's data

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Sql Azure, OnPrem
Attributes	SQL Version - V12, SQL Version - MsSQL2016
References	SQL Server Row-Level Security (RLS)
Steps	<p>Row-Level Security enables customers to control access to rows in a database table based on the characteristics of the user executing a query (e.g., group membership or execution context).</p> <p>Row-Level Security (RLS) simplifies the design and coding of security in your application. RLS enables you to implement restrictions on data row access. For example ensuring that workers can access only those data rows that are pertinent to their department, or restricting a customer's data access to only the data relevant to their company.</p> <p>The access restriction logic is located in the database tier rather than away from the data in another application tier. The database system applies the access restrictions every time that data access is attempted from any tier. This makes the security system more reliable and robust by reducing the surface area of the security system.</p>

Please note that RLS as an out-of-the-box database feature is applicable only to SQL Server starting 2016 and Azure SQL database. If the out-of-the-box RLS feature is not implemented, it should be ensured that data access is restricted Using Views and Procedures

Sysadmin role should only have valid necessary users

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	SQL Database permissions hierarchy, SQL database securables
Steps	Members of the SysAdmin fixed server role should be very limited and never contain accounts used by applications. Please review the list of users in the role and remove any unnecessary accounts

Connect to Cloud Gateway using least-privileged tokens

TITLE	DETAILS
Component	IoT Cloud Gateway
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	Gateway choice - Azure IoT Hub
References	IoT Hub Access Control
Steps	Provide least privilege permissions to various components that connect to Cloud Gateway (IoT Hub). Typical example is – Device management/provisioning component uses registryread/write, Event Processor (ASA) uses Service Connect. Individual devices connect using Device credentials

Use a send-only permissions SAS Key for generating device tokens

TITLE	DETAILS
Component	Azure Event Hub
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Event Hubs authentication and security model overview
Steps	A SAS key is used to generate individual device tokens. Use a send-only permissions SAS key while generating the device token for a given publisher

Do not use access tokens that provide direct access to the Event Hub

TITLE	DETAILS
Component	Azure Event Hub
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Event Hubs authentication and security model overview

TITLE	DETAILS
Steps	A token that grants direct access to the event hub should not be given to the device. Using a least privileged token for the device that gives access only to a publisher would help identify and blacklist it if found to be a rogue or compromised device.

Connect to Event Hub using SAS keys that have the minimum permissions required

TITLE	DETAILS
Component	Azure Event Hub
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Event Hubs authentication and security model overview
Steps	Provide least privilege permissions to various back-end applications that connect to the Event Hub. Generate separate SAS keys for each back-end application and only provide the required permissions - Send, Receive or Manage to them.

Use resource tokens to connect to Cosmos DB whenever possible

TITLE	DETAILS
Component	Azure Document DB
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	A resource token is associated with an Azure Cosmos DB permission resource and captures the relationship between the user of a database and the permission that user has for a specific Azure Cosmos DB application resource (e.g. collection, document). Always use a resource token to access the Azure Cosmos DB if the client cannot be trusted with handling master or read-only keys - like an end user application like a mobile or desktop client. Use Master key or read-only keys from backend applications which can store these keys securely.

Enable fine-grained access management to Azure Subscription using

RBAC

TITLE	DETAILS
Component	Azure Trust Boundary
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Use role assignments to manage access to your Azure subscription resources
Steps	Azure Role-Based Access Control (RBAC) enables fine-grained access management for Azure. Using RBAC, you can grant only the amount of access that users need to perform their jobs.

Restrict client's access to cluster operations using RBAC

TITLE	DETAILS
Component	Service Fabric Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	Environment - Azure
References	Role-based access control for Service Fabric clients
Steps	<p>Azure Service Fabric supports two different access control types for clients that are connected to a Service Fabric cluster: administrator and user. Access control allows the cluster administrator to limit access to certain cluster operations for different groups of users, making the cluster more secure.</p> <p>Administrators have full access to management capabilities (including read/write capabilities). Users, by default, have only read access to management capabilities (for example, query capabilities), and the ability to resolve applications and services.</p> <p>You specify the two client roles (administrator and client) at the time of cluster creation by providing separate certificates for each.</p>

Perform security modeling and use Field Level Security where required

TITLE	DETAILS
Component	Dynamics CRM
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Perform security modeling and use Field Level Security where required

Perform security modeling of portal accounts keeping in mind that the security model for the portal differs from the rest of CRM

TITLE	DETAILS
Component	Dynamics CRM Portal
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Perform security modeling of portal accounts keeping in mind that the security model for the portal differs from the rest of CRM

Grant fine-grained permission on a range of entities in Azure Table Storage

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	StorageType - Table
References	How to delegate access to objects in your Azure storage account using SAS

TITLE	DETAILS
Steps	In certain business scenarios, Azure Table Storage may be required to store sensitive data that caters to different parties. E.g., sensitive data pertaining to different countries. In such cases, SAS signatures can be constructed by specifying the partition and row key ranges, such that a user can access data specific to a particular country.

Enable Role-Based Access Control (RBAC) to Azure storage account using Azure Resource Manager

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	How to secure your storage account with Role-Based Access Control (RBAC)
Steps	<p>When you create a new storage account, you select a deployment model of Classic or Azure Resource Manager. The Classic model of creating resources in Azure only allows all-or-nothing access to the subscription, and in turn, the storage account.</p> <p>With the Azure Resource Manager model, you put the storage account in a resource group and control access to the management plane of that specific storage account using Azure Active Directory. For example, you can give specific users the ability to access the storage account keys, while other users can view information about the storage account, but cannot access the storage account keys.</p>

Implement implicit jailbreak or rooting detection

TITLE	DETAILS
Component	Mobile Client
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>Application should safeguard its own configuration and user data in case if phone is rooted or jail broken. Rooting/jail breaking implies unauthorized access, which normal users won't do on their own phones. Therefore application should have implicit detection logic on application startup, to detect if the phone has been rooted.</p> <p>The detection logic can be simply accessing files which normally only root user can access, for example:</p> <ul style="list-style-type: none"> • /system/app/Superuser.apk • /sbin/su • /system/bin/su • /system/xbin/su • /data/local/xbin/su • /data/local/bin/su • /system/sd/xbin/su • /system/bin/failsafe/su • /data/local/su <p>If the application can access any of these files, it denotes that the application is running as root user.</p>

Weak Class Reference in WCF

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	N/A
References	MSDN , Fortify Kingdom

TITLE	DETAILS
Steps	<p>The system uses a weak class reference, which might allow an attacker to execute unauthorized code. The program references a user-defined class that is not uniquely identified. When .NET loads this weakly identified class, the CLR type loader searches for the class in the following locations in the specified order:</p> <ol style="list-style-type: none"> 1. If the assembly of the type is known, the loader searches the configuration file's redirect locations, GAC, the current assembly using configuration information, and the application base directory 2. If the assembly is unknown, the loader searches the current assembly, mscorelib, and the location returned by the TypeResolve event handler 3. This CLR search order can be modified with hooks such as the Type Forwarding mechanism and the AppDomain.TypeResolve event <p>If an attacker exploits the CLR search order by creating an alternative class with the same name and placing it in an alternative location that the CLR will load first, the CLR will unintentionally execute the attacker-supplied code</p>

Example

The `<behaviorExtensions/>` element of the WCF configuration file below instructs WCF to add a custom behavior class to a particular WCF extension.

```

<system.serviceModel>
    <extensions>
        <behaviorExtensions>
            <add name=""myBehavior"" type=""MyBehavior"" />
        </behaviorExtensions>
    </extensions>
</system.serviceModel>

```

Using fully qualified (strong) names uniquely identifies a type and further increases security of your system. Use fully qualified assembly names when registering types in the machine.config and app.config files.

Example

The `<behaviorExtensions/>` element of the WCF configuration file below instructs WCF to add strongly-referenced custom behavior class to a particular WCF extension.

```

<system.serviceModel>
    <extensions>
        <behaviorExtensions>
            <add name=""myBehavior"" type=""Microsoft.ServiceModel.Samples.MyBehaviorSection, MyBehavior,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=null"" />
        </behaviorExtensions>
    </extensions>
</system.serviceModel>

```

WCF-Implement Authorization control

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	N/A
References	MSDN , Fortify Kingdom
Steps	This service does not use an authorization control. When a client calls a particular WCF service, WCF provides various authorization schemes that verify that the caller has permission to execute the service method on the server. If authorization controls are not enabled for WCF services, an authenticated user can achieve privilege escalation.

Example

The following configuration instructs WCF to not check the authorization level of the client when executing the service:

```
<behaviors>
  <serviceBehaviors>
    <behavior>
      ...
      <serviceAuthorization principalPermissionMode=""None"" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

Use a service authorization scheme to verify that the caller of the service method is authorized to do so. WCF provides two modes and allows the definition of a custom authorization scheme. The UseWindowsGroups mode uses Windows roles and users and the UseAspNetRoles mode uses an ASP.NET role provider, such as SQL Server, to authenticate.

Example

The following configuration instructs WCF to make sure that the client is part of the Administrators group before executing the Add service:

```
<behaviors>
  <serviceBehaviors>
    <behavior>
      ...
      <serviceAuthorization principalPermissionMode=""UseWindowsGroups"" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

The service is then declared as the following:

```
[PrincipalPermission(SecurityAction.Demand,
Role = "Builtin\Administrators")]
public double Add(double n1, double n2)
{
    double result = n1 + n2;
    return result;
}
```

Implement proper authorization mechanism in ASP.NET Web API

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	Generic, MVC5
Attributes	N/A, Identity Provider - ADFS, Identity Provider - Azure AD
References	Authentication and Authorization in ASP.NET Web API
Steps	<p>Role information for the application users can be derived from Azure AD or ADFS claims if the application relies on them as Identity provider or the application itself might provided it. In any of these cases, the custom authorization implementation should validate the user role information.</p> <p>Role information for the application users can be derived from Azure AD or ADFS claims if the application relies on them as Identity provider or the application itself might provided it. In any of these cases, the custom authorization implementation should validate the user role information.</p>

Example

```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, Inherited = true, AllowMultiple = true)]
public class ApiAuthorizeAttribute : System.Web.Http.AuthorizeAttribute
{
    public async override Task OnAuthorizationAsync(HttpContext actionContext, CancellationToken cancellationToken)
    {
        if (actionContext == null)
        {
            throw new Exception();
        }

        if (!string.IsNullOrEmpty(base.Roles))
        {
            bool isAuthorized = ValidateRoles(actionContext);
            if (!isAuthorized)
            {
                HandleUnauthorizedRequest(actionContext);
            }
        }

        base.OnAuthorization(actionContext);
    }

    public bool ValidateRoles(HttpContext)
    {
        //Authorization logic here; returns true or false
    }
}
```

All the controllers and action methods which needs to protected should be decorated with above attribute.

```
[ApiAuthorize]
public class CustomController : ApiController
{
    //Application code goes here
}
```

Perform authorization checks in the device if it supports various actions that require different permission levels

TITLE	DETAILS
Component	IoT Device
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>The Device should authorize the caller to check if the caller has the required permissions to perform the action requested. For e.g. Lets say the device is a Smart Door Lock that can be monitored from the cloud, plus it provides functionalities like Remotely locking the door.</p> <p>The Smart Door Lock provides unlocking functionality only when someone physically comes near the door with a Card. In this case, the implementation of the remote command and control should be done in such a way that it does not provide any functionality to unlock the door as the cloud gateway is not authorized to send a command to unlock the door.</p>

Perform authorization checks in the Field Gateway if it supports various actions that require different permission levels

TITLE	DETAILS
Component	IoT Field Gateway
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>The Field Gateway should authorize the caller to check if the caller has the required permissions to perform the action requested. For e.g. there should be different permissions for an admin user interface/API used to configure a field gateway v/s devices that connect to it.</p>

Security Frame: Communication Security | Mitigations

12/6/2018 • 14 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Azure Event Hub	<ul style="list-style-type: none">Secure communication to Event Hub using SSL/TLS
Dynamics CRM	<ul style="list-style-type: none">Check service account privileges and check that the custom Services or ASP.NET Pages respect CRM's security
Azure Data Factory	<ul style="list-style-type: none">Use Data management gateway while connecting On Prem SQL Server to Azure Data Factory
Identity Server	<ul style="list-style-type: none">Ensure that all traffic to Identity Server is over HTTPS connection
Web Application	<ul style="list-style-type: none">Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connectionsConfigure SSL certificate for custom domain in Azure App ServiceForce all traffic to Azure App Service over HTTPS connectionEnable HTTP Strict Transport Security (HSTS)
Database	<ul style="list-style-type: none">Ensure SQL server connection encryption and certificate validationForce Encrypted communication to SQL server
Azure Storage	<ul style="list-style-type: none">Ensure that communication to Azure Storage is over HTTPSValidate MD5 hash after downloading blob if HTTPS cannot be enabledUse SMB 3.0 compatible client to ensure in-transit data encryption to Azure File Shares
Mobile Client	<ul style="list-style-type: none">Implement Certificate Pinning
WCF	<ul style="list-style-type: none">Enable HTTPS - Secure Transport channelWCF: Set Message security Protection level to EncryptAndSignWCF: Use a least-privileged account to run your WCF service
Web API	<ul style="list-style-type: none">Force all traffic to Web APIs over HTTPS connection

PRODUCT/SERVICE	ARTICLE
Azure Cache for Redis	<ul style="list-style-type: none"> • Ensure that communication to Azure Cache for Redis is over SSL
IoT Field Gateway	<ul style="list-style-type: none"> • Secure Device to Field Gateway communication
IoT Cloud Gateway	<ul style="list-style-type: none"> • Secure Device to Cloud Gateway communication using SSL/TLS

Secure communication to Event Hub using SSL/TLS

TITLE	DETAILS
Component	Azure Event Hub
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Event Hubs authentication and security model overview
Steps	Secure AMQP or HTTP connections to Event Hub using SSL/TLS

Check service account privileges and check that the custom Services or ASP.NET Pages respect CRM's security

TITLE	DETAILS
Component	Dynamics CRM
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Check service account privileges and check that the custom Services or ASP.NET Pages respect CRM's security

Use Data management gateway while connecting On Prem SQL Server to Azure Data Factory

TITLE	DETAILS
Component	Azure Data Factory
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	Linked Service Types - Azure and On Prem
References	Moving data between On Prem and Azure Data Factory, Data management gateway
Steps	<p>The Data Management Gateway (DMG) tool is required to connect to data sources which are protected behind corpnet or a firewall.</p> <ol style="list-style-type: none"> 1. Locking down the machine isolates the DMG tool and prevents malfunctioning programs from damaging or snooping on the data source machine. (E.g. latest updates must be installed, enable minimum required ports, controlled accounts provisioning, auditing enabled, disk encryption enabled etc.) 2. Data Gateway key must be rotated at frequent intervals or whenever the DMG service account password renews 3. Data transits through Link Service must be encrypted

Ensure that all traffic to Identity Server is over HTTPS connection

TITLE	DETAILS
Component	Identity Server
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	IdentityServer3 - Keys, Signatures and Cryptography , IdentityServer3 - Deployment
Steps	<p>By default, IdentityServer requires all incoming connections to come over HTTPS. It is absolutely mandatory that communication with IdentityServer is done over secured transports only. There are certain deployment scenarios like SSL offloading where this requirement can be relaxed. See the Identity Server deployment page in the references for more information.</p>

Verify X.509 certificates used to authenticate SSL, TLS, and DTLS connections

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>Applications that use SSL, TLS, or DTLS must fully verify the X.509 certificates of the entities they connect to. This includes verification of the certificates for:</p> <ul style="list-style-type: none"> • Domain name • Validity dates (both beginning and expiration dates) • Revocation status • Usage (for example, Server Authentication for servers, Client Authentication for clients) • Trust chain. Certificates must chain to a root certification authority (CA) that is trusted by the platform or explicitly configured by the administrator • Key length of certificate's public key must be >2048 bits • Hashing algorithm must be SHA256 and above

Configure SSL certificate for custom domain in Azure App Service

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	EnvironmentType - Azure
References	Enable HTTPS for an app in Azure App Service
Steps	<p>By default, Azure already enables HTTPS for every app with a wildcard certificate for the *.azurewebsites.net domain. However, like all wildcard domains, it is not as secure as using a custom domain with own certificate Refer. It is recommended to enable SSL for the custom domain which the deployed app will be accessed through</p>

Force all traffic to Azure App Service over HTTPS connection

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	EnvironmentType - Azure
References	Enforce HTTPS on Azure App Service
Steps	<p>Though Azure already enables HTTPS for Azure app services with a wildcard certificate for the domain *.azurewebsites.net, it does not enforce HTTPS. Visitors may still access the app using HTTP, which may compromise the app's security and hence HTTPS has to be enforced explicitly. ASP.NET MVC applications should use the RequireHttps filter that forces an unsecured HTTP request to be re-sent over HTTPS.</p> <p>Alternatively, the URL Rewrite module, which is included with Azure App Service can be used to enforce HTTPS. URL Rewrite module enables developers to define rules that are applied to incoming requests before the requests are handed to your application. URL Rewrite rules are defined in a web.config file stored in the root of the application</p>

Example

The following example contains a basic URL Rewrite rule that forces all incoming traffic to use HTTPS

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="Force HTTPS" enabled="true">
          <match url="(.*)" ignoreCase="false" />
          <conditions>
            <add input="{HTTPS}" pattern="off" />
          </conditions>
          <action type="Redirect" url="https:///{HTTP_HOST}/{R:1}" appendQueryString="true"
redirectType="Permanent" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
```

This rule works by returning an HTTP status code of 301 (permanent redirect) when the user requests a page using HTTP. The 301 redirect is to the same URL as the visitor requested, but replaces the HTTP portion of the request with HTTPS. For example, [HTTP://contoso.com](http://contoso.com) would be redirected to [HTTPS://contoso.com](https://contoso.com).

Enable HTTP Strict Transport Security (HSTS)

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	OWASP HTTP Strict Transport Security Cheat Sheet
Steps	<p>HTTP Strict Transport Security (HSTS) is an opt-in security enhancement that is specified by a web application through the use of a special response header. Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS. It also prevents HTTPS click through prompts on browsers.</p> <p>To implement HSTS, the following response header has to be configured for a website globally, either in code or in config. Strict-Transport-Security: max-age=300; includeSubDomains HSTS addresses the following threats:</p> <ul style="list-style-type: none"> • User bookmarks or manually types http://example.com and is subject to a man-in-the-middle attacker: HSTS automatically redirects HTTP requests to HTTPS for the target domain • Web application that is intended to be purely HTTPS inadvertently contains HTTP links or serves content over HTTP: HSTS automatically redirects HTTP requests to HTTPS for the target domain • A man-in-the-middle attacker attempts to intercept traffic from a victim user using an invalid certificate and hopes the user will accept the bad certificate: HSTS does not allow a user to override the invalid certificate message

Ensure SQL server connection encryption and certificate validation

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	SQL Azure
Attributes	SQL Version - V12
References	Best Practices on Writing Secure Connection Strings for SQL Database

TITLE	DETAILS
Steps	<p>All communications between SQL Database and a client application are encrypted using Secure Sockets Layer (SSL) at all times. SQL Database doesn't support unencrypted connections. To validate certificates with application code or tools, explicitly request an encrypted connection and do not trust the server certificates. If your application code or tools do not request an encrypted connection, they will still receive encrypted connections</p> <p>However, they may not validate the server certificates and thus will be susceptible to "man in the middle" attacks. To validate certificates with ADO.NET application code, set <code>Encrypt=True</code> and <code>TrustServerCertificate=False</code> in the database connection string. To validate certificates via SQL Server Management Studio, open the Connect to Server dialog box. Click Encrypt connection on the Connection Properties tab</p>

Force Encrypted communication to SQL server

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	OnPrem
Attributes	SQL Version - MsSQL2016, SQL Version - MsSQL2012, SQL Version - MsSQL2014
References	Enable Encrypted Connections to the Database Engine
Steps	Enabling SSL encryption increases the security of data transmitted across networks between instances of SQL Server and applications.

Ensure that communication to Azure Storage is over HTTPS

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	Azure Storage Transport-Level Encryption – Using HTTPS

TITLE	DETAILS
Steps	To ensure the security of Azure Storage data in-transit, always use the HTTPS protocol when calling the REST APIs or accessing objects in storage. Also, Shared Access Signatures, which can be used to delegate access to Azure Storage objects, include an option to specify that only the HTTPS protocol can be used when using Shared Access Signatures, ensuring that anybody sending out links with SAS tokens will use the proper protocol.

Validate MD5 hash after downloading blob if HTTPS cannot be enabled

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	StorageType - Blob
References	Windows Azure Blob MD5 Overview
Steps	<p>Windows Azure Blob service provides mechanisms to ensure data integrity both at the application and transport layers. If for any reason you need to use HTTP instead of HTTPS and you are working with block blobs, you can use MD5 checking to help verify the integrity of the blobs being transferred</p> <p>This will help with protection from network/transport layer errors, but not necessarily with intermediary attacks. If you can use HTTPS, which provides transport level security, then using MD5 checking is redundant and unnecessary.</p>

Use SMB 3.0 compatible client to ensure in-transit data encryption to Azure File shares

TITLE	DETAILS
Component	Mobile Client
SDL Phase	Build
Applicable Technologies	Generic
Attributes	StorageType - File
References	Azure File Storage, Azure File Storage SMB Support for Windows Clients

TITLE	DETAILS
Steps	Azure File Storage supports HTTPS when using the REST API, but is more commonly used as an SMB file share attached to a VM. SMB 2.1 does not support encryption, so connections are only allowed within the same region in Azure. However, SMB 3.0 supports encryption, and can be used with Windows Server 2012 R2, Windows 8, Windows 8.1, and Windows 10, allowing cross-region access and even access on the desktop.

Implement Certificate Pinning

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic, Windows Phone
Attributes	N/A
References	Certificate and Public Key Pinning
Steps	<p>Certificate pinning defends against Man-In-The-Middle (MITM) attacks. Pinning is the process of associating a host with their expected X509 certificate or public key. Once a certificate or public key is known or seen for a host, the certificate or public key is associated or 'pinned' to the host.</p> <p>Thus, when an adversary attempts to do SSL MITM attack, during SSL handshake the key from attacker's server will be different from the pinned certificate's key, and the request will be discarded, thus preventing MITM. Certificate pinning can be achieved by implementing ServicePointManager's <code>ServerCertificateValidationCallback</code> delegate.</p>

Example

```

using System;
using System.Net;
using System.Net.Security;
using System.Security.Cryptography;

namespace CertificatePinningExample
{
    class CertificatePinningExample
    {
        /* Note: In this example, we're hardcoding a the certificate's public key and algorithm for
           demonstration purposes. In a real-world application, this should be stored in a secure
           configuration area that can be updated as needed. */

        private static readonly string PINNED_ALGORITHM = "RSA";

        private static readonly string PINNED_PUBLIC_KEY =
            "3082010A0282010100B0E75B7CBE56D31658EF79B3A1" +
            "294D506A88DFCDD603F6EF15E7F5BCBDF32291EC50B2B82BA158E905FE6A83EE044A48258B07FAC3D6356AF09B2" +
            "3EDAB15D00507B70DB08DB9A20C7D1201417B3071A346D663A241061C151B6EC5B5B4ECCDCDBEA24F051962809" +
            "FEC499BF2D093C06E3BDA7D0BB83CDC1C2C6660B8ECB2EA30A685ADE2DC83C88314010FC7F4F0F895EDDBE5C02" +
            "ABF78E50B708E0A0EB984A9AA536BCE61A0C31DB95425C6FEE5A564B158EE7C4F0693C439AE010EF83CA8155750" +
            "09B17537C29F86071E5DD8CA50EBD8A409494F479B07574D83EDCE6F68A8F7D40447471D05BC3F5EAD7862FA748" +
            "EA3C92A60A128344B1CEF7A0B0D94E50203010001";

        public static void Main(string[] args)
        {
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create("https://azure.microsoft.com");
            request.ServerCertificateValidationCallback = (sender, certificate, chain, sslPolicyErrors) =>
            {
                if (certificate == null || sslPolicyErrors != SslPolicyErrors.None)
                {
                    // Error getting certificate or the certificate failed basic validation
                    return false;
                }

                var targetKeyAlgorithm = new Oid(certificate.GetKeyAlgorithm()).FriendlyName;
                var targetPublicKey = certificate.GetPublicKeyString();

                if (targetKeyAlgorithm == PINNED_ALGORITHM &&
                    targetPublicKey == PINNED_PUBLIC_KEY)
                {
                    // Success, the certificate matches the pinned value.
                    return true;
                }
                // Reject, either the key or the algorithm does not match the expected value.
                return false;
            };

            try
            {
                var response = (HttpWebResponse)request.GetResponse();
                Console.WriteLine($"Success, HTTP status code: {response.StatusCode}");
            }
            catch(Exception ex)
            {
                Console.WriteLine($"Failure, {ex.Message}");
            }
            Console.WriteLine("Press any key to end.");
            Console.ReadKey();
        }
    }
}

```

Enable HTTPS - Secure Transport channel

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework 3
Attributes	N/A
References	MSDN , Fortify Kingdom
Steps	<p>The application configuration should ensure that HTTPS is used for all access to sensitive information.</p> <ul style="list-style-type: none"> • EXPLANATION: If an application handles sensitive information and does not use message-level encryption, then it should only be allowed to communicate over an encrypted transport channel. • RECOMMENDATIONS: Ensure that HTTP transport is disabled and enable HTTPS transport instead. For example, replace the <code><httpTransport/></code> with <code><httpsTransport/></code> tag. Do not rely on a network configuration (firewall) to guarantee that the application can only be accessed over a secure channel. From a philosophical point of view, the application should not depend on the network for its security. <p>From a practical point of view, the people responsible for securing the network do not always track the security requirements of the application as they evolve.</p>

WCF: Set Message security Protection level to EncryptAndSign

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework 3
Attributes	N/A
References	MSDN

TITLE	DETAILS
Steps	<ul style="list-style-type: none"> EXPLANATION: When Protection level is set to "none" it will disable message protection. Confidentiality and integrity is achieved with appropriate level of setting. RECOMMENDATIONS: <ul style="list-style-type: none"> when <code>Mode=None</code> - Disables message protection when <code>Mode=Sign</code> - Signs but does not encrypt the message; should be used when data integrity is important when <code>Mode=EncryptAndSign</code> - Signs and encrypts the message <p>Consider turning off encryption and only signing your message when you just need to validate the integrity of the information without concerns of confidentiality. This may be useful for operations or service contracts in which you need to validate the original sender but no sensitive data is transmitted. When reducing the protection level, be careful that the message does not contain any personally identifiable information (PII).</p>

Example

Configuring the service and the operation to only sign the message is shown in the following examples. Service Contract Example of `ProtectionLevel.Sign`: The following is an example of using `ProtectionLevel.Sign` at the Service Contract level:

```
[ServiceContract(Protection Level=ProtectionLevel.Sign)]
public interface IService
{
    string GetData(int value);
}
```

Example

Operation Contract Example of `ProtectionLevel.Sign` (for Granular Control): The following is an example of using `ProtectionLevel.Sign` at the OperationContract level:

```
[OperationContract(ProtectionLevel=ProtectionLevel.Sign)]
string GetData(int value);
```

WCF: Use a least-privileged account to run your WCF service

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework 3
Attributes	N/A

TITLE	DETAILS
References	MSDN
Steps	<ul style="list-style-type: none"> EXPLANATION: Do not run WCF services under admin or high privilege account. in case of services compromise it will result in high impact. RECOMMENDATIONS: Use a least-privileged account to host your WCF service because it will reduce your application's attack surface and reduce the potential damage if you are attacked. If the service account requires additional access rights on infrastructure resources such as MSMQ, the event log, performance counters, and the file system, appropriate permissions should be given to these resources so that the WCF service can run successfully. <p>If your service needs to access specific resources on behalf of the original caller, use impersonation and delegation to flow the caller's identity for a downstream authorization check. In a development scenario, use the local network service account, which is a special built-in account that has reduced privileges. In a production scenario, create a least-privileged custom domain service account.</p>

Force all traffic to Web APIs over HTTPS connection

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	MVC5, MVC6
Attributes	N/A
References	Enforcing SSL in a Web API Controller
Steps	If an application has both an HTTPS and an HTTP binding, clients can still use HTTP to access the site. To prevent this, use an action filter to ensure that requests to protected APIs are always over HTTPS.

Example

The following code shows a Web API authentication filter that checks for SSL:

```

public class RequireHttpsAttribute : AuthorizationFilterAttribute
{
    public override void OnAuthorization(HttpActionContext actionContext)
    {
        if (actionContext.Request.RequestUri.Scheme != Uri.UriSchemeHttps)
        {
            actionContext.Response = new HttpResponseMessage(System.Net.HttpStatusCode.Forbidden)
            {
                ReasonPhrase = "HTTPS Required"
            };
        }
        else
        {
            base.OnAuthorization(actionContext);
        }
    }
}

```

Add this filter to any Web API actions that require SSL:

```

public class ValuesController : ApiController
{
    [RequireHttps]
    public HttpResponseMessage Get() { ... }
}

```

Ensure that communication to Azure Cache for Redis is over SSL

TITLE	DETAILS
Component	Azure Cache for Redis
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Azure Redis SSL support
Steps	Redis server does not support SSL out of the box, but Azure Cache for Redis does. If you are connecting to Azure Cache for Redis and your client supports SSL, like StackExchange.Redis, then you should use SSL. By default non-SSL port is disabled for new Azure Cache for Redis instances. Ensure that the secure defaults are not changed unless there is a dependency on SSL support for redis clients.

Please note that Redis is designed to be accessed by trusted clients inside trusted environments. This means that usually it is not a good idea to expose the Redis instance directly to the internet or, in general, to an environment where untrusted clients can directly access the Redis TCP port or UNIX socket.

Secure Device to Field Gateway communication

TITLE	DETAILS
Component	IoT Field Gateway
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	For IP based devices, the communication protocol could typically be encapsulated in a SSL/TLS channel to protect data in transit. For other protocols that do not support SSL/TLS investigate if there are secure versions of the protocol that provide security at transport or message layer.

Secure Device to Cloud Gateway communication using SSL/TLS

TITLE	DETAILS
Component	IoT Cloud Gateway
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Choose your Communication Protocol
Steps	Secure HTTP/AMQP or MQTT protocols using SSL/TLS.

Security Frame: Configuration Management | Mitigations

12/3/2018 • 20 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Web Application	<ul style="list-style-type: none">• Implement Content Security Policy (CSP), and disable inline javascript• Enable browser's XSS filter• ASP.NET applications must disable tracing and debugging prior to deployment• Access third-party javascripts from trusted sources only• Ensure that authenticated ASP.NET pages incorporate UI Redressing or click-jacking defenses• Ensure that only trusted origins are allowed if CORS is enabled on ASP.NET Web Applications• Enable ValidateRequest attribute on ASP.NET Pages• Use locally-hosted latest versions of JavaScript libraries• Disable automatic MIME sniffing• Remove standard server headers on Windows Azure Web Sites to avoid fingerprinting
Database	<ul style="list-style-type: none">• Configure a Windows Firewall for Database Engine Access
Web API	<ul style="list-style-type: none">• Ensure that only trusted origins are allowed if CORS is enabled on ASP.NET Web API• Encrypt sections of Web API's configuration files that contain sensitive data
IoT Device	<ul style="list-style-type: none">• Ensure that all admin interfaces are secured with strong credentials• Ensure that unknown code cannot execute on devices• Encrypt OS and additional partitions of IoT Device with bit-locker• Ensure that only the minimum services/features are enabled on devices
IoT Field Gateway	<ul style="list-style-type: none">• Encrypt OS and additional partitions of IoT Field Gateway with bit-locker• Ensure that the default login credentials of the field gateway are changed during installation
IoT Cloud Gateway	<ul style="list-style-type: none">• Ensure that the Cloud Gateway implements a process to keep the connected devices firmware up to date

PRODUCT/SERVICE	ARTICLE
Machine Trust Boundary	<ul style="list-style-type: none"> • Ensure that devices have end-point security controls configured as per organizational policies
Azure Storage	<ul style="list-style-type: none"> • Ensure secure management of Azure storage access keys • Ensure that only trusted origins are allowed if CORS is enabled on Azure storage
WCF	<ul style="list-style-type: none"> • Enable WCF's service throttling feature • WCF-Information disclosure through metadata

Implement Content Security Policy (CSP), and disable inline javascript

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	An Introduction to Content Security Policy, Content Security Policy Reference, Security features, Introduction to content security policy, Can I use CSP?

TITLE	DETAILS
Steps	<p>Content Security Policy (CSP) is a defense-in-depth security mechanism, a W3C standard, that enables web application owners to have control on the content embedded in their site. CSP is added as an HTTP response header on the web server and is enforced on the client side by browsers. It is a whitelist-based policy - a website can declare a set of trusted domains from which active content such as JavaScript can be loaded.</p> <p>CSP provides the following security benefits:</p> <ul style="list-style-type: none"> • Protection against XSS: If a page is vulnerable to XSS, an attacker can exploit it in 2 ways: <ul style="list-style-type: none"> ◦ Inject <code><script>malicious code</script></code>. This exploit will not work due to CSP's Base Restriction-1 ◦ Inject <code><script src="http://attacker.com/maliciousCode.js"/></code>. This exploit will not work since the attacker controlled domain will not be in CSP's whitelist of domains • Control over data exfiltration: If any malicious content on a webpage attempts to connect to an external website and steal data, the connection will be aborted by CSP. This is because the target domain will not be in CSP's whitelist • Defense against click-jacking: click-jacking is an attack technique using which an adversary can frame a genuine website and force users to click on UI elements. Currently defense against click-jacking is achieved by configuring a response header- X-Frame-Options. Not all browsers respect this header and going forward CSP will be a standard way to defend against click-jacking • Real-time attack reporting: If there is an injection attack on a CSP-enabled website, browsers will automatically trigger a notification to an endpoint configured on the webserver. This way, CSP serves as a real-time warning system.

Example

Example policy:

```
Content-Security-Policy: default-src 'self'; script-src 'self' www.google-analytics.com
```

This policy allows scripts to load only from the web application's server and google analytics server. Scripts loaded from any other site will be rejected. When CSP is enabled on a website, the following features are automatically disabled to mitigate XSS attacks.

Example

Inline scripts will not execute. Following are examples of inline scripts

```
<script> some Javascript code </script>
Event handling attributes of HTML tags (e.g., <button onclick="function(){}
javascript:alert(1);"
```

Example

Strings will not be evaluated as code.

```
Example: var str="alert(1)"; eval(str);
```

Enable browser's XSS filter

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	XSS Protection Filter
Steps	<p>X-XSS-Protection response header configuration controls the browser's cross site script filter. This response header can have following values:</p> <ul style="list-style-type: none">• <code>0</code>: This will disable the filter• <code>1: Filter enabled</code>: If a cross-site scripting attack is detected, in order to stop the attack, the browser will sanitize the page• <code>1: mode=block : Filter enabled</code>: Rather than sanitize the page, when a XSS attack is detected, the browser will prevent rendering of the page• <code>1: report=http://[YOURDOMAIN]/your_report_URI : Filter enabled</code>: The browser will sanitize the page and report the violation. <p>This is a Chromium function utilizing CSP violation reports to send details to a URI of your choice. The last 2 options are considered safe values.</p>

ASP.NET applications must disable tracing and debugging prior to deployment

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	ASP.NET Debugging Overview , ASP.NET Tracing Overview , How to: Enable Tracing for an ASP.NET Application , How to: Enable Debugging for ASP.NET Applications
Steps	When tracing is enabled for the page, every browser requesting it also obtains the trace information that contains data about internal server state and workflow. That information could be security sensitive. When debugging is enabled for the page, errors happening on the server result in a full stack trace data presented to the browser. That data may expose security-sensitive information about the server's workflow.

Access third-party javascripts from trusted sources only

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	third-party JavaScripts should be referenced only from trusted sources. The reference endpoints should always be on SSL.

Ensure that authenticated ASP.NET pages incorporate UI Redressing or click-jacking defenses

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	OWASP click-jacking Defense Cheat Sheet , IE Internals - Combating click-jacking With X-Frame-Options

TITLE	DETAILS
Steps	<p>click-jacking, also known as a "UI redress attack", is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top-level page.</p> <p>This layering is achieved by crafting a malicious page with an iframe, which loads the victim's page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both. To prevent click-jacking attacks, set the proper X-Frame-Options HTTP response headers that instruct the browser to not allow framing from other domains</p>

Example

The X-FRAME-OPTIONS header can be set via IIS web.config. Web.config code snippet for sites that should never be framed:

```
<system.webServer>
  <httpProtocol>
    <customHeader>
      <add name="X-FRAME-OPTIONS" value="DENY"/>
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

Example

Web.config code for sites that should only be framed by pages in the same domain:

```
<system.webServer>
  <httpProtocol>
    <customHeader>
      <add name="X-FRAME-OPTIONS" value="SAMEORIGIN"/>
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

Ensure that only trusted origins are allowed if CORS is enabled on ASP.NET Web Applications

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Web Forms, MVC5
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>Browser security prevents a web page from making AJAX requests to another domain. This restriction is called the same-origin policy, and prevents a malicious site from reading sensitive data from another site. However, sometimes it might be required to expose APIs securely which other sites can consume. Cross Origin Resource Sharing (CORS) is a W3C standard that allows a server to relax the same-origin policy. Using CORS, a server can explicitly allow some cross-origin requests while rejecting others.</p> <p>CORS is safer and more flexible than earlier techniques such as JSONP. At its core, enabling CORS translates to adding a few HTTP response headers (Access-Control-*) to the web application and this can be done in a couple of ways.</p>

Example

If access to Web.config is available, then CORS can be added through the following code:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <clear />
      <add name="Access-Control-Allow-Origin" value="http://example.com" />
    </customHeaders>
  </httpProtocol>
```

Example

If access to web.config is not available, then CORS can be configured by adding the following CSharp code:

```
HttpContext.Response.AppendHeader("Access-Control-Allow-Origin", "http://example.com")
```

Please note that it is critical to ensure that the list of origins in "Access-Control-Allow-Origin" attribute is set to a finite and trusted set of origins. Failing to configure this appropriately (e.g., setting the value as '*') will allow malicious sites to trigger cross origin requests to the web application >without any restrictions, thereby making the application vulnerable to CSRF attacks.

Enable ValidateRequest attribute on ASP.NET Pages

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Web Forms, MVC5
Attributes	N/A
References	Request Validation - Preventing Script Attacks

TITLE	DETAILS
Steps	<p>Request validation, a feature of ASP.NET since version 1.1, prevents the server from accepting content containing un-encoded HTML. This feature is designed to help prevent some script-injection attacks whereby client script code or HTML can be unknowingly submitted to a server, stored, and then presented to other users. We still strongly recommend that you validate all input data and HTML encode it when appropriate.</p> <p>Request validation is performed by comparing all input data to a list of potentially dangerous values. If a match occurs, ASP.NET raises an <code>HttpRequestValidationException</code>. By default, Request Validation feature is enabled.</p>

Example

However, this feature can be disabled at page level:

```
<%@ Page validateRequest="false" %>
```

or, at application level

```
<configuration>
  <system.web>
    <pages validateRequest="false" />
  </system.web>
</configuration>
```

Please note that Request Validation feature is not supported, and is not part of MVC6 pipeline.

Use locally-hosted latest versions of JavaScript libraries

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>Developers using standard JavaScript libraries like JQuery must use approved versions of common JavaScript libraries that do not contain known security flaws. A good practice is to use the most latest version of the libraries, since they contain security fixes for known vulnerabilities in their older versions.</p> <p>If the most recent release cannot be used due to compatibility reasons, the below minimum versions should be used.</p> <p>Acceptable minimum versions:</p> <ul style="list-style-type: none"> • JQuery <ul style="list-style-type: none"> ◦ JQuery 1.7.1 ◦ JQueryUI 1.10.0 ◦ JQuery Validate 1.9 ◦ JQuery Mobile 1.0.1 ◦ JQuery Cycle 2.99 ◦ JQuery DataTables 1.9.0 • Ajax Control Toolkit <ul style="list-style-type: none"> ◦ Ajax Control Toolkit 40412 • ASP.NET Web Forms and Ajax <ul style="list-style-type: none"> ◦ ASP.NET Web Forms and Ajax 4 ◦ ASP.NET Ajax 3.5 • ASP.NET MVC <ul style="list-style-type: none"> ◦ ASP.NET MVC 3.0 <p>Never load any JavaScript library from external sites such as public CDNs</p>

Disable automatic MIME sniffing

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	IE8 Security Part V: Comprehensive Protection, MIME type
Steps	<p>The X-Content-Type-Options header is an HTTP header that allows developers to specify that their content should not be MIME-sniffed. This header is designed to mitigate MIME-Sniffing attacks. For each page that could contain user controllable content, you must use the HTTP Header X-Content-Type-Options:nosniff. To enable the required header globally for all pages in the application, you can do one of the following</p>

Example

Add the header in the web.config file if the application is hosted by Internet Information Services (IIS) 7 onwards.

```
<system.webServer>
<httpProtocol>
<customHeaders>
<add name="X-Content-Type-Options" value="nosniff"/>
</customHeaders>
</httpProtocol>
</system.webServer>
```

Example

Add the header through the global Application_BeginRequest

```
void Application_BeginRequest(object sender, EventArgs e)
{
    this.Response.Headers["X-Content-Type-Options"] = "nosniff";
}
```

Example

Implement custom HTTP module

```
public class XContentTypeOptionsModule : IHttpModule
{
    #region IHttpModule Members
    public void Dispose()
    {
    }
    public void Init(HttpApplication context)
    {
        context.PreSendRequestHeaders += new EventHandler(context_PreSendRequestHeaders);
    }
    #endregion
    void context_PreSendRequestHeaders(object sender, EventArgs e)
    {
        HttpApplication application = sender as HttpApplication;
        if (application == null)
            return;
        if (application.Response.Headers["X-Content-Type-Options "] != null)
            return;
        application.Response.Headers.Add("X-Content-Type-Options ", "nosniff");
    }
}
```

Example

You can enable the required header only for specific pages by adding it to individual responses:

```
this.Response.Headers["X-Content-Type-Options"] = "nosniff";
```

Remove standard server headers on Windows Azure Web Sites to avoid fingerprinting

TITLE	DETAILS
Component	Web Application
SDL Phase	Build

TITLE	DETAILS
Applicable Technologies	Generic
Attributes	EnvironmentType - Azure
References	Removing standard server headers on Windows Azure Web Sites
Steps	Headers such as Server, X-Powered-By, X-AspNet-Version reveal information about the server and the underlying technologies. It is recommended to suppress these headers thereby preventing fingerprinting the application

Configure a Windows Firewall for Database Engine Access

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	SQL Azure, OnPrem
Attributes	N/A, SQL Version - V12
References	How to configure an Azure SQL database firewall , Configure a Windows Firewall for Database Engine Access
Steps	Firewall systems help prevent unauthorized access to computer resources. To access an instance of the SQL Server Database Engine through a firewall, you must configure the firewall on the computer running SQL Server to allow access

Ensure that only trusted origins are allowed if CORS is enabled on ASP.NET Web API

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	MVC 5
Attributes	N/A
References	Enabling Cross-Origin Requests in ASP.NET Web API 2 , ASP.NET Web API - CORS Support in ASP.NET Web API 2

TITLE	DETAILS
Steps	<p>Browser security prevents a web page from making AJAX requests to another domain. This restriction is called the same-origin policy, and prevents a malicious site from reading sensitive data from another site. However, sometimes it might be required to expose APIs securely which other sites can consume. Cross Origin Resource Sharing (CORS) is a W3C standard that allows a server to relax the same-origin policy.</p> <p>Using CORS, a server can explicitly allow some cross-origin requests while rejecting others. CORS is safer and more flexible than earlier techniques such as JSONP.</p>

Example

In the App_Start/WebApiConfig.cs, add the following code to the WebApiConfig.Register method

```
using System.Web.Http;
namespace WebService
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // New code
            config.EnableCors();

            config.Routes.MapHttpRoute(
                name: "DefaultApi",
                routeTemplate: "api/{controller}/{id}",
                defaults: new { id = RouteParameter.Optional }
            );
        }
    }
}
```

Example

EnableCors attribute can be applied to action methods in a controller as follows:

```

public class ResourcesController : ApiController
{
    [EnableCors("http://localhost:55912", // Origin
                null,                  // Request headers
                "GET",                 // HTTP methods
                "bar",                 // Response headers
                SupportsCredentials=true // Allow credentials
    )]
    public HttpResponseMessage Get(int id)
    {
        var resp = Request.CreateResponse(HttpStatusCode.NoContent);
        resp.Headers.Add("bar", "a bar value");
        return resp;
    }
    [EnableCors("http://localhost:55912",      // Origin
                "Accept, Origin, Content-Type", // Request headers
                "PUT",                      // HTTP methods
                PreflightMaxAge=600          // Preflight cache duration
    )]
    public HttpResponseMessage Put(Resource data)
    {
        return Request.CreateResponse(HttpStatusCode.OK, data);
    }
    [EnableCors("http://localhost:55912",      // Origin
                "Accept, Origin, Content-Type", // Request headers
                "POST",                     // HTTP methods
                PreflightMaxAge=600          // Preflight cache duration
    )]
    public HttpResponseMessage Post(Resource data)
    {
        return Request.CreateResponse(HttpStatusCode.OK, data);
    }
}

```

Please note that it is critical to ensure that the list of origins in EnableCors attribute is set to a finite and trusted set of origins. Failing to configure this inappropriately (e.g., setting the value as '*') will allow malicious sites to trigger cross origin requests to the API without any restrictions, thereby making the API vulnerable to CSRF attacks. EnableCors can be decorated at controller level.

Example

To disable CORS on a particular method in a class, the DisableCors attribute can be used as shown below:

```

[EnableCors("http://example.com", "Accept, Origin, Content-Type", "POST")]
public class ResourcesController : ApiController
{
    public HttpResponseMessage Put(Resource data)
    {
        return Request.CreateResponse(HttpStatusCode.OK, data);
    }
    public HttpResponseMessage Post(Resource data)
    {
        return Request.CreateResponse(HttpStatusCode.OK, data);
    }
    // CORS not allowed because of the [DisableCors] attribute
    [DisableCors]
    public HttpResponseMessage Delete(int id)
    {
        return Request.CreateResponse(HttpStatusCode.NoContent);
    }
}

```

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	MVC 6
Attributes	N/A
References	Enabling Cross-Origin Requests (CORS) in ASP.NET Core 1.0
Steps	In ASP.NET Core 1.0, CORS can be enabled either using middleware or using MVC. When using MVC to enable CORS the same CORS services are used, but the CORS middleware is not.

Approach-1 Enabling CORS with middleware: To enable CORS for the entire application add the CORS middleware to the request pipeline using the `UseCors` extension method. A cross-origin policy can be specified when adding the CORS middleware using the `CorsPolicyBuilder` class. There are two ways to do this:

Example

The first is to call `UseCors` with a lambda. The lambda takes a `CorsPolicyBuilder` object:

```
public void Configure(IApplicationBuilder app)
{
    app.UseCors(builder =>
        builder.WithOrigins("http://example.com")
            .WithMethods("GET", "POST", "HEAD")
            .WithHeaders("accept", "content-type", "origin", "x-custom-header"));
}
```

Example

The second is to define one or more named CORS policies, and then select the policy by name at run time.

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddCors(options =>
    {
        options.AddPolicy("AllowSpecificOrigin",
            builder => builder.WithOrigins("http://example.com"));
    });
}

public void Configure(IApplicationBuilder app)
{
    app.UseCors("AllowSpecificOrigin");
    app.Run(async (context) =>
    {
        await context.Response.WriteAsync("Hello World!");
    });
}
```

Approach-2 Enabling CORS in MVC: Developers can alternatively use MVC to apply specific CORS per action, per controller, or globally for all controllers.

Example

Per action: To specify a CORS policy for a specific action add the [EnableCors] attribute to the action. Specify the policy name.

```
public class HomeController : Controller
{
    [EnableCors("AllowSpecificOrigin")]
    public IActionResult Index()
    {
        return View();
    }
}
```

Example

Per controller:

```
[EnableCors("AllowSpecificOrigin")]
public class HomeController : Controller
{
```

Example

Globally:

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc();
    services.Configure<MvcOptions>(options =>
    {
        options.Filters.Add(new CorsAuthorizationFilterFactory("AllowSpecificOrigin"));
    });
}
```

Please note that it is critical to ensure that the list of origins in EnableCors attribute is set to a finite and trusted set of origins. Failing to configure this inappropriately (e.g., setting the value as '*') will allow malicious sites to trigger cross origin requests to the API without any restrictions, thereby making the API vulnerable to CSRF attacks.

Example

To disable CORS for a controller or action, use the [DisableCors] attribute.

```
[DisableCors]
public IActionResult About()
{
    return View();
}
```

Encrypt sections of Web API's configuration files that contain sensitive data

TITLE	DETAILS
Component	Web API
SDL Phase	Deployment
Applicable Technologies	Generic

TITLE	DETAILS
Attributes	N/A
References	How To: Encrypt Configuration Sections in ASP.NET 2.0 Using DPAPI, Specifying a Protected Configuration Provider, Using Azure Key Vault to protect application secrets
Steps	Configuration files such as the Web.config, appsettings.json are often used to hold sensitive information, including user names, passwords, database connection strings, and encryption keys. If you do not protect this information, your application is vulnerable to attackers or malicious users obtaining sensitive information such as account user names and passwords, database names and server names. Based on the deployment type (azure/on-prem), encrypt the sensitive sections of config files using DPAPI or services like Azure Key Vault.

Ensure that all admin interfaces are secured with strong credentials

TITLE	DETAILS
Component	IoT Device
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Any administrative interfaces that the device or field gateway exposes should be secured using strong credentials. Also, any other exposed interfaces like WiFi, SSH, File shares, FTP should be secured with strong credentials. Default weak passwords should not be used.

Ensure that unknown code cannot execute on devices

TITLE	DETAILS
Component	IoT Device
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Enabling Secure Boot and bit-locker Device Encryption on Windows 10 IoT Core

TITLE	DETAILS
Steps	UEFI Secure Boot restricts the system to only allow execution of binaries signed by a specified authority. This feature prevents unknown code from being executed on the platform and potentially weakening the security posture of it. Enable UEFI Secure Boot and restrict the list of certificate authorities that are trusted for signing code. Sign all code that is deployed on the device using one of the trusted authorities.

Encrypt OS and additional partitions of IoT Device with bit-locker

TITLE	DETAILS
Component	IoT Device
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Windows 10 IoT Core implements a lightweight version of bit-locker Device Encryption, which has a strong dependency on the presence of a TPM on the platform, including the necessary preOS protocol in UEFI that conducts the necessary measurements. These preOS measurements ensure that the OS later has a definitive record of how the OS was launched. Encrypt OS partitions using bit-locker and any additional partitions also in case they store any sensitive data.

Ensure that only the minimum services/features are enabled on devices

TITLE	DETAILS
Component	IoT Device
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Do not enable or turn off any features or services in the OS that is not required for the functioning of the solution. For e.g. if the device does not require a UI to be deployed, install Windows IoT Core in headless mode.

Encrypt OS and additional partitions of IoT Field Gateway with bit-

locker

TITLE	DETAILS
Component	IoT Field Gateway
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Windows 10 IoT Core implements a lightweight version of bit-locker Device Encryption, which has a strong dependency on the presence of a TPM on the platform, including the necessary preOS protocol in UEFI that conducts the necessary measurements. These preOS measurements ensure that the OS later has a definitive record of how the OS was launched. Encrypt OS partitions using bit-locker and any additional partitions also in case they store any sensitive data.

Ensure that the default login credentials of the field gateway are changed during installation

TITLE	DETAILS
Component	IoT Field Gateway
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that the default login credentials of the field gateway are changed during installation

Ensure that the Cloud Gateway implements a process to keep the connected devices firmware up to date

TITLE	DETAILS
Component	IoT Cloud Gateway
SDL Phase	Build
Applicable Technologies	Generic

TITLE	DETAILS
Attributes	Gateway choice - Azure IoT Hub
References	IoT Hub Device Management Overview , How to update Device Firmware
Steps	LWM2M is a protocol from the Open Mobile Alliance for IoT Device Management. Azure IoT device management allows to interact with physical devices using device jobs. Ensure that the Cloud Gateway implements a process to routinely keep the device and other configuration data up to date using Azure IoT Hub Device Management.

Ensure that devices have end-point security controls configured as per organizational policies

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that devices have end-point security controls such as bit-locker for disk-level encryption, anti-virus with updated signatures, host based firewall, OS upgrades, group policies etc. are configured as per organizational security policies.

Ensure secure management of Azure storage access keys

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	Azure Storage security guide - Managing Your Storage Account Keys

TITLE	DETAILS
Steps	<p>Key Storage: It is recommended to store the Azure Storage access keys in Azure Key Vault as a secret and have the applications retrieve the key from key vault. This is recommended due to the following reasons:</p> <ul style="list-style-type: none"> • The application will never have the storage key hardcoded in a configuration file, which removes that avenue of somebody getting access to the keys without specific permission • Access to the keys can be controlled using Azure Active Directory. This means an account owner can grant access to the handful of applications that need to retrieve the keys from Azure Key Vault. Other applications will not be able to access the keys without granting them permission specifically • Key Regeneration: It is recommended to have a process in place to regenerate Azure storage access keys for security reasons. Details on why and how to plan for key regeneration are documented in the Azure Storage Security Guide reference article

Ensure that only trusted origins are allowed if CORS is enabled on Azure storage

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	CORS Support for the Azure Storage Services
Steps	Azure Storage allows you to enable CORS – Cross Origin Resource Sharing. For each storage account, you can specify domains that can access the resources in that storage account. By default, CORS is disabled on all services. You can enable CORS by using the REST API or the storage client library to call one of the methods to set the service policies.

Enable WCF's service throttling feature

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework 3

TITLE	DETAILS
Attributes	N/A
References	MSDN , Fortify Kingdom
Steps	<p>Not placing a limit on the use of system resources could result in resource exhaustion and ultimately a denial of service.</p> <ul style="list-style-type: none"> • EXPLANATION: Windows Communication Foundation (WCF) offers the ability to throttle service requests. Allowing too many client requests can flood a system and exhaust its resources. On the other hand, allowing only a small number of requests to a service can prevent legitimate users from using the service. Each service should be individually tuned to and configured to allow the appropriate amount of resources. • RECOMMENDATIONS Enable WCF's service throttling feature and set limits appropriate for your application.

Example

The following is an example configuration with throttling enabled:

```
<system.serviceModel>
  <behaviors>
    <serviceBehaviors>
      <behavior name="Throttled">
        <serviceThrottling maxConcurrentCalls="[YOUR SERVICE VALUE]" maxConcurrentSessions="[YOUR SERVICE VALUE]"
maxConcurrentInstances="[YOUR SERVICE VALUE]" />
        ...
    </serviceBehaviors>
  </behaviors>
</system.serviceModel>
```

WCF-Information disclosure through metadata

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework 3
Attributes	N/A
References	MSDN , Fortify Kingdom
Steps	<p>Metadata can help attackers learn about the system and plan a form of attack. WCF services can be configured to expose metadata. Metadata gives detailed service description information and should not be broadcast in production environments. The <code>HttpGetEnabled</code> / <code>HttpsGetEnabled</code> properties of the ServiceMetaData class defines whether a service will expose the metadata</p>

Example

The code below instructs WCF to broadcast a service's metadata

```
ServiceMetadataBehavior smb = new ServiceMetadataBehavior();
smb.HttpGetEnabled = true;
smb.HttpGetUrl = new Uri(EndPointAddress);
Host.Description.Behaviors.Add(smb);
```

Do not broadcast service metadata in a production environment. Set the `HttpGetEnabled` / `HttpsGetEnabled` properties of the `ServiceMetaData` class to false.

Example

The code below instructs WCF to not broadcast a service's metadata.

```
ServiceMetadataBehavior smb = new ServiceMetadataBehavior();
smb.HttpGetEnabled = false;
smb.HttpGetUrl = new Uri(EndPointAddress);
Host.Description.Behaviors.Add(smb);
```

Security Frame: Cryptography | Mitigations

12/3/2018 • 14 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Web Application	<ul style="list-style-type: none">• Use only approved symmetric block ciphers and key lengths• Use approved block cipher modes and initialization vectors for symmetric ciphers• Use approved asymmetric algorithms, key lengths, and padding• Use approved random number generators• Do not use symmetric stream ciphers• Use approved MAC/HMAC/keyed hash algorithms• Use only approved cryptographic hash functions
Database	<ul style="list-style-type: none">• Use strong encryption algorithms to encrypt data in the database• SSIS packages should be encrypted and digitally signed• Add digital signature to critical database securables• Use SQL server EKM to protect encryption keys• Use AlwaysEncrypted feature if encryption keys should not be revealed to Database engine
IoT Device	<ul style="list-style-type: none">• Store Cryptographic Keys securely on IoT Device
IoT Cloud Gateway	<ul style="list-style-type: none">• Generate a random symmetric key of sufficient length for authentication to IoT Hub
Dynamics CRM Mobile Client	<ul style="list-style-type: none">• Ensure a device management policy is in place that requires a use PIN and allows remote wiping
Dynamics CRM Outlook Client	<ul style="list-style-type: none">• Ensure a device management policy is in place that requires a PIN/password/auto lock and encrypts all data (e.g. Bitlocker)
Identity Server	<ul style="list-style-type: none">• Ensure that signing keys are rolled over when using Identity Server• Ensure that cryptographically strong client ID, client secret are used in Identity Server

Use only approved symmetric block ciphers and key lengths

TITLE	DETAILS
Component	Web Application

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>Products must use only those symmetric block ciphers and associated key lengths which have been explicitly approved by the Crypto Advisor in your organization. Approved symmetric algorithms at Microsoft include the following block ciphers:</p> <ul style="list-style-type: none"> • For new code AES-128, AES-192, and AES-256 are acceptable • For backward compatibility with existing code, three-key 3DES is acceptable • For products using symmetric block ciphers: <ul style="list-style-type: none"> ◦ Advanced Encryption Standard (AES) is required for new code ◦ Three-key triple Data Encryption Standard (3DES) is permissible in existing code for backward compatibility ◦ All other block ciphers, including RC2, DES, 2 Key 3DES, DESX, and Skipjack, may only be used for decrypting old data, and must be replaced if used for encryption • For symmetric block encryption algorithms, a minimum key length of 128 bits is required. The only block encryption algorithm recommended for new code is AES (AES-128, AES-192 and AES-256 are all acceptable) • Three-key 3DES is currently acceptable if already in use in existing code; transition to AES is recommended. DES, DESX, RC2, and Skipjack are no longer considered secure. These algorithms may only be used for decrypting existing data for the sake of backward-compatibility, and data should be re-encrypted using a recommended block cipher <p>Please note that all symmetric block ciphers must be used with an approved cipher mode, which requires use of an appropriate initialization vector (IV). An appropriate IV, is typically a random number and never a constant value</p> <p>The use of legacy or otherwise unapproved crypto algorithms and smaller key lengths for reading existing data (as opposed to writing new data) may be permitted after your organization's Crypto Board review. However, you must file for an exception against this requirement. Additionally, in enterprise deployments, products should consider warning administrators when weak crypto is used to read data. Such warnings should be explanatory and actionable. In some cases, it may be appropriate to have Group Policy control the use of weak crypto</p> <p>Allowed .NET algorithms for managed crypto agility (in order of preference)</p> <ul style="list-style-type: none"> • AesCng (FIPS compliant)

TITLE	DETAILS
	<ul style="list-style-type: none"> AuthenticatedAesCng (FIPS compliant) AESCryptoServiceProvider (FIPS compliant) AESManaged (non-FIPS-compliant) <p>Please note that none of these algorithms can be specified via the <code>SymmetricAlgorithm.Create</code> or <code>CryptoConfig.CreateFromName</code> methods without making changes to the machine.config file. Also, note that AES in versions of .NET prior to .NET 3.5 is named <code>RijndaelManaged</code>, and <code>AesCng</code> and <code>AuthenticatedAesCng</code> are >available through CodePlex and require CNG in the underlying OS</p>

Use approved block cipher modes and initialization vectors for symmetric ciphers

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>All symmetric block ciphers must be used with an approved symmetric cipher mode. The only approved modes are CBC and CTS. In particular, the electronic code book (ECB) mode of operation should be avoided; use of ECB requires your organization's Crypto Board review. All usage of OFB, CFB, CTR, CCM, and GCM or any other encryption mode must be reviewed by your organization's Crypto Board. Reusing the same initialization vector (IV) with block ciphers in "streaming ciphers modes," such as CTR, may cause encrypted data to be revealed. All symmetric block ciphers must also be used with an appropriate initialization vector (IV). An appropriate IV is a cryptographically strong, random number and never a constant value.</p>

Use approved asymmetric algorithms, key lengths, and padding

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	N/A
Steps	<p>The use of banned cryptographic algorithms introduces significant risk to product security and must be avoided. Products must use only those cryptographic algorithms and associated key lengths and padding that have been explicitly approved by your organization's Crypto Board.</p> <ul style="list-style-type: none"> • RSA- may be used for encryption, key exchange and signature. RSA encryption must use only the OAEP or RSA-KEM padding modes. Existing code may use PKCS #1 v1.5 padding mode for compatibility only. Use of null padding is explicitly banned. Keys \geq 2048 bits is required for new code. Existing code may support keys $<$ 2048 bits only for backwards compatibility after a review by your organization's Crypto Board. Keys $<$ 1024 bits may only be used for decrypting/verifying old data, and must be replaced if used for encryption or signing operations • ECDSA- may be used for signature only. ECDSA with \geq=256-bit keys is required for new code. ECDSA-based signatures must use one of the three NIST approved curves (P-256, P-384, or P521). Curves that have been thoroughly analyzed may be used only after a review with your organization's Crypto Board. • ECDH- may be used for key exchange only. ECDH with \geq=256-bit keys is required for new code. ECDH-based key exchange must use one of the three NIST approved curves (P-256, P-384, or P521). Curves that have been thoroughly analyzed may be used only after a review with your organization's Crypto Board. • DSA- may be acceptable after review and approval from your organization's Crypto Board. Contact your security advisor to schedule your organization's Crypto Board review. If your use of DSA is approved, note that you will need to prohibit use of keys less than 2048 bits in length. CNG supports 2048-bit and greater key lengths as of Windows 8. • Diffie-Hellman- may be used for session key management only. Key length \geq 2048 bits is required for new code. Existing code may support key lengths $<$ 2048 bits only for backwards compatibility after a review by your organization's Crypto Board. Keys $<$ 1024 bits may not be used.

Use approved random number generators

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	N/A
Steps	<p>Products must use approved random number generators. Pseudorandom functions such as the C runtime function rand, the .NET Framework class System.Random, or system functions such as GetTickCount must, therefore, never be used in such code. Use of the dual elliptic curve random number generator (DUAL_EC_DRBG) algorithm is prohibited</p> <ul style="list-style-type: none"> • CNG- BCryptGenRandom(use of the BCRYPT_USE_SYSTEM_PREFERRED_RNG flag recommended unless the caller might run at any IRQL greater than 0 [that is, PASSIVE_LEVEL]) • CAPI- cryptGenRandom • Win32/64- RtlGenRandom (new implementations should use BCryptGenRandom or CryptGenRandom) * rand_s * SystemPrng (for kernel mode) • .NET- RNGCryptoServiceProvider or RNGCng • Windows Store Apps- Windows.Security.Cryptography.CryptographicBuffer.GenerateRandom or .GenerateRandomNumber • Apple OS X (10.7+)/iOS(2.0+)- int SecRandomCopyBytes (SecRandomRef random, size_t count, uint8_t *bytes) • Apple OS X (<10.7)- Use /dev/random to retrieve random numbers • Java(including Google Android Java code)- java.security.SecureRandom class. Note that for Android 4.3 (Jelly Bean), developers must follow the Android recommended workaround and update their applications to explicitly initialize the PRNG with entropy from /dev/urandom or /dev/random

Do not use symmetric stream ciphers

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Symmetric stream ciphers, such as RC4, must not be used. Instead of symmetric stream ciphers, products should use a block cipher, specifically AES with a key length of at least 128 bits.

Use approved MAC/HMAC/keyed hash algorithms

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>Products must use only approved message authentication code (MAC) or hash-based message authentication code (HMAC) algorithms.</p> <p>A message authentication code (MAC) is a piece of information attached to a message that allows its recipient to verify both the authenticity of the sender and the integrity of the message using a secret key. The use of either a hash-based MAC (HMAC) or block-cipher-based MAC is permissible as long as all underlying hash or symmetric encryption algorithms are also approved for use; currently this includes the HMAC-SHA2 functions (HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) and the CMAC/OMAC1 and OMAC2 block cipher-based MACs (these are based on AES).</p> <p>Use of HMAC-SHA1 may be permissible for platform compatibility, but you will be required to file an exception to this procedure and undergo your organization's Crypto review. Truncation of HMACs to less than 128 bits is not permitted. Using customer methods to hash a key and data is not approved, and must undergo your organization's Crypto Board review prior to use.</p>

Use only approved cryptographic hash functions

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>Products must use the SHA-2 family of hash algorithms (SHA256, SHA384, and SHA512). If a shorter hash is needed, such as a 128-bit output length in order to fit a data structure designed with the shorter MD5 hash in mind, product teams may truncate one of the SHA2 hashes (typically SHA256). Note that SHA384 is a truncated version of SHA512. Truncation of cryptographic hashes for security purposes to less than 128 bits is not permitted. New code must not use the MD2, MD4, MD5, SHA-0, SHA-1, or RIPEMD hash algorithms. Hash collisions are computationally feasible for these algorithms, which effectively breaks them.</p> <p>Allowed .NET hash algorithms for managed crypto agility (in order of preference):</p> <ul style="list-style-type: none"> • SHA512Cng (FIPS compliant) • SHA384Cng (FIPS compliant) • SHA256Cng (FIPS compliant) • SHA512Managed (non-FIPS-compliant) (use SHA512 as algorithm name in calls to HashAlgorithm.Create or CryptoConfig.CreateFromName) • SHA384Managed (non-FIPS-compliant) (use SHA384 as algorithm name in calls to HashAlgorithm.Create or CryptoConfig.CreateFromName) • SHA256Managed (non-FIPS-compliant) (use SHA256 as algorithm name in calls to HashAlgorithm.Create or CryptoConfig.CreateFromName) • SHA512CryptoServiceProvider (FIPS compliant) • SHA256CryptoServiceProvider (FIPS compliant) • SHA384CryptoServiceProvider (FIPS compliant)

Use strong encryption algorithms to encrypt data in the database

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Choosing an encryption algorithm
Steps	Encryption algorithms define data transformations that cannot be easily reversed by unauthorized users. SQL Server allows administrators and developers to choose from among several algorithms, including DES, Triple DES, TRIPLE_DES_3KEY, RC2, RC4, 128-bit RC4, DESX, 128-bit AES, 192-bit AES, and 256-bit AES

SSIS packages should be encrypted and digitally signed

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Identify the Source of Packages with Digital Signatures, Threat and Vulnerability Mitigation (Integration Services)
Steps	The source of a package is the individual or organization that created the package. Running a package from an unknown or untrusted source might be risky. To prevent unauthorized tampering of SSIS packages, digital signatures should be used. Also, to ensure the confidentiality of the packages during storage/transit, SSIS packages have to be encrypted

Add digital signature to critical database securables

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	ADD SIGNATURE (Transact-SQL)
Steps	In cases where the integrity of a critical database securable has to be verified, digital signatures should be used. Database securables such as a stored procedure, function, assembly, or trigger can be digitally signed. Below is an example of when this can be useful: Let us say an ISV (Independent Software Vendor) has provided support to a software delivered to one of their customers. Before providing support, the ISV would want to ensure that a database securable in the software was not tampered either by mistake or by a malicious attempt. If the securable is digitally signed, the ISV can verify its digital signature and validate its integrity.

Use SQL server EKM to protect encryption keys

TITLE	DETAILS
Component	Database
SDL Phase	Build

TITLE	DETAILS
Applicable Technologies	Generic
Attributes	N/A
References	SQL Server Extensible Key Management (EKM) , Extensible Key Management Using Azure Key Vault (SQL Server)
Steps	SQL Server Extensible Key Management enables the encryption keys that protect the database files to be stored in an off-box device such as a smartcard, USB device, or EKM/HSM module. This also enables data protection from database administrators (except members of the sysadmin group). Data can be encrypted by using encryption keys that only the database user has access to on the external EKM/HSM module.

Use AlwaysEncrypted feature if encryption keys should not be revealed to Database engine

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	SQL Azure, OnPrem
Attributes	SQL Version - V12, MsSQL2016
References	Always Encrypted (Database Engine)
Steps	Always Encrypted is a feature designed to protect sensitive data, such as credit card numbers or national identification numbers (e.g. U.S. social security numbers), stored in Azure SQL Database or SQL Server databases. Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to the Database Engine (SQL Database or SQL Server). As a result, Always Encrypted provides a separation between those who own the data (and can view it) and those who manage the data (but should have no access)

Store Cryptographic Keys securely on IoT Device

TITLE	DETAILS
Component	IoT Device
SDL Phase	Build
Applicable Technologies	Generic

TITLE	DETAILS
Attributes	Device OS - Windows IoT Core, Device Connectivity - Azure IoT device SDKs
References	TPM on Windows IoT Core , Set up TPM on Windows IoT Core , Azure IoT Device SDK TPM
Steps	<p>Symmetric or Certificate Private keys securely in a hardware protected storage like TPM or Smart Card chips. Windows 10 IoT Core supports the user of a TPM and there are several compatible TPMs that can be used:</p> <p>https://docs.microsoft.com/windows/iot-core/secure-your-device/tpm#discrete-tpm-dtpm. It is recommended to use a Firmware or Discrete TPM. A Software TPM should only be used for development and testing purposes. Once a TPM is available and the keys are provisioned in it, the code that generates the token should be written without hard coding any sensitive information in it.</p>

Example

```

TpmDevice myDevice = new TpmDevice(0);
// Use logical device 0 on the TPM
string hubUri = myDevice.GetHostName();
string deviceId = myDevice.GetDeviceId();
string sasToken = myDevice.GetSASToken();

var deviceClient = DeviceClient.Create( hubUri, AuthenticationMethodFactory.
CreateAuthenticationWithToken(deviceId, sasToken), TransportType.Amqp);

```

As can be seen, the device primary key is not present in the code. Instead, it is stored in the TPM at slot 0. TPM device generates a short-lived SAS token that is then used to connect to the IoT Hub.

Generate a random symmetric key of sufficient length for authentication to IoT Hub

TITLE	DETAILS
Component	IoT Cloud Gateway
SDL Phase	Build
Applicable Technologies	Generic
Attributes	Gateway choice - Azure IoT Hub
References	N/A

TITLE	DETAILS
Steps	IoT Hub contains a device Identity Registry and while provisioning a device, automatically generates a random Symmetric key. It is recommended to use this feature of the Azure IoT Hub Identity Registry to generate the key used for authentication. IoT Hub also allows for a key to be specified while creating the device. If a key is generated outside of IoT Hub during device provisioning, it is recommended to create a random symmetric key or at least 256 bits.

Ensure a device management policy is in place that requires a use PIN and allows remote wiping

TITLE	DETAILS
Component	Dynamics CRM Mobile Client
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure a device management policy is in place that requires a use PIN and allows remote wiping

Ensure a device management policy is in place that requires a PIN/password/auto lock and encrypts all data (e.g. Bitlocker)

TITLE	DETAILS
Component	Dynamics CRM Outlook Client
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure a device management policy is in place that requires a PIN/password/auto lock and encrypts all data (e.g. Bitlocker)

Ensure that signing keys are rolled over when using Identity Server

TITLE	DETAILS
Component	Identity Server
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	Identity Server - Keys, Signatures and Cryptography
Steps	Ensure that signing keys are rolled over when using Identity Server. The link in the references section explains how this should be planned without causing outages to applications relying on Identity Server.

Ensure that cryptographically strong client ID, client secret are used in Identity Server

TITLE	DETAILS
Component	Identity Server
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that cryptographically strong client ID, client secret are used in Identity Server. The following guidelines should be used while generating a client ID and secret: <ul style="list-style-type: none"> • Generate a random GUID as the client ID • Generate a cryptographically random 256-bit key as the secret

Security Frame: Exception Management | Mitigations

8/30/2018 • 7 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
WCF	<ul style="list-style-type: none">• WCF- Do not include serviceDebug node in configuration file• WCF- Do not include serviceMetadata node in configuration file
Web API	<ul style="list-style-type: none">• Ensure that proper exception handling is done in ASP.NET Web API
Web Application	<ul style="list-style-type: none">• Do not expose security details in error messages• Implement Default error handling page• Set Deployment Method to Retail in IIS• Exceptions should fail safely

WCF- Do not include serviceDebug node in configuration file

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	N/A
References	MSDN , Fortify Kingdom
Steps	Windows Communication Framework (WCF) services can be configured to expose debugging information. Debug information should not be used in production environments. The <code><serviceDebug></code> tag defines whether the debug information feature is enabled for a WCF service. If the attribute <code>includeExceptionDetailInFaults</code> is set to true, exception information from the application will be returned to clients. Attackers can leverage the additional information they gain from debugging output to mount attacks targeted on the framework, database, or other resources used by the application.

Example

The following configuration file includes the `<serviceDebug>` tag:

```

<configuration>
<system.serviceModel>
<behaviors>
<serviceBehaviors>
<behavior name=""MyServiceBehavior"">
<serviceDebug includeExceptionDetailInFaults=""True"" httpHelpPageEnabled=""True""/>
...

```

Disable debugging information in the service. This can be accomplished by removing the `<serviceDebug>` tag from your application's configuration file.

WCF- Do not include serviceMetadata node in configuration file

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic
Attributes	Generic, .NET Framework 3
References	MSDN , Fortify Kingdom
Steps	Publicly exposing information about a service can provide attackers with valuable insight into how they might exploit the service. The <code><serviceMetadata></code> tag enables the metadata publishing feature. Service metadata could contain sensitive information that should not be publicly accessible. At a minimum, only allow trusted users to access the metadata and ensure that unnecessary information is not exposed. Better yet, entirely disable the ability to publish metadata. A safe WCF configuration will not contain the <code><serviceMetadata></code> tag.

Ensure that proper exception handling is done in ASP.NET Web API

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	MVC 5, MVC 6
Attributes	N/A
References	Exception Handling in ASP.NET Web API , Model Validation in ASP.NET Web API
Steps	By default, most uncaught exceptions in ASP.NET Web API are translated into an HTTP response with status code <code>500, Internal Server Error</code>

Example

To control the status code returned by the API, `HttpResponseException` can be used as shown below:

```
public Product GetProduct(int id)
{
    Product item = repository.Get(id);
    if (item == null)
    {
        throw new HttpResponseException(HttpStatusCode.NotFound);
    }
    return item;
}
```

Example

For further control on the exception response, the `HttpResponseMessage` class can be used as shown below:

```
public Product GetProduct(int id)
{
    Product item = repository.Get(id);
    if (item == null)
    {
        var resp = new HttpResponseMessage(HttpStatusCode.NotFound)
        {
            Content = new StringContent(string.Format("No product with ID = {0}", id)),
            ReasonPhrase = "Product ID Not Found"
        }
        throw new HttpResponseException(resp);
    }
    return item;
}
```

To catch unhandled exceptions that are not of the type `HttpResponseException`, Exception Filters can be used.

Exception filters implement the `System.Web.Http.Filters.IExceptionFilter` interface. The simplest way to write an exception filter is to derive from the `System.Web.Http.Filters.ExceptionFilterAttribute` class and override the `OnException` method.

Example

Here is a filter that converts `NotImplementedException` exceptions into HTTP status code `501, Not Implemented`:

```
namespace ProductStore.Filters
{
    using System;
    using System.Net;
    using System.Net.Http;
    using System.Web.Http.Filters;

    public class NotImplExceptionFilterAttribute : ExceptionFilterAttribute
    {
        public override void OnException(HttpActionExecutedContext context)
        {
            if (context.Exception is NotImplementedException)
            {
                context.Response = new HttpResponseMessage(HttpStatusCode.NotImplemented);
            }
        }
    }
}
```

There are several ways to register a Web API exception filter:

- By action
- By controller
- Globally

Example

To apply the filter to a specific action, add the filter as an attribute to the action:

```
public class ProductsController : ApiController
{
    [NotImplExceptionFilter]
    public Contact GetContact(int id)
    {
        throw new NotImplementedException("This method is not implemented");
    }
}
```

Example

To apply the filter to all of the actions on a `controller`, add the filter as an attribute to the `controller` class:

```
[NotImplExceptionFilter]
public class ProductsController : ApiController
{
    // ...
}
```

Example

To apply the filter globally to all Web API controllers, add an instance of the filter to the `GlobalConfiguration.Configuration.Filters` collection. Exception filters in this collection apply to any Web API controller action.

```
GlobalConfiguration.Configuration.Filters.Add(
    new ProductStore.NotImplExceptionFilterAttribute());
```

Example

For model validation, the model state can be passed to `CreateErrorResponse` method as shown below:

```
public HttpResponseMessage PostProduct(Product item)
{
    if (!ModelState.IsValid)
    {
        return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ModelState);
    }
    // Implementation not shown...
}
```

Check the links in the references section for additional details about exceptional handling and model validation in ASP.NET Web API

Do not expose security details in error messages

TITLE	DETAILS
Component	Web Application

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>Generic error messages are provided directly to the user without including sensitive application data. Examples of sensitive data include:</p> <ul style="list-style-type: none"> • Server names • Connection strings • Usernames • Passwords • SQL procedures • Details of dynamic SQL failures • Stack trace and lines of code • Variables stored in memory • Drive and folder locations • Application install points • Host configuration settings • Other internal application details <p>Trapping all errors within an application and providing generic error messages, as well as enabling custom errors within IIS will help prevent information disclosure. SQL Server database and .NET Exception handling, among other error handling architectures, are especially verbose and extremely useful to a malicious user profiling your application. Do not directly display the contents of a class derived from the .NET Exception class, and ensure that you have proper exception handling so that an unexpected exception isn't inadvertently raised directly to the user.</p> <ul style="list-style-type: none"> • Provide generic error messages directly to the user that abstract away specific details found directly in the exception/error message • Do not display the contents of a .NET exception class directly to the user • Trap all error messages and if appropriate inform the user via a generic error message sent to the application client • Do not expose the contents of the Exception class directly to the user, especially the return value from <code>.ToString()</code>, or the values of the Message or StackTrace properties. Securely log this information and display a more innocuous message to the user

Implement Default error handling page

TITLE	DETAILS
Component	Web Application

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Edit ASP.NET Error Pages Settings Dialog Box
Steps	<p>When an ASP.NET application fails and causes an HTTP/1.x 500 Internal Server Error, or a feature configuration (such as Request Filtering) prevents a page from being displayed, an error message will be generated. Administrators can choose whether or not the application should display a friendly message to the client, detailed error message to the client, or detailed error message to localhost only. The tag in the web.config has three modes:</p> <ul style="list-style-type: none"> • On: Specifies that custom errors are enabled. If no defaultRedirect attribute is specified, users see a generic error. The custom errors are shown to the remote clients and to the local host • Off: Specifies that custom errors are disabled. The detailed ASP.NET errors are shown to the remote clients and to the local host • RemoteOnly: Specifies that custom errors are shown only to the remote clients, and that ASP.NET errors are shown to the local host. This is the default value <p>Open the <code>web.config</code> file for the application/site and ensure that the tag has either <code><customErrors mode="RemoteOnly" /></code> or <code><customErrors mode="On" /></code> defined.</p>

Set Deployment Method to Retail in IIS

TITLE	DETAILS
Component	Web Application
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	deployment Element (ASP.NET Settings Schema)

TITLE	DETAILS
Steps	<p>The <code><deployment retail></code> switch is intended for use by production IIS servers. This switch is used to help applications run with the best possible performance and least possible security information leakages by disabling the application's ability to generate trace output on a page, disabling the ability to display detailed error messages to end users, and disabling the debug switch.</p> <p>Often times, switches and options that are developer-focused, such as failed request tracing and debugging, are enabled during active development. It is recommended that the deployment method on any production server be set to retail. Open the machine.config file and ensure that <code><deployment retail="true" /></code> remains set to true.</p>

Exceptions should fail safely

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Fail securely
Steps	<p>Application should fail safely. Any method that returns a Boolean value, based on which certain decision is made, should have exception block carefully created. There are lot of logical errors due to which security issues creep in, when the exception block is written carelessly.</p>

Example

```

public static bool ValidateDomain(string pathToValidate, Uri currentUrl)
{
    try
    {
        if (!string.IsNullOrWhiteSpace(pathToValidate))
        {
            var domain = RetrieveDomain(currentUrl);
            var replyPath = new Uri(pathToValidate);
            var replyDomain = RetrieveDomain(replyPath);

            if (string.Compare(domain, replyDomain, StringComparison.OrdinalIgnoreCase) != 0)
            {
                //// Adding additional check to enable CMS urls if they are not hosted on same domain.
                if (!string.IsNullOrWhiteSpace(Utilities.CmsBase))
                {
                    var cmsDomain = RetrieveDomain(new Uri(Utilities.Base.Trim()));
                    if (string.Compare(cmsDomain, replyDomain, StringComparison.OrdinalIgnoreCase) != 0)
                    {
                        return false;
                    }
                    else
                    {
                        return true;
                    }
                }
            }

            return false;
        }
    }

    return true;
}
catch (UriFormatException ex)
{
    LogHelper.LogException("Utilities:ValidateDomain", ex);
    return true;
}
}

```

The above method will always return True, if some exception happens. If the end user provides a malformed URL, that the browser respects, but the `Uri()` constructor doesn't, this will throw an exception, and the victim will be taken to the valid but malformed URL.

Security Frame: Input Validation | Mitigations

11/7/2018 • 31 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Web Application	<ul style="list-style-type: none">• Disable XSLT scripting for all transforms using untrusted style sheets• Ensure that each page that could contain user controllable content opts out of automatic MIME sniffing• Harden or Disable XML Entity Resolution• Applications utilizing http.sys perform URL canonicalization verification• Ensure appropriate controls are in place when accepting files from users• Ensure that type-safe parameters are used in Web Application for data access• Use separate model binding classes or binding filter lists to prevent MVC mass assignment vulnerability• Encode untrusted web output prior to rendering• Perform input validation and filtering on all string type Model properties• Sanitization should be applied on form fields that accept all characters, e.g. rich text editor• Do not assign DOM elements to sinks that do not have inbuilt encoding• Validate all redirects within the application are closed or done safely• Implement input validation on all string type parameters accepted by Controller methods• Set upper limit timeout for regular expression processing to prevent DoS due to bad regular expressions• Avoid using Html.Raw in Razor views
Database	<ul style="list-style-type: none">• Do not use dynamic queries in stored procedures
Web API	<ul style="list-style-type: none">• Ensure that model validation is done on Web API methods• Implement input validation on all string type parameters accepted by Web API methods• Ensure that type-safe parameters are used in Web API for data access
Azure Document DB	<ul style="list-style-type: none">• Use parametrized SQL queries for Azure Cosmos DB
WCF	<ul style="list-style-type: none">• WCF Input validation through Schema binding• WCF- Input validation through Parameter Inspectors

Disable XSLT scripting for all transforms using untrusted style sheets

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	XSLT Security , XsltSettings.EnableScript Property
Steps	XSLT supports scripting inside style sheets using the <code><msxml:script></code> element. This allows custom functions to be used in an XSLT transformation. The script is executed under the context of the process performing the transform. XSLT script must be disabled when in an untrusted environment to prevent execution of untrusted code. <i>If using .NET:</i> XSLT scripting is disabled by default; however, you must ensure that it has not been explicitly enabled through the <code>XsltSettings.EnableScript</code> property.

Example

```
XsltSettings settings = new XsltSettings();
settings.EnableScript = true; // WRONG: THIS SHOULD BE SET TO false
```

Example

If you are using MSXML 6.0, XSLT scripting is disabled by default; however, you must ensure that it has not been explicitly enabled through the XML DOM object property `AllowXsltScript`.

```
doc.setProperty("AllowXsltScript", true); // WRONG: THIS SHOULD BE SET TO false
```

Example

If you are using MSXML 5 or below, XSLT scripting is enabled by default and you must explicitly disable it. Set the XML DOM object property `AllowXsltScript` to false.

```
doc.setProperty("AllowXsltScript", false); // CORRECT. Setting to false disables XSLT scripting.
```

Ensure that each page that could contain user controllable content opts out of automatic MIME sniffing

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	IE8 Security Part V - Comprehensive Protection
Steps	<p>For each page that could contain user controllable content, you must use the HTTP Header <code>X-Content-Type-Options:nosniff</code>. To comply with this requirement, you can either set the required header page by page for only those pages that might contain user-controllable content, or you can set it globally for all pages in the application.</p> <p>Each type of file delivered from a web server has an associated MIME type (also called a <i>content-type</i>) that describes the nature of the content (that is, image, text, application, etc.)</p> <p>The X-Content-Type-Options header is an HTTP header that allows developers to specify that their content should not be MIME-sniffed. This header is designed to mitigate MIME-Sniffing attacks. Support for this header was added in Internet Explorer 8 (IE8)</p> <p>Only users of Internet Explorer 8 (IE8) will benefit from X-Content-Type-Options. Previous versions of Internet Explorer do not currently respect the X-Content-Type-Options header</p> <p>Internet Explorer 8 (and later) are the only major browsers to implement a MIME-sniffing opt-out feature. If and when other major browsers (Firefox, Safari, Chrome) implement similar features, this recommendation will be updated to include syntax for those browsers as well</p>

Example

To enable the required header globally for all pages in the application, you can do one of the following:

- Add the header in the web.config file if the application is hosted by Internet Information Services (IIS) 7

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name=""X-Content-Type-Options"" value=""nosniff""/>
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

- Add the header through the global Application_BeginRequest

```
void Application_BeginRequest(object sender, EventArgs e)
{
  this.Response.Headers["X-Content-Type-Options"] = "nosniff";
}
```

- Implement custom HTTP module

```

public class XContentTypeOptionsModule : IHttpModule
{
    #region IHttpModule Members
    public void Dispose()
    {

    }
    public void Init(HttpApplication context)
    {
        context.PreSendRequestHeaders += new EventHandler(context_PresendRequestHeaders);
    }
    #endregion
    void context_PresendRequestHeaders(object sender, EventArgs e)
    {
        HttpApplication application = sender as HttpApplication;
        if (application == null)
            return;
        if (application.Response.Headers["X-Content-Type-Options"] != null)
            return;
        application.Response.Headers.Add("X-Content-Type-Options", "nosniff");
    }
}

```

- You can enable the required header only for specific pages by adding it to individual responses:

```
this.Response.Headers["X-Content-Type-Options"] = "nosniff";
```

Harden or Disable XML Entity Resolution

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	XML Entity Expansion , XML Denial of Service Attacks and Defenses , MSXML Security Overview , Best Practices for Securing MSXML Code , NSXMLParserDelegate Protocol Reference , Resolving External References

TITLE	DETAILS
Steps	<p>Although it is not widely used, there is a feature of XML that allows the XML parser to expand macro entities with values defined either within the document itself or from external sources. For example, the document might define an entity "companyname" with the value "Microsoft," so that every time the text "&companyname;" appears in the document, it is automatically replaced with the text Microsoft. Or, the document might define an entity "MSFTStock" that references an external web service to fetch the current value of Microsoft stock.</p> <p>Then any time "&MSFTStock;" appears in the document, it is automatically replaced with the current stock price. However, this functionality can be abused to create denial of service (DoS) conditions. An attacker can nest multiple entities to create an exponential expansion XML bomb that consumes all available memory on the system.</p> <p>Alternatively, he can create an external reference that streams back an infinite amount of data or that simply hangs the thread. As a result, all teams must disable internal and/or external XML entity resolution entirely if their application does not use it, or manually limit the amount of memory and time that the application can consume for entity resolution if this functionality is absolutely necessary. If entity resolution is not required by your application, then disable it.</p>

Example

For .NET Framework code, you can use the following approaches:

```

XmlTextReader reader = new XmlTextReader(stream);
reader.ProhibitDtd = true;

XmlReaderSettings settings = new XmlReaderSettings();
settings.ProhibitDtd = true;
XmlReader reader = XmlReader.Create(stream, settings);

// for .NET 4
XmlReaderSettings settings = new XmlReaderSettings();
settings.DtdProcessing = DtdProcessing.Prohibit;
XmlReader reader = XmlReader.Create(stream, settings);

```

Note that the default value of `ProhibitDtd` in `XmlReaderSettings` is true, but in `XmlTextReader` it is false. If you are using `XmlReaderSettings`, you do not need to set `ProhibitDtd` to true explicitly, but it is recommended for safety sake that you do. Also note that the `XmlDocument` class allows entity resolution by default.

Example

To disable entity resolution for `XmlDocument`s, use the `XmlDocument.Load(XmlReader)` overload of the `Load` method and set the appropriate properties in the `XmlReader` argument to disable resolution, as illustrated in the following code:

```

XmlReaderSettings settings = new XmlReaderSettings();
settings.ProhibitDtd = true;
XmlReader reader = XmlReader.Create(stream, settings);
 XmlDocument doc = new XmlDocument();
doc.Load(reader);

```

Example

If disabling entity resolution is not possible for your application, set the `XmlReaderSettings.MaxCharactersFromEntities` property to a reasonable value according to your application's needs. This will limit the impact of potential exponential expansion DoS attacks. The following code provides an example of this approach:

```
XmlReaderSettings settings = new XmlReaderSettings();
settings.ProhibitDtd = false;
settings.MaxCharactersFromEntities = 1000;
XmlReader reader = XmlReader.Create(stream, settings);
```

Example

If you need to resolve inline entities but do not need to resolve external entities, set the `XmlReaderSettings.XmlResolver` property to null. For example:

```
XmlReaderSettings settings = new XmlReaderSettings();
settings.ProhibitDtd = false;
settings.MaxCharactersFromEntities = 1000;
settings.XmlResolver = null;
XmlReader reader = XmlReader.Create(stream, settings);
```

Note that in MSXML6, `ProhibitDTD` is set to true (disabling DTD processing) by default. For Apple OSX/iOS code, there are two XML parsers you can use: NSXMLParser and libXML2.

Applications utilizing http.sys perform URL canonicalization verification

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	<p>Any application that uses http.sys should follow these guidelines:</p> <ul style="list-style-type: none"> • Limit the URL length to no more than 16,384 characters (ASCII or Unicode). This is the absolute maximum URL length based on the default Internet Information Services (IIS) 6 setting. Websites should strive for a length shorter than this if possible • Use the standard .NET Framework file I/O classes (such as FileStream) as these will take advantage of the canonicalization rules in the .NET FX • Explicitly build an allow-list of known filenames • Explicitly reject known filetypes you will not serve UrlScan rejects: exe, bat, cmd, com, htw, ida, idq, htr, idc, shtm[], stm, printer, ini, pol, dat files • Catch the following exceptions: <ul style="list-style-type: none"> ◦ System.ArgumentException (for device names) ◦ System.NotSupportedException (for data streams) ◦ System.IO.FileNotFoundException (for invalid escaped filenames) ◦ System.IO.DirectoryNotFoundException (for invalid escaped dirs) • <i>Do not</i> call out to Win32 file I/O APIs. On an invalid URL gracefully return a 400 error to the user, and log the real error.

Ensure appropriate controls are in place when accepting files from users

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Unrestricted File Upload , File Signature Table

TITLE	DETAILS
Steps	<p>Uploaded files represent a significant risk to applications.</p> <p>The first step in many attacks is to get some code to the system to be attacked. Then the attack only needs to find a way to get the code executed. Using a file upload helps the attacker accomplish the first step. The consequences of unrestricted file upload can vary, including complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, and simple defacement.</p> <p>It depends on what the application does with the uploaded file and especially where it is stored. Server side validation of file uploads is missing. Following security controls should be implemented for File Upload functionality:</p> <ul style="list-style-type: none"> • File Extension check (only a valid set of allowed file type should be accepted) • Maximum file size limit • File should not be uploaded to webroot; the location should be a directory on non-system drive • Naming convention should be followed, such that the uploaded file name have some randomness, so as to prevent file overwrites • Files should be scanned for anti-virus before writing to the disk • Ensure that the file name and any other metadata (e.g., file path) are validated for malicious characters • File format signature should be checked, to prevent a user from uploading a masqueraded file (e.g., uploading an exe file by changing extension to txt)

Example

For the last point regarding file format signature validation, refer to the class below for details:

```

private static Dictionary<string, List<byte[]>> fileSignature = new Dictionary<string, List<byte[]>>
{
    { ".DOC", new List<byte[]> { new byte[] { 0xD0, 0xCF, 0x11, 0xE0, 0xA1, 0xB1, 0x1A, 0xE1 } } },
    { ".DOCX", new List<byte[]> { new byte[] { 0x50, 0x4B, 0x03, 0x04 } } },
    { ".PDF", new List<byte[]> { new byte[] { 0x25, 0x50, 0x44, 0x46 } } },
    { ".ZIP", new List<byte[]>
    {
        new byte[] { 0x50, 0x4B, 0x03, 0x04 },
        new byte[] { 0x50, 0x4B, 0x4C, 0x49, 0x54, 0x55 },
        new byte[] { 0x50, 0x4B, 0x53, 0x70, 0x58 },
        new byte[] { 0x50, 0x4B, 0x05, 0x06 },
        new byte[] { 0x50, 0x4B, 0x07, 0x08 },
        new byte[] { 0x57, 0x69, 0x6E, 0x5A, 0x69, 0x70 }
    } },
    { ".PNG", new List<byte[]> { new byte[] { 0x89, 0x50, 0x4E, 0x47, 0x0D, 0x0A, 0x1A, 0x0A } } },
    { ".JPG", new List<byte[]>
    {
        new byte[] { 0xFF, 0xD8, 0xFF, 0xE0 },
        new byte[] { 0xFF, 0xD8, 0xFF, 0xE1 },
        new byte[] { 0xFF, 0xD8, 0xFF, 0xE8 }
    } },
    { ".JPEG", new List<byte[]> { } }
}

```

```

        {
            new byte[] { 0xFF, 0xD8, 0xFF, 0xE0 },
            new byte[] { 0xFF, 0xD8, 0xFF, 0xE2 },
            new byte[] { 0xFF, 0xD8, 0xFF, 0xE3 }
        }
    },
    { ".XLS", new List<byte[]>
        {
            new byte[] { 0xD0, 0xCF, 0x11, 0xE0, 0xA1, 0xB1, 0x1A, 0xE1 },
            new byte[] { 0x09, 0x08, 0x10, 0x00, 0x00, 0x06, 0x05, 0x00 },
            new byte[] { 0xFD, 0xFF, 0xFF, 0xFF }
        }
    },
    { ".XLSX", new List<byte[]> { new byte[] { 0x50, 0x4B, 0x03, 0x04 } } },
    { ".GIF", new List<byte[]> { new byte[] { 0x47, 0x49, 0x46, 0x38 } } }
};

public static bool IsValidFileExtension(string fileName, byte[] fileData, byte[] allowedChars)
{
    if (string.IsNullOrEmpty(fileName) || fileData == null || fileData.Length == 0)
    {
        return false;
    }

    bool flag = false;
    string ext = Path.GetExtension(fileName);
    if (string.IsNullOrEmpty(ext))
    {
        return false;
    }

    ext = ext.ToUpperInvariant();

    if (ext.Equals(".TXT") || ext.Equals(".CSV") || ext.Equals(".PRN"))
    {
        foreach (byte b in fileData)
        {
            if (b > 0x7F)
            {
                if (allowedChars != null)
                {
                    if (!allowedChars.Contains(b))
                    {
                        return false;
                    }
                }
                else
                {
                    return false;
                }
            }
        }
    }

    return true;
}

if (!fileSignature.ContainsKey(ext))
{
    return true;
}

List<byte[]> sig = fileSignature[ext];
foreach (byte[] b in sig)
{
    var curFileSig = new byte[b.Length];
    Array.Copy(fileData, curFileSig, b.Length);
    if (curFileSig.SequenceEqual(b))
    {

```

```

        tag = true;
        break;
    }

    return flag;
}

```

Ensure that type-safe parameters are used in Web Application for data access

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>If you use the Parameters collection, SQL treats the input as a literal value rather than as executable code. The Parameters collection can be used to enforce type and length constraints on input data. Values outside of the range trigger an exception. If type-safe SQL parameters are not used, attackers might be able to execute injection attacks that are embedded in the unfiltered input.</p> <p>Use type safe parameters when constructing SQL queries to avoid possible SQL injection attacks that can occur with unfiltered input. You can use type safe parameters with stored procedures and with dynamic SQL statements. Parameters are treated as literal values by the database and not as executable code. Parameters are also checked for type and length.</p>

Example

The following code shows how to use type safe parameters with the SqlParameterCollection when calling a stored procedure.

```

using System.Data;
using System.Data.SqlClient;

using (SqlConnection connection = new SqlConnection(connectionString))
{
    DataSet userDataset = new DataSet();
    SqlDataAdapter myCommand = new SqlDataAdapter("LoginStoredProcedure", connection);
    myCommand.SelectCommand.CommandType = CommandType.StoredProcedure;
    myCommand.SelectCommand.Parameters.Add("@au_id", SqlDbType.VarChar, 11);
    myCommand.SelectCommand.Parameters["@au_id"].Value = SSN.Text;
    myCommand.Fill(userDataset);
}

```

In the preceding code example, the input value cannot be longer than 11 characters. If the data does not conform

to the type or length defined by the parameter, the SqlParameter class throws an exception.

Use separate model binding classes or binding filter lists to prevent MVC mass assignment vulnerability

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	MVC5, MVC6
Attributes	N/A
References	Metadata Attributes, Public Key Security Vulnerability And Mitigation, Complete Guide to Mass Assignment in ASP.NET MVC, Getting Started with EF using MVC
Steps	<ul style="list-style-type: none">When should I look for over-posting vulnerabilities? - Over-posting vulnerabilities can occur any place you bind model classes from user input. Frameworks like MVC can represent user data in custom .NET classes, including Plain Old CLR Objects (POCOs). MVC automatically populates these model classes with data from the request, providing a convenient representation for dealing with user input. When these classes include properties that should not be set by the user, the application can be vulnerable to over-posting attacks, which allow user control of data that the application never intended. Like MVC model binding, database access technologies such as object/relational mappers like Entity Framework often also support using POCO objects to represent database data. These data model classes provide the same convenience in dealing with database data as MVC does in dealing with user input. Because both MVC and the database support similar models, like POCO objects, it seems easy to reuse the same classes for both purposes. This practice fails to preserve separation of concerns, and it is one common area where unintended properties are exposed to model binding, enabling over-posting attacks.Why shouldn't I use my unfiltered database model classes as parameters to my MVC actions? - Because MVC model binding will bind anything in that class. Even if the data does not appear in your view, a malicious user can send an HTTP request with this data included, and MVC will gladly bind it because your action says that database class is the shape of data it should accept for user input.Why should I care about the shape used for model binding? - Using ASP.NET MVC model binding with overly broad models exposes an application to over-posting attacks. Over-posting could enable attackers to change application data beyond what the developer intended, such as overriding the price for an item or the security privileges for an account. Applications should use action-specific binding models (or specific allowed property filter lists) to provide an explicit

TITLE	DETAILS
	<p>contract for what untrusted input to allow via model binding.</p> <ul style="list-style-type: none"> Is having separate binding models just duplicating code? - No, it is a matter of separation of concerns. If you reuse database models in action methods, you are saying any property (or sub-property) in that class can be set by the user in an HTTP request. If that is not what you want MVC to do, you need a filter list or a separate class shape to show MVC what data can come from user input instead. If I have separate binding models for user input, do I have to duplicate all my data annotation attributes? - Not necessarily. You can use MetadataTypeAttribute on the database model class to link to the metadata on a model binding class. Just note that the type referenced by the MetadataTypeAttribute must be a subset of the referencing type (it can have fewer properties, but not more). Moving data back and forth between user input models and database models is tedious. Can I just copy over all properties using reflection? - Yes. The only properties that appear in the binding models are the ones you have determined to be safe for user input. There is no security reason that prevents using reflection to copy over all properties that exist in common between these two models. What about [Bind(Exclude = "")]. Can I use that instead of having separate binding models? - This approach is not recommended. Using [Bind(Exclude = "")] means that any new property is bindable by default. When a new property is added, there is an extra step to remember to keep things secure, rather than having the design be secure by default. Depending on the developer checking this list every time a property is added is risky. Is [Bind(Include = "")] useful for Edit operations? - No. [Bind(Include = "")] is only suitable for INSERT-style operations (adding new data). For UPDATE-style operations (revising existing data), use another approach, like having separate binding models or passing an explicit list of allowed properties to UpdateModel or TryUpdateModel. Adding a [Bind(Include = "")] attribute on an Edit operation means that MVC will create an object instance and set only the listed properties, leaving all others at their default values. When the data is persisted, it will entirely replace the existing entity, resetting the values for any omitted properties to their defaults. For example, if IsAdmin was omitted from a [Bind(Include = "")] attribute on an Edit operation, any user whose name was edited via this action would be reset to IsAdmin = false (any edited user would lose administrator status). If you want to prevent updates to certain properties, use one of the other approaches above. Note that some versions of MVC tooling generate controller classes with [Bind(Include = "")] on Edit actions and imply that removing a property from that list will prevent over-posting attacks. However, as described above, that approach does not work as intended and instead will reset any data in the omitted properties to their default values. For Create operations, are there any caveats using [Bind(Include = "")] rather than separate

TITLE	DETAILS
	<p>binding models? - Yes. First this approach does not work for Edit scenarios, requiring maintaining two separate approaches for mitigating all over-posting vulnerabilities. Second, separate binding models enforce separation of concerns between the shape used for user input and the shape used for persistence, something [Bind(Include = "Ã¢â€š]) does not do. Third, note that [Bind(Include = "Ã¢â€š]) can only handle top-level properties; you cannot allow only portions of sub-properties (such as "Details.Name") in the attribute. Finally, and perhaps most importantly, using [Bind(Include = "Ã¢â€š]) adds an extra step that must be remembered any time the class is used for model binding. If a new action method binds to the data class directly and forgets to include a [Bind(Include = "Ã¢â€š]) attribute, it can be vulnerable to over-posting attacks, so the [Bind(Include = "Ã¢â€š]) approach is somewhat less secure by default. If you use [Bind(Include = "Ã¢â€š]), take care always to remember to specify it every time your data classes appear as action method parameters.</p> <ul style="list-style-type: none"> • For Create operations, what about putting the [Bind(Include = "Ã¢â€š]) attribute on the model class itself? Does not this approach avoid the need to remember putting the attribute on every action method? - This approach works in some cases. Using [Bind(Include = "Ã¢â€š]) on the model type itself (rather than on action parameters using this class), does avoid the need to remember to include the [Bind(Include = "Ã¢â€š]) attribute on every action method. Using the attribute directly on the class effectively creates a separate surface area of this class for model binding purposes. However, this approach only allows for one model binding shape per model class. If one action method needs to allow model binding of a field (for example, an administrator-only action that updates user roles) and other actions need to prevent model binding of this field, this approach will not work. Each class can only have one model binding shape; if different actions need different model binding shapes, they need to represent these separate shapes using either separate model binding classes or separate [Bind(Include = "Ã¢â€š]) attributes on the action methods. • What are binding models? Are they the same thing as view models? - These are two related concepts. The term binding model refers to a model class used in an action's parameter list (the shape passed from MVC model binding to the action method). The term view model refers to a model class passed from an action method to a view. Using a view-specific model is a common approach for passing data from an action method to a view. Often, this shape is also suitable for model binding, and the term view model can be used to refer the same model used in both places. To be precise, this procedure talks specifically about binding models, focusing on the shape passed to the action, which is what matters for mass assignment purposes.

Encode untrusted web output prior to rendering

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic, Web Forms, MVC5, MVC6
Attributes	N/A
References	How to prevent Cross-site scripting in ASP.NET , Cross-site Scripting, XSS (Cross Site Scripting) Prevention Cheat Sheet
Steps	Cross-site scripting (commonly abbreviated as XSS) is an attack vector for online services or any application/component that consumes input from the web. XSS vulnerabilities may allow an attacker to execute script on another user's machine through a vulnerable web application. Malicious scripts can be used to steal cookies and otherwise tamper with a victim's machine through JavaScript. XSS is prevented by validating user input, ensuring it is well formed and encoding before it is rendered in a web page. Input validation and output encoding can be done by using Web Protection Library. For Managed code (C#, VB.net, etc.), use one or more appropriate encoding methods from the Web Protection (Anti-XSS) Library, depending on the context where the user input gets manifested:

Example

```
* Encoder.HtmlEncode
* Encoder.HtmlAttributeEncode
* Encoder.JavaScriptEncode
* Encoder.UrlEncode
* Encoder.VisualBasicScriptEncode
* Encoder.XmlEncode
* Encoder.XmlAttributeEncode
* Encoder.CssEncode
* Encoder.LdapEncode
```

Perform input validation and filtering on all string type Model properties

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic, MVC5, MVC6
Attributes	N/A

TITLE	DETAILS
References	<p>Adding Validation, Validating Model Data in an MVC Application, Guiding Principles For Your ASP.NET MVC Applications</p>
Steps	<p>All the input parameters must be validated before they are used in the application to ensure that the application is safeguarded against malicious user inputs. Validate the input values using regular expression validations on server side with a whitelist validation strategy. Unsanitized user inputs / parameters passed to the methods can cause code injection vulnerabilities.</p> <p>For web applications, entry points can also include form fields, QueryStrings, cookies, HTTP headers, and web service parameters.</p> <p>The following input validation checks must be performed upon model binding:</p> <ul style="list-style-type: none"> • The model properties should be annotated with RegularExpression annotation, for accepting allowed characters and maximum permissible length • The controller methods should perform ModelState validity

Sanitization should be applied on form fields that accept all characters, e.g, rich text editor

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	<p>Encode Unsafe Input, HTML Sanitizer</p>

TITLE	DETAILS
Steps	<p>Identify all static markup tags that you want to use. A common practice is to restrict formatting to safe HTML elements, such as <code></code> (bold) and <code><i></code> (italic).</p> <p>Before writing the data, HTML-encode it. This makes any malicious script safe by causing it to be handled as text, not as executable code.</p> <ol style="list-style-type: none"> 1. Disable ASP.NET request validation by adding the <code>ValidateRequest="false"</code> attribute to the @ Page directive 2. Encode the string input with the <code>HtmlEncode</code> method 3. Use a <code>StringBuilder</code> and call its <code>Replace</code> method to selectively remove the encoding on the HTML elements that you want to permit <p>The <code>Page</code> in the references disables ASP.NET request validation by setting <code>ValidateRequest="false"</code>. It HTML-encodes the input and selectively allows the <code></code> and <code><i></code>. Alternatively, a .NET library for HTML sanitization may also be used.</p> <p><code>HtmlSanitizer</code> is a .NET library for cleaning HTML fragments and documents from constructs that can lead to XSS attacks. It uses <code>AngleSharp</code> to parse, manipulate, and render HTML and CSS. <code>HtmlSanitizer</code> can be installed as a NuGet package, and the user input can be passed through relevant HTML or CSS sanitization methods, as applicable, on the server side. Please note that Sanitization as a security control should be considered only as a last option.</p> <p>Input validation and Output Encoding are considered better security controls.</p>

Do not assign DOM elements to sinks that do not have inbuilt encoding

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Many javascript functions don't do encoding by default. When assigning untrusted input to DOM elements via such functions, may result in cross site script (XSS) executions.

Example

Following are insecure examples:

```

document.getElementById("div1").innerHTML = value;
$("#userName").html(res.Name);
return $('<div/>').html(value)
$('body').append(resHTML);

```

Don't use `innerHTML`; instead use `innerText`. Similarly, instead of `$("#elm").html()`, use `$("#elm").text()`

Validate all redirects within the application are closed or done safely

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	The OAuth 2.0 Authorization Framework - Open Redirectors
Steps	<p>Application design requiring redirection to a user-supplied location must constrain the possible redirection targets to a predefined "safe" list of sites or domains. All redirects in the application must be closed/safe.</p> <p>To do this:</p> <ul style="list-style-type: none"> • Identify all redirects • Implement an appropriate mitigation for each redirect. Appropriate mitigations include redirect whitelist or user confirmation. If a web site or service with an open redirect vulnerability uses Facebook/OAuth/OpenID identity providers, an attacker can steal a user's logon token and impersonate that user. This is an inherent risk when using OAuth, which is documented in RFC 6749 "The OAuth 2.0 Authorization Framework", Section 10.15 "Open Redirects". Similarly, users' credentials can be compromised by spear phishing attacks using open redirects

Implement input validation on all string type parameters accepted by Controller methods

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic, MVC5, MVC6
Attributes	N/A

TITLE	DETAILS
References	Validating Model Data in an MVC Application, Guiding Principles For Your ASP.NET MVC Applications
Steps	For methods that just accept primitive data type, and not models as argument, input validation using Regular Expression should be done. Here Regex.IsMatch should be used with a valid regex pattern. If the input doesn't match the specified Regular Expression, control should not proceed further, and an adequate warning regarding validation failure should be displayed.

Set upper limit timeout for regular expression processing to prevent DoS due to bad regular expressions

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic, Web Forms, MVC5, MVC6
Attributes	N/A
References	DefaultRegexMatchTimeout Property
Steps	To ensure denial of service attacks against badly created regular expressions, that cause a lot of backtracking, set the global default timeout. If the processing time takes longer than the defined upper limit, it would throw a Timeout exception. If nothing is configured, the timeout would be infinite.

Example

For example, the following configuration will throw a RegexMatchTimeoutException, if the processing takes more than 5 seconds:

```
<httpRuntime targetFramework="4.5" defaultRegexMatchTimeout="00:00:05" />
```

Avoid using Html.Raw in Razor views

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	MVC5, MVC6
Attributes	N/A

TITLE	DETAILS
References	N/A
Step	ASP.Net WebPages (Razor) perform automatic HTML encoding. All strings printed by embedded code nuggets (@ blocks) are automatically HTML-encoded. However, when <code>HtmlHelper.Raw()</code> Method is invoked, it returns markup that is not HTML encoded. If <code>Html.Raw()</code> helper method is used, it bypasses the automatic encoding protection that Razor provides.

Example

Following is an insecure example:

```
<div class="form-group">
    @Html.Raw(Model.AccountConfirmText)
</div>
<div class="form-group">
    @Html.Raw(Model.PaymentConfirmText)
</div>
</div>
```

Do not use `Html.Raw()` unless you need to display markup. This method does not perform output encoding implicitly. Use other ASP.NET helpers e.g., `@Html.DisplayFor()`

Do not use dynamic queries in stored procedures

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	A SQL injection attack exploits vulnerabilities in input validation to run arbitrary commands in the database. It can occur when your application uses input to construct dynamic SQL statements to access the database. It can also occur if your code uses stored procedures that are passed strings that contain raw user input. Using the SQL injection attack, the attacker can execute arbitrary commands in the database. All SQL statements (including the SQL statements in stored procedures) must be parameterized. Parameterized SQL statements will accept characters that have special meaning to SQL (like single quote) without problems because they are strongly typed.

Example

Following is an example of insecure dynamic Stored Procedure:

```

CREATE PROCEDURE [dbo].[uspGetProductsByCriteria]
(
    @productName nvarchar(200) = NULL,
    @startPrice float = NULL,
    @endPrice float = NULL
)
AS
BEGIN
    DECLARE @sql nvarchar(max)
    SELECT @sql = ' SELECT ProductID, ProductName, Description, UnitPrice, ImagePath' +
        ' FROM dbo.Products WHERE 1 = 1 '
    PRINT @sql
    IF @productName IS NOT NULL
        SELECT @sql = @sql + ' AND ProductName LIKE ''%' + @productName + '%''''
    IF @startPrice IS NOT NULL
        SELECT @sql = @sql + ' AND UnitPrice > ''' + CONVERT(VARCHAR(10),@startPrice) + ''''
    IF @endPrice IS NOT NULL
        SELECT @sql = @sql + ' AND UnitPrice < ''' + CONVERT(VARCHAR(10),@endPrice) + ''''

    PRINT @sql
    EXEC(@sql)
END

```

Example

Following is the same stored procedure implemented securely:

```

CREATE PROCEDURE [dbo].[uspGetProductsByCriteriaSecure]
(
    @productName nvarchar(200) = NULL,
    @startPrice float = NULL,
    @endPrice float = NULL
)
AS
BEGIN
    SELECT ProductID, ProductName, Description, UnitPrice, ImagePath
    FROM dbo.Products where
    (@productName IS NULL or ProductName like '%' + @productName + '%')
    AND
    (@startPrice IS NULL or UnitPrice > @startPrice)
    AND
    (@endPrice IS NULL or UnitPrice < @endPrice)
END

```

Ensure that model validation is done on Web API methods

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	MVC5, MVC6
Attributes	N/A
References	Model Validation in ASP.NET Web API

TITLE	DETAILS
Steps	When a client sends data to a web API, it is mandatory to validate the data before doing any processing. For ASP.NET Web APIs which accept models as input, use data annotations on models to set validation rules on the properties of the model.

Example

The following code demonstrates the same:

```
using System.ComponentModel.DataAnnotations;

namespace MyApi.Models
{
    public class Product
    {
        public int Id { get; set; }
        [Required]
        [RegularExpression(@"^[\w-]*$", ErrorMessage="Only alphanumeric characters are allowed.")]
        public string Name { get; set; }
        public decimal Price { get; set; }
        [Range(0, 999)]
        public double Weight { get; set; }
    }
}
```

Example

In the action method of the API controllers, validity of the model has to be explicitly checked as shown below:

```
namespace MyApi.Controllers
{
    public class ProductsController : ApiController
    {
        public HttpResponseMessage Post(Product product)
        {
            if (ModelState.IsValid)
            {
                // Do something with the product (not shown).

                return new HttpResponseMessage(HttpStatusCode.OK);
            }
            else
            {
                return Request.CreateErrorResponse(HttpStatusCode.BadRequest, ModelState);
            }
        }
    }
}
```

Implement input validation on all string type parameters accepted by Web API methods

TITLE	DETAILS
Component	Web API

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic, MVC 5, MVC 6
Attributes	N/A
References	Validating Model Data in an MVC Application, Guiding Principles For Your ASP.NET MVC Applications
Steps	For methods that just accept primitive data type, and not models as argument, input validation using Regular Expression should be done. Here Regex.IsMatch should be used with a valid regex pattern. If the input doesn't match the specified Regular Expression, control should not proceed further, and an adequate warning regarding validation failure should be displayed.

Ensure that type-safe parameters are used in Web API for data access

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	<p>If you use the Parameters collection, SQL treats the input as a literal value rather than as executable code. The Parameters collection can be used to enforce type and length constraints on input data. Values outside of the range trigger an exception. If type-safe SQL parameters are not used, attackers might be able to execute injection attacks that are embedded in the unfiltered input.</p> <p>Use type safe parameters when constructing SQL queries to avoid possible SQL injection attacks that can occur with unfiltered input. You can use type safe parameters with stored procedures and with dynamic SQL statements. Parameters are treated as literal values by the database and not as executable code. Parameters are also checked for type and length.</p>

Example

The following code shows how to use type safe parameters with the SqlParameterCollection when calling a stored procedure.

```

using System.Data;
using System.Data.SqlClient;

using (SqlConnection connection = new SqlConnection(connectionString))
{
    DataSet userDataset = new DataSet();
    SqlDataAdapter myCommand = new SqlDataAdapter("LoginStoredProcedure", connection);
    myCommand.SelectCommand.CommandType = CommandType.StoredProcedure;
    myCommand.SelectCommand.Parameters.Add("@au_id", SqlDbType.VarChar, 11);
    myCommand.SelectCommand.Parameters["@au_id"].Value = SSN.Text;
    myCommand.Fill(userDataset);
}

```

In the preceding code example, the input value cannot be longer than 11 characters. If the data does not conform to the type or length defined by the parameter, the SqlParameter class throws an exception.

Use parametrized SQL queries for Cosmos DB

TITLE	DETAILS
Component	Azure Document DB
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Announcing SQL Parameterization in Azure Cosmos DB
Steps	Although Azure Cosmos DB only supports read-only queries, SQL injection is still possible if queries are constructed by concatenating with user input. It might be possible for a user to gain access to data they shouldn't be accessing within the same collection by crafting malicious SQL queries. Use parameterized SQL queries if queries are constructed based on user input.

WCF Input validation through Schema binding

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	N/A
References	MSDN

TITLE	DETAILS
Steps	<p>Lack of validation leads to different type injection attacks.</p> <p>Message validation represents one line of defense in the protection of your WCF application. With this approach, you validate messages using schemas to protect WCF service operations from attack by a malicious client. Validate all messages received by the client to protect the client from attack by a malicious service. Message validation makes it possible to validate messages when operations consume message contracts or data contracts, which cannot be done using parameter validation. Message validation allows you to create validation logic inside schemas, thereby providing more flexibility and reducing development time. Schemas can be reused across different applications inside the organization, creating standards for data representation. Additionally, message validation allows you to protect operations when they consume more complex data types involving contracts representing business logic.</p> <p>To perform message validation, you first build a schema that represents the operations of your service and the data types consumed by those operations. You then create a .NET class that implements a custom client message inspector and custom dispatcher message inspector to validate the messages sent/received to/from the service. Next, you implement a custom endpoint behavior to enable message validation on both the client and the service. Finally, you implement a custom configuration element on the class that allows you to expose the extended custom endpoint behavior in the configuration file of the service or the client"</p>

WCF- Input validation through Parameter Inspectors

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	N/A
References	MSDN

TITLE	DETAILS
Steps	<p>Input and data validation represents one important line of defense in the protection of your WCF application. You should validate all parameters exposed in WCF service operations to protect the service from attack by a malicious client. Conversely, you should also validate all return values received by the client to protect the client from attack by a malicious service.</p> <p>WCF provides different extensibility points that allow you to customize the WCF runtime behavior by creating custom extensions. Message Inspectors and Parameter Inspectors are two extensibility mechanisms used to gain greater control over the data passing between a client and a service. You should use parameter inspectors for input validation and use message inspectors only when you need to inspect the entire message flowing in and out of a service.</p> <p>To perform input validation, you will build a .NET class and implement a custom parameter inspector in order to validate parameters on operations in your service. You will then implement a custom endpoint behavior to enable validation on both the client and the service. Finally, you will implement a custom configuration element on the class that allows you to expose the extended custom endpoint behavior in the configuration file of the service or the client.</p>

Security Frame: Sensitive Data | Mitigations

12/6/2018 • 16 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Machine Trust Boundary	<ul style="list-style-type: none">• Ensure that binaries are obfuscated if they contain sensitive information• Consider using Encrypted File System (EFS) is used to protect confidential user-specific data• Ensure that sensitive data stored by the application on the file system is encrypted
Web Application	<ul style="list-style-type: none">• Ensure that sensitive content is not cached on the browser• Encrypt sections of Web App's configuration files that contain sensitive data• Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs• Ensure that sensitive data displayed on the user screen is masked
Database	<ul style="list-style-type: none">• Implement dynamic data masking to limit sensitive data exposure non privileged users• Ensure that passwords are stored in salted hash format• Ensure that sensitive data in database columns is encrypted• Ensure that database-level encryption (TDE) is enabled• Ensure that database backups are encrypted
Web API	<ul style="list-style-type: none">• Ensure that sensitive data relevant to Web API is not stored in browser's storage
Azure Document DB	<ul style="list-style-type: none">• Encrypt sensitive data stored in Azure Cosmos DB
Azure IaaS VM Trust Boundary	<ul style="list-style-type: none">• Use Azure Disk Encryption to encrypt disks used by Virtual Machines
Service Fabric Trust Boundary	<ul style="list-style-type: none">• Encrypt secrets in Service Fabric applications
Dynamics CRM	<ul style="list-style-type: none">• Perform security modeling and use Business Units/Teams where required• Minimize access to share feature on critical entities• Train users on the risks associated with the Dynamics CRM Share feature and good security practices• Include a development standards rule proscribing showing config details in exception management

PRODUCT/SERVICE	ARTICLE
Azure Storage	<ul style="list-style-type: none"> • Use Azure Storage Service Encryption (SSE) for Data at Rest (Preview) • Use Client-Side Encryption to store sensitive data in Azure Storage
Mobile Client	<ul style="list-style-type: none"> • Encrypt sensitive or PII data written to phones local storage • Obfuscate generated binaries before distributing to end users
WCF	<ul style="list-style-type: none"> • Set clientCredentialType to Certificate or Windows • WCF-Security Mode is not enabled

Ensure that binaries are obfuscated if they contain sensitive information

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that binaries are obfuscated if they contain sensitive information such as trade secrets, sensitive business logic that should not be reversed. This is to stop reverse engineering of assemblies. Tools like CryptoObfuscator may be used for this purpose.

Consider using Encrypted File System (EFS) is used to protect confidential user-specific data

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	Consider using Encrypted File System (EFS) is used to protect confidential user-specific data from adversaries with physical access to the computer.

Ensure that sensitive data stored by the application on the file system is encrypted

TITLE	DETAILS
Component	Machine Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Ensure that sensitive data stored by the application on the file system is encrypted (e.g., using DPAPI), if EFS cannot be enforced

Ensure that sensitive content is not cached on the browser

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic, Web Forms, MVC5, MVC6
Attributes	N/A
References	N/A
Steps	Browsers can store information for purposes of caching and history. These cached files are stored in a folder, like the Temporary Internet Files folder in the case of Internet Explorer. When these pages are referred again, the browser displays them from its cache. If sensitive information is displayed to the user (such as their address, credit card details, Social Security Number, or username), then this information could be stored in browser's cache, and therefore retrievable through examining the browser's cache or by simply pressing the browser's "Back" button. Set cache-control response header value to "no-store" for all pages.

Example

```

<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <add name="Cache-Control" value="no-cache" />
        <add name="Pragma" value="no-cache" />
        <add name="Expires" value="-1" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>

```

Example

This may be implemented through a filter. Following example may be used:

```

public override void OnActionExecuting(ActionExecutingContext filterContext)
{
    if (filterContext == null || (filterContext.HttpContext != null &&
filterContext.HttpContext.Response != null && filterContext.HttpContext.Response.IsRequestBeingRedirected))
    {
        //// Since this is MVC pipeline, this should never be null.
        return;
    }

    var attributes =
filterContext.ActionDescriptor.GetCustomAttributes(typeof(System.Web.Mvc.OutputCacheAttribute), false);
    if (attributes == null || **Attributes**.Count() == 0)
    {
        filterContext.HttpContext.Response.Cache.SetNoStore();
        filterContext.HttpContext.Response.Cache.SetCacheability(HttpCacheability.NoCache);
        filterContext.HttpContext.Response.Cache.SetExpires(DateTime.UtcNow.AddHours(-1));
        if (!filterContext.IsChildAction)
        {
            filterContext.HttpContext.Response.AppendHeader("Pragma", "no-cache");
        }
    }

    base.OnActionExecuting(filterContext);
}

```

Encrypt sections of Web App's configuration files that contain sensitive data

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	How To: Encrypt Configuration Sections in ASP.NET 2.0 Using DPAPI, Specifying a Protected Configuration Provider, Using Azure Key Vault to protect application secrets

TITLE	DETAILS
Steps	Configuration files such as the Web.config, appsettings.json are often used to hold sensitive information, including user names, passwords, database connection strings, and encryption keys. If you do not protect this information, your application is vulnerable to attackers or malicious users obtaining sensitive information such as account user names and passwords, database names and server names. Based on the deployment type (azure/on-prem), encrypt the sensitive sections of config files using DPAPI or services like Azure Key Vault.

Explicitly disable the autocomplete HTML attribute in sensitive forms and inputs

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	MSDN: autocomplete attribute, Using AutoComplete in HTML, HTML Sanitization Vulnerability, Autocomplete.,again?!
Steps	The autocomplete attribute specifies whether a form should have autocomplete on or off. When autocomplete is on, the browser automatically complete values based on values that the user has entered before. For example, when a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved. Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the clear text password from the browser cache. By default autocomplete is enabled, and it must explicitly be disabled.

Example

```
<form action="Login.aspx" method="post" autocomplete="off">
    Social Security Number: <input type="text" name="ssn" />
    <input type="submit" value="Submit" />
</form>
```

Ensure that sensitive data displayed on the user screen is masked

TITLE	DETAILS
Component	Web Application

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Sensitive data such as passwords, credit card numbers, SSN etc. should be masked when displayed on the screen. This is to prevent unauthorized personnel from accessing the data (e.g., shoulder-surfing passwords, support personnel viewing SSN numbers of users) . Ensure that these data elements are not visible in plain text and are appropriately masked. This has to be taken care while accepting them as input (e.g., input type="password") as well as displaying back on the screen (e.g., display only the last 4 digits of the credit card number).

Implement dynamic data masking to limit sensitive data exposure non privileged users

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Sql Azure, OnPrem
Attributes	SQL Version - V12, SQL Version - MsSQL2016
References	Dynamic Data Masking
Steps	The purpose of dynamic data masking is to limit exposure of sensitive data, preventing users who should not have access to the data from viewing it. Dynamic data masking does not aim to prevent database users from connecting directly to the database and running exhaustive queries that expose pieces of the sensitive data. Dynamic data masking is complementary to other SQL Server security features (auditing, encryption, row level security...) and it is highly recommended to use this feature in conjunction with them in addition in order to better protect the sensitive data in the database. Please note that this feature is supported only by SQL Server starting with 2016 and Azure SQL Database.

Ensure that passwords are stored in salted hash format

TITLE	DETAILS
Component	Database

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Password Hashing using .NET Crypto APIs
Steps	Passwords should not be stored in custom user store databases. Password hashes should be stored with salt values instead. Make sure the salt for the user is always unique and you apply b-crypt, s-crypt or PBKDF2 before storing the password, with a minimum work factor iteration count of 150,000 loops to eliminate the possibility of brute forcing.

Ensure that sensitive data in database columns is encrypted

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	SQL Version - All
References	Encrypting sensitive data in SQL server , How to: Encrypt a Column of Data in SQL Server , Encrypt by Certificate
Steps	Sensitive data such as credit card numbers has to be encrypted in the database. Data can be encrypted using column-level encryption or by an application function using the encryption functions.

Ensure that database-level encryption (TDE) is enabled

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Understanding SQL Server Transparent Data Encryption (TDE)

TITLE	DETAILS
Steps	Transparent Data Encryption (TDE) feature in SQL server helps in encrypting sensitive data in a database and protect the keys that are used to encrypt the data with a certificate. This prevents anyone without the keys from using the data. TDE protects data "at rest", meaning the data and log files. It provides the ability to comply with many laws, regulations, and guidelines established in various industries.

Ensure that database backups are encrypted

TITLE	DETAILS
Component	Database
SDL Phase	Build
Applicable Technologies	SQL Azure, OnPrem
Attributes	SQL Version - V12, SQL Version - MsSQL2014
References	SQL database backup encryption
Steps	SQL Server has the ability to encrypt the data while creating a backup. By specifying the encryption algorithm and the encryptor (a Certificate or Asymmetric Key) when creating a backup, one can create an encrypted backup file.

Ensure that sensitive data relevant to Web API is not stored in browser's storage

TITLE	DETAILS
Component	Web API
SDL Phase	Build
Applicable Technologies	MVC 5, MVC 6
Attributes	Identity Provider - ADFS, Identity Provider - Azure AD
References	N/A

TITLE	DETAILS
Steps	<p>In certain implementations, sensitive artifacts relevant to Web API's authentication are stored in browser's local storage. E.g., Azure AD authentication artifacts like adal.idtoken, adal.nonce.idtoken, adal.access.token.key, adal.token.keys, adal.state.login, adal.session.state, adal.expiration.key etc.</p> <p>All these artifacts are available even after sign out or browser is closed. If an adversary gets access to these artifacts, he/she can reuse them to access the protected resources (APIs). Ensure that all sensitive artifacts related to Web API is not stored in browser's storage. In cases where client-side storage is unavoidable (e.g., Single Page Applications (SPA) that leverage Implicit OpenIdConnect/OAuth flows need to store access tokens locally), use storage choices with do not have persistence. e.g., prefer SessionStorage to LocalStorage.</p>

Example

The below JavaScript snippet is from a custom authentication library which stores authentication artifacts in local storage. Such implementations should be avoided.

```
ns.AuthHelper.Authenticate = function () {
  window.config = {
    instance: 'https://login.microsoftonline.com/',
    tenant: ns.Configurations.Tenant,
    clientId: ns.Configurations.AADApplicationClientID,
    postLogoutRedirectUri: window.location.origin,
    cacheLocation: 'localStorage', // enable this for IE, as sessionStorage does not work for localhost.
  };
}
```

Encrypt sensitive data stored in Cosmos DB

TITLE	DETAILS
Component	Azure Document DB
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Encrypt sensitive data at application level before storing in document DB or store any sensitive data in other storage solutions like Azure Storage or Azure SQL

Use Azure Disk Encryption to encrypt disks used by Virtual Machines

TITLE	DETAILS
Component	Azure IaaS VM Trust Boundary
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	Using Azure Disk Encryption to encrypt disks used by your virtual machines
Steps	<p>Azure Disk Encryption is a new feature that is currently in preview. This feature allows you to encrypt the OS disks and Data disks used by an IaaS Virtual Machine. For Windows, the drives are encrypted using industry-standard BitLocker encryption technology. For Linux, the disks are encrypted using the DM-Crypt technology. This is integrated with Azure Key Vault to allow you to control and manage the disk encryption keys. The Azure Disk Encryption solution supports the following three customer encryption scenarios:</p> <ul style="list-style-type: none"> • Enable encryption on new IaaS VMs created from customer-encrypted VHD files and customer-provided encryption keys, which are stored in Azure Key Vault. • Enable encryption on new IaaS VMs created from the Azure Marketplace. • Enable encryption on existing IaaS VMs already running in Azure.

Encrypt secrets in Service Fabric applications

TITLE	DETAILS
Component	Service Fabric Trust Boundary
SDL Phase	Build
Applicable Technologies	Generic
Attributes	Environment - Azure
References	Managing secrets in Service Fabric applications
Steps	<p>Secrets can be any sensitive information, such as storage connection strings, passwords, or other values that should not be handled in plain text. Use Azure Key Vault to manage keys and secrets in service fabric applications.</p>

Perform security modeling and use Business Units/Teams where required

TITLE	DETAILS
Component	Dynamics CRM
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Perform security modeling and use Business Units/Teams where required

Minimize access to share feature on critical entities

TITLE	DETAILS
Component	Dynamics CRM
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Minimize access to share feature on critical entities

Train users on the risks associated with the Dynamics CRM Share feature and good security practices

TITLE	DETAILS
Component	Dynamics CRM
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Train users on the risks associated with the Dynamics CRM Share feature and good security practices

Include a development standards rule proscribing showing config

details in exception management

TITLE	DETAILS
Component	Dynamics CRM
SDL Phase	Deployment
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Include a development standards rule proscribing showing config details in exception management outside development. Test for this as part of code reviews or periodic inspection.

Use Azure Storage Service Encryption (SSE) for Data at Rest (Preview)

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	StorageType - Blob
References	Azure Storage Service Encryption for Data at Rest (Preview)

TITLE	DETAILS
Steps	<p>Azure Storage Service Encryption (SSE) for Data at Rest helps you protect and safeguard your data to meet your organizational security and compliance commitments. With this feature, Azure Storage automatically encrypts your data prior to persisting to storage and decrypts prior to retrieval. The encryption, decryption and key management is totally transparent to users. SSE applies only to block blobs, page blobs, and append blobs. The other types of data, including tables, queues, and files, will not be encrypted.</p> <p>Encryption and Decryption Workflow:</p> <ul style="list-style-type: none"> • The customer enables encryption on the storage account • When the customer writes new data (PUT Blob, PUT Block, PUT Page, etc.) to Blob storage; every write is encrypted using 256-bit AES encryption, one of the strongest block ciphers available • When the customer needs to access data (GET Blob, etc.), data is automatically decrypted before returning to the user • If encryption is disabled, new writes are no longer encrypted and existing encrypted data remains encrypted until rewritten by the user. While encryption is enabled, writes to Blob storage will be encrypted. The state of data does not change with the user toggling between enabling/disabling encryption for the storage account • All encryption keys are stored, encrypted, and managed by Microsoft <p>Please note that at this time, the keys used for the encryption are managed by Microsoft. Microsoft generates the keys originally, and manage the secure storage of the keys as well as the regular rotation as defined by internal Microsoft policy. In the future, customers will get the ability to manage their own encryption keys, and provide a migration path from Microsoft-managed keys to customer-managed keys.</p>

Use Client-Side Encryption to store sensitive data in Azure Storage

TITLE	DETAILS
Component	Azure Storage
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A

TITLE	DETAILS
References	<p>Client-Side Encryption and Azure Key Vault for Microsoft Azure Storage, Tutorial: Encrypt and decrypt blobs in Microsoft Azure Storage using Azure Key Vault, Storing Data Securely in Azure Blob Storage with Azure Encryption Extensions</p>
Steps	<p>The Azure Storage Client Library for .NET Nuget package supports encrypting data within client applications before uploading to Azure Storage, and decrypting data while downloading to the client. The library also supports integration with Azure Key Vault for storage account key management. Here is a brief description of how client side encryption works:</p> <ul style="list-style-type: none"> • The Azure Storage client SDK generates a content encryption key (CEK), which is a one-time-use symmetric key • Customer data is encrypted using this CEK • The CEK is then wrapped (encrypted) using the key encryption key (KEK). The KEK is identified by a key identifier and can be an asymmetric key pair or a symmetric key and can be managed locally or stored in Azure Key Vault. The Storage client itself never has access to the KEK. It just invokes the key wrapping algorithm that is provided by Key Vault. Customers can choose to use custom providers for key wrapping/unwrapping if they want • The encrypted data is then uploaded to the Azure Storage service. Check the links in the references section for low-level implementation details.

Encrypt sensitive or PII data written to phones local storage

TITLE	DETAILS
Component	Mobile Client
SDL Phase	Build
Applicable Technologies	Generic, Xamarin
Attributes	N/A
References	<p>Manage settings and features on your devices with Microsoft Intune policies, Keychain Valet</p>
Steps	<p>If the application writes sensitive information like user's PII (email, phone number, first name, last name, preferences etc.)- on mobile's file system, then it should be encrypted before writing to the local file system. If the application is an enterprise application, then explore the possibility of publishing application using Windows Intune.</p>

Example

Intune can be configured with following security policies to safeguard sensitive data:

```
Require encryption on mobile device  
Require encryption on storage cards  
Allow screen capture
```

Example

If the application is not an enterprise application, then use platform provided keystore, keychains to store encryption keys, using which cryptographic operation may be performed on the file system. Following code snippet shows how to access key from keychain using xamarin:

```
protected static string EncryptionKey
{
    get
    {
        if (String.IsNullOrEmpty(_Key))
        {
            var query = new SecRecord(SecKind.GenericPassword);
            query.Service = NSBundle.MainBundle.BundleIdentifier;
            query.Account = "UniqueID";

            NSData uniqueId = SecKeyChain.QueryAsData(query);
            if (uniqueId == null)
            {
                query.ValueData = NSData.FromString(System.Guid.NewGuid().ToString());
                var err = SecKeyChain.Add(query);
                _Key = query.ValueData.ToString();
            }
            else
            {
                _Key = uniqueId.ToString();
            }
        }

        return _Key;
    }
}
```

Obfuscate generated binaries before distributing to end users

TITLE	DETAILS
Component	Mobile Client
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Crypto Obfuscation For .Net
Steps	Generated binaries (assemblies within apk) should be obfuscated to stop reverse engineering of assemblies. Tools like CryptoObfuscator may be used for this purpose.

Set clientCredentialType to Certificate or Windows

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	.NET Framework 3
Attributes	N/A
References	Fortify
Steps	Using a UsernameToken with a plaintext password over an unencrypted channel exposes the password to attackers who can sniff the SOAP messages. Service Providers that use the UsernameToken might accept passwords sent in plaintext. Sending plaintext passwords over an unencrypted channel can expose the credential to attackers who can sniff the SOAP message.

Example

The following WCF service provider configuration uses the UsernameToken:

```
<security mode="Message">
<message clientCredentialType="UserName" />
```

Set clientCredentialType to Certificate or Windows.

WCF-Security Mode is not enabled

TITLE	DETAILS
Component	WCF
SDL Phase	Build
Applicable Technologies	Generic, .NET Framework 3
Attributes	Security Mode - Transport, Security Mode - Message
References	MSDN , Fortify Kingdom , Fundamentals of WCF Security CoDe Magazine
Steps	No transport or message security has been defined. Applications that transmit messages without transport or message security cannot guarantee the integrity or confidentiality of the messages. When a WCF security binding is set to None, both transport and message security are disabled.

Example

The following configuration sets the security mode to None.

```
<system.serviceModel>
  <bindings>
    <wsHttpBinding>
      <binding name=""MyBinding"">
        <security mode=""None""/>
      </binding>
    </bindings>
  </system.serviceModel>
```

Example

Security Mode Across all service bindings there are five possible security modes:

- None. Turns security off.
- Transport. Uses transport security for mutual authentication and message protection.
- Message. Uses message security for mutual authentication and message protection.
- Both. Allows you to supply settings for transport and message-level security (only MSMQ supports this).
- TransportWithMessageCredential. Credentials are passed with the message and message protection and server authentication are provided by the transport layer.
- TransportCredentialOnly. Client credentials are passed with the transport layer and no message protection is applied. Use transport and message security to protect the integrity and confidentiality of messages. The configuration below tells the service to use transport security with message credentials.

```
<system.serviceModel> <bindings> <wsHttpBinding> <binding name=""MyBinding""> <security
mode=""TransportWithMessageCredential""/> <message clientCredentialType=""Windows""/> </binding> </bindings>
</system.serviceModel>
```

Security Frame: Session Management

11/7/2018 • 14 minutes to read • [Edit Online](#)

PRODUCT/SERVICE	ARTICLE
Azure AD	<ul style="list-style-type: none">• Implement proper logout using ADAL methods when using Azure AD
IoT Device	<ul style="list-style-type: none">• Use finite lifetimes for generated SaaS tokens
Azure Document DB	<ul style="list-style-type: none">• Use minimum token lifetimes for generated Resource tokens
ADFS	<ul style="list-style-type: none">• Implement proper logout using WsFederation methods when using ADFS
Identity Server	<ul style="list-style-type: none">• Implement proper logout when using Identity Server
Web Application	<ul style="list-style-type: none">• Applications available over HTTPS must use secure cookies• All http based application should specify http only for cookie definition• Mitigate against Cross-Site Request Forgery (CSRF) attacks on ASP.NET web pages• Set up session for inactivity lifetime• Implement proper logout from the application
Web API	<ul style="list-style-type: none">• Mitigate against Cross-Site Request Forgery (CSRF) attacks on ASP.NET Web APIs

Implement proper logout using ADAL methods when using Azure AD

TITLE	DETAILS
Component	Azure AD
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A

TITLE	DETAILS
Steps	If the application relies on access token issued by Azure AD, the logout event handler should call

Example

```
HttpContext.GetOwinContext().Authentication.SignOut(OpenIdConnectAuthenticationDefaults.AuthenticationType,
CookieAuthenticationDefaults.AuthenticationType)
```

Example

It should also destroy user's session by calling Session.Abandon() method. Following method shows secure implementation of user logout:

```
[HttpPost]
[ValidateAntiForgeryToken]
public void LogOff()
{
    string userObjectID =
ClaimsPrincipal.Current.FindFirst("http://schemas.microsoft.com/identity/claims/objectidentifier").Value;
    AuthenticationContext authContext = new AuthenticationContext(Authority + TenantId, new
NaiveSessionCache(userObjectID));
    authContext.TokenCache.Clear();
    Session.Clear();
    Session.Abandon();
    Response.SetCookie(new HttpCookie("ASP.NET_SessionId", string.Empty));
    HttpContext.GetOwinContext().Authentication.SignOut(
        OpenIdConnectAuthenticationDefaults.AuthenticationType,
        CookieAuthenticationDefaults.AuthenticationType);
}
```

Use finite lifetimes for generated SaS tokens

TITLE	DETAILS
Component	IoT Device
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	SaS tokens generated for authenticating to Azure IoT Hub should have a finite expiry period. Keep the SaS token lifetimes to a minimum to limit the amount of time they can be replayed in case the tokens are compromised.

Use minimum token lifetimes for generated Resource tokens

TITLE	DETAILS
Component	Azure Document DB
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Reduce the timespan of resource token to a minimum value required. Resource tokens have a default valid timespan of 1 hour.

Implement proper logout using WsFederation methods when using ADFS

TITLE	DETAILS
Component	ADFS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	If the application relies on STS token issued by ADFS, the logout event handler should call <code>WSFederationAuthenticationModule.FederatedSignOut()</code> method to log out the user. Also the current session should be destroyed, and the session token value should be reset and nullified.

Example

```

[HttpPost, ValidateAntiForgeryToken]
[Authorization]
public ActionResult SignOut(string redirectUrl)
{
    if (!this.User.Identity.IsAuthenticated)
    {
        return this.View("LogOff", null);
    }

    // Removes the user profile.
    this.Session.Clear();
    this.Session.Abandon();
    HttpContext.Current.Response.Cookies.Add(new System.Web.HttpCookie("ASP.NET_SessionId",
string.Empty)
    {
        Expires = DateTime.Now.AddDays(-1D),
        Secure = true,
        HttpOnly = true
    });
}

// Signs out at the specified security token service (STS) by using the WS-Federation protocol.
Uri signOutUrl = new Uri(FederatedAuthentication.WSFederationAuthenticationModule.Issuer);
Uri replyUrl = new Uri(FederatedAuthentication.WSFederationAuthenticationModule.Realm);
if (!string.IsNullOrEmpty(redirectUrl))
{
    replyUrl = new Uri(FederatedAuthentication.WSFederationAuthenticationModule.Realm +
redirectUrl);
}
// Signs out of the current session and raises the appropriate events.
var authModule = FederatedAuthentication.WSFederationAuthenticationModule;
authModule.SignOut(false);
// Signs out at the specified security token service (STS) by using the WS-Federation
// protocol.
WSFederationAuthenticationModule.FederatedSignOut(signOutUrl, replyUrl);
return new RedirectResult(redirectUrl);
}

```

Implement proper logout when using Identity Server

TITLE	DETAILS
Component	Identity Server
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	IdentityServer3-Federated sign out
Steps	IdentityServer supports the ability to federate with external identity providers. When a user signs out of an upstream identity provider, depending upon the protocol used, it might be possible to receive a notification when the user signs out. It allows IdentityServer to notify its clients so they can also sign the user out. Check the documentation in the references section for the implementation details.

Applications available over HTTPS must use secure cookies

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	EnvironmentType - OnPrem
References	httpCookies Element (ASP.NET Settings Schema) , HttpCookie.Secure Property
Steps	Cookies are normally only accessible to the domain for which they were scoped. Unfortunately, the definition of "domain" does not include the protocol so cookies that are created over HTTPS are accessible over HTTP. The "secure" attribute indicates to the browser that the cookie should only be made available over HTTPS. Ensure that all cookies set over HTTPS use the secure attribute. The requirement can be enforced in the web.config file by setting the requireSSL attribute to true. It is the preferred approach because it will enforce the secure attribute for all current and future cookies without the need to make any additional code changes.

Example

```
<configuration>
  <system.web>
    <httpCookies requireSSL="true"/>
  </system.web>
</configuration>
```

The setting is enforced even if HTTP is used to access the application. If HTTP is used to access the application, the setting breaks the application because the cookies are set with the secure attribute and the browser will not send them back to the application.

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Web Forms, MVC5
Attributes	EnvironmentType - OnPrem
References	N/A
Steps	When the web application is the Relying Party, and the IdP is ADFS server, the FedAuth token's secure attribute can be configured by setting requireSSL to True in <code>system.identityModel.services</code> section of web.config:

Example

```
<system.identityModel.services>
  <federationConfiguration>
    <!-- Set requireSsl=true; domain=application domain name used by FedAuth cookies (Ex: .gdinfra.com); -->
    <cookieHandler requireSsl="true" persistentSessionLifetime="0.0:20:0" />
    ...
  </federationConfiguration>
</system.identityModel.services>
```

All http based application should specify http only for cookie definition

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	Secure Cookie Attribute
Steps	To mitigate the risk of information disclosure with a cross-site scripting (XSS) attack, a new attribute - httpOnly - was introduced to cookies and is supported by all major browsers. The attribute specifies that a cookie is not accessible through script. By using HttpOnly cookies, a web application reduces the possibility that sensitive information contained in the cookie can be stolen via script and sent to an attacker's website.

Example

All HTTP-based applications that use cookies should specify HttpOnly in the cookie definition, by implementing following configuration in web.config:

```
<system.web>
  .
  .
  <httpCookies requireSSL="false" httpOnlyCookies="true"/>
  .
  .
</system.web>
```

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Web Forms
Attributes	N/A

TITLE	DETAILS
References	FormsAuthentication.RequireSSL Property
Steps	The RequireSSL property value is set in the configuration file for an ASP.NET application by using the requireSSL attribute of the configuration element. You can specify in the Web.config file for your ASP.NET application whether SSL (Secure Sockets Layer) is required to return the forms-authentication cookie to the server by setting the requireSSL attribute.

Example

The following code example sets the requireSSL attribute in the Web.config file.

```
<authentication mode="Forms">
  <forms loginUrl="member_login.aspx" cookieless="UseCookies" requireSSL="true"/>
</authentication>
```

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	MVC5
Attributes	EnvironmentType - OnPrem
References	Windows Identity Foundation (WIF) Configuration – Part II
Steps	To set httpOnly attribute for FedAuth cookies, hideFromCscript attribute value should be set to True.

Example

Following configuration shows the correct configuration:

```
<federatedAuthentication>
  <cookieHandler mode="Custom">
    hideFromScript="true"
    name="FedAuth"
    path="/"
    requireSsl="true"
    persistentSessionLifetime="25">
  </cookieHandler>
</federatedAuthentication>
```

Mitigate against Cross-Site Request Forgery (CSRF) attacks on ASP.NET web pages

TITLE	DETAILS
Component	Web Application

TITLE	DETAILS
SDL Phase	Build
Applicable Technologies	Generic
Attributes	N/A
References	N/A
Steps	Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker can carry out actions in the security context of a different user's established session on a web site. The goal is to modify or delete content, if the targeted web site relies exclusively on session cookies to authenticate received request. An attacker could exploit this vulnerability by getting a different user's browser to load a URL with a command from a vulnerable site on which the user is already logged in. There are many ways for an attacker to do that, such as by hosting a different web site that loads a resource from the vulnerable server, or getting the user to click a link. The attack can be prevented if the server sends an additional token to the client, requires the client to include that token in all future requests, and verifies that all future requests include a token that pertains to the current session, such as by using the ASP.NET AntiForgeryToken or ViewState.

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	MVC5, MVC6
Attributes	N/A
References	XSRF/CSRF Prevention in ASP.NET MVC and Web Pages
Steps	Anti-CSRF and ASP.NET MVC forms - Use the <code>AntiForgeryToken()</code> helper method on Views; put an <code>Html.AntiForgeryToken()</code> into the form, for example,

Example

```
@using (Html.BeginForm("UserProfile", "SubmitUpdate")) {
    @Html.ValidationSummary(true)
    @Html.AntiForgeryToken()
    <fieldset>
```

Example

```

<form action="/UserProfile/SubmitUpdate" method="post">
    <input name="__RequestVerificationToken" type="hidden"
value="saTFWpkKN0BYazFtN6c4YbZAmsEwG0srqlUqqloifVgeV2ciIFVmelvzwRZpArs" />
    <!-- rest of form goes here -->
</form>

```

Example

At the same time, `Html.AntiForgeryToken()` gives the visitor a cookie called `__RequestVerificationToken`, with the same value as the random hidden value shown above. Next, to validate an incoming form post, add the `[ValidateAntiForgeryToken]` filter to the target action method. For example:

```

[ValidateAntiForgeryToken]
public ViewResult SubmitUpdate()
{
// ... etc.
}

```

Authorization filter that checks that:

- The incoming request has a cookie called `__RequestVerificationToken`
- The incoming request has a `Request.Form` entry called `__RequestVerificationToken`
- These cookie and `Request.Form` values match Assuming all is well, the request goes through as normal. But if not, then an authorization failure with message "A required anti-forgery token was not supplied or was invalid".

Example

Anti-CSRF and AJAX: The form token can be a problem for AJAX requests, because an AJAX request might send JSON data, not HTML form data. One solution is to send the tokens in a custom HTTP header. The following code uses Razor syntax to generate the tokens, and then adds the tokens to an AJAX request.

```

<script>
@functions{
    public string TokenHeaderValue()
    {
        string cookieToken, formToken;
        AntiForgery.GetTokens(null, out cookieToken, out formToken);
        return cookieToken + ":" + formToken;
    }
}

$.ajax("api/values", {
    type: "post",
    contentType: "application/json",
    data: { }, // JSON data goes here
    dataType: "json",
    headers: {
        'RequestVerificationToken': '@TokenHeaderValue()'
    }
});
</script>

```

Example

When you process the request, extract the tokens from the request header. Then call the `AntiForgery.Validate` method to validate the tokens. The `Validate` method throws an exception if the tokens are not valid.

```

void ValidateRequestHeader(HttpRequestMessage request)
{
    string cookieToken = "";
    string formToken = "";

    IEnumerable<string> tokenHeaders;
    if (request.Headers.TryGetValues("RequestVerificationToken", out tokenHeaders))
    {
        string[] tokens = tokenHeaders.First().Split(':');
        if (tokens.Length == 2)
        {
            cookieToken = tokens[0].Trim();
            formToken = tokens[1].Trim();
        }
    }
    AntiForgery.Validate(cookieToken, formToken);
}

```

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Web Forms
Attributes	N/A
References	Take Advantage of ASP.NET Built-in Features to Fend Off Web Attacks
Steps	CSRF attacks in WebForm based applications can be mitigated by setting ViewStateUserKey to a random string that varies for each user - user ID or, better yet, session ID. For a number of technical and social reasons, session ID is a much better fit because a session ID is unpredictable, times out, and varies on a per-user basis.

Example

Here's the code you need to have in all of your pages:

```

void Page_Init (object sender, EventArgs e) {
    ViewStateUserKey = Session.SessionID;
    :
}

```

Set up session for inactivity lifetime

TITLE	DETAILS
Component	Web Application
SDL Phase	Build
Applicable Technologies	Generic

TITLE	DETAILS
Attributes	N/A
References	HttpSessionState.Timeout Property
Steps	Session timeout represents the event occurring when a user does not perform any action on a web site during a interval (defined by web server). The event, on server side, change the status of the user session to 'invalid' (for example "not used anymore") and instruct the web server to destroy it (deleting all data contained into it). The following code example sets the timeout session attribute to 15 minutes in the Web.config file.

Example

```
<configuration>
  <system.web>
    <sessionState mode="InProc" cookieless="true" timeout="15" />
  </system.web>
</configuration>
```

Enable Threat detection on Azure SQL

Title	Details
Component	Web Application
SDL Phase	Build
Applicable Technologies	Web Forms
Attributes	N/A
References	[forms Element for authentication (ASP.NET Settings Schema)] (https://msdn.microsoft.com/library/1d3t3c61(v=vs.100).aspx)
Steps	Set the Forms Authentication Ticket cookie timeout to 15 minutes

```
### Example
```XML
<forms name=".ASPxAUTH" loginUrl="login.aspx" defaultUrl="default.aspx" protection="All" timeout="15"
path="/" requireSSL="true" slidingExpiration="true"/>
</forms>
```

TITLE	DETAILS
<b>Component</b>	Web Application
<b>SDL Phase</b>	Build
<b>Applicable Technologies</b>	Web Forms, MVC5
<b>Attributes</b>	EnvironmentType - OnPrem
<b>References</b>	<a href="#">asdeqa</a>

TITLE	DETAILS
<b>Steps</b>	When the web application is Relying Party and ADFS is the STS, the lifetime of the authentication cookies - FedAuth tokens - can be set by the following configuration in web.config:

## Example

```
<system.identityModel.services>
 <federationConfiguration>
 <!-- Set requireSsl=true; domain=application domain name used by FedAuth cookies (Ex: .gdinfra.com); -->
 <cookieHandler requireSsl="true" persistentSessionLifetime="0.0:15:0" />
 <!-- Set requireHttps=true; -->
 <wsFederation passiveRedirectEnabled="true" issuer="http://localhost:39529/" realm="https://localhost:44302/" reply="https://localhost:44302/" requireHttps="true"/>
 <!--
 Use the code below to enable encryption-decryption of claims received from ADFS. Thumbprint value varies based on the certificate being used.
 <serviceCertificate>
 <certificateReference findValue="4FBBA33A1D11A9022A5BF3492FF83320007686A"
 storeLocation="LocalMachine" storeName="My" x509FindType="FindByThumbprint" />
 </serviceCertificate>
 -->
 </federationConfiguration>
</system.identityModel.services>
```

## Example

Also the ADFS issued SAML claim token's lifetime should be set to 15 minutes, by executing the following powershell command on the ADFS server:

```
Set-ADFSRelyingPartyTrust -TargetName "<RelyingPartyWebApp>" -ClaimsProviderName @("Active Directory") - TokenLifetime 15 -AlwaysRequireAuthentication $true
```

## Implement proper logout from the application

TITLE	DETAILS
<b>Component</b>	Web Application
<b>SDL Phase</b>	Build
<b>Applicable Technologies</b>	Generic
<b>Attributes</b>	N/A
<b>References</b>	N/A
<b>Steps</b>	Perform proper Sign Out from the application, when user presses log out button. Upon logout, application should destroy user's session, and also reset and nullify session cookie value, along with resetting and nullifying authentication cookie value. Also, when multiple sessions are tied to a single user identity, they must be collectively terminated on the server side at timeout or logout. Lastly, ensure that Logout functionality is available on every page.

# Mitigate against Cross-Site Request Forgery (CSRF) attacks on ASP.NET Web APIs

TITLE	DETAILS
<b>Component</b>	Web API
<b>SDL Phase</b>	Build
<b>Applicable Technologies</b>	Generic
<b>Attributes</b>	N/A
<b>References</b>	N/A
<b>Steps</b>	Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker can carry out actions in the security context of a different user's established session on a web site. The goal is to modify or delete content, if the targeted web site relies exclusively on session cookies to authenticate received request. An attacker could exploit this vulnerability by getting a different user's browser to load a URL with a command from a vulnerable site on which the user is already logged in. There are many ways for an attacker to do that, such as by hosting a different web site that loads a resource from the vulnerable server, or getting the user to click a link. The attack can be prevented if the server sends an additional token to the client, requires the client to include that token in all future requests, and verifies that all future requests include a token that pertains to the current session, such as by using the ASP.NET AntiForgeryToken or ViewState.

TITLE	DETAILS
<b>Component</b>	Web API
<b>SDL Phase</b>	Build
<b>Applicable Technologies</b>	MVC5, MVC6
<b>Attributes</b>	N/A
<b>References</b>	<a href="#">Preventing Cross-Site Request Forgery (CSRF) Attacks in ASP.NET Web API</a>
<b>Steps</b>	Anti-CSRF and AJAX: The form token can be a problem for AJAX requests, because an AJAX request might send JSON data, not HTML form data. One solution is to send the tokens in a custom HTTP header. The following code uses Razor syntax to generate the tokens, and then adds the tokens to an AJAX request.

## Example

```

<script>
 @functions{
 public string TokenHeaderValue()
 {
 string cookieToken, formToken;
 AntiForgeryToken.GetTokens(null, out cookieToken, out formToken);
 return cookieToken + ":" + formToken;
 }
 }
 $.ajax("api/values", {
 type: "post",
 contentType: "application/json",
 data: { },
 dataType: "json",
 headers: {
 'RequestVerificationToken': '@TokenHeaderValue()'
 }
 });
</script>

```

## Example

When you process the request, extract the tokens from the request header. Then call the `AntiForgery.Validate` method to validate the tokens. The `Validate` method throws an exception if the tokens are not valid.

```

void ValidateRequestHeader(HttpRequestMessage request)
{
 string cookieToken = "";
 string formToken = "";

 IEnumerable<string> tokenHeaders;
 if (request.Headers.TryGetValues("RequestVerificationToken", out tokenHeaders))
 {
 string[] tokens = tokenHeaders.First().Split(':');
 if (tokens.Length == 2)
 {
 cookieToken = tokens[0].Trim();
 formToken = tokens[1].Trim();
 }
 }
 AntiForgery.Validate(cookieToken, formToken);
}

```

## Example

Anti-CSRF and ASP.NET MVC forms - Use the `AntiForgeryToken` helper method on Views; put an `Html.AntiForgeryToken()` into the form, for example,

```

@using (Html.BeginForm("UserProfile", "SubmitUpdate")) {
 @Html.ValidationSummary(true)
 @Html.AntiForgeryToken()
 <fieldset>
}

```

## Example

The example above will output something like the following:

```

<form action="/UserProfile/SubmitUpdate" method="post">
 <input name="__RequestVerificationToken" type="hidden"
value="saTFWpkKN0BYazFtN6c4YbZAmsEwG0srqlUqqloifVgeV2ciIFVmelvezwRZpArs" />
 <!-- rest of form goes here -->
</form>

```

## Example

At the same time, `Html.AntiForgeryToken()` gives the visitor a cookie called `__RequestVerificationToken`, with the same value as the random hidden value shown above. Next, to validate an incoming form post, add the `[ValidateAntiForgeryToken]` filter to the target action method. For example:

```

[ValidateAntiForgeryToken]
public ViewResult SubmitUpdate()
{
// ... etc.
}

```

Authorization filter that checks that:

- The incoming request has a cookie called `__RequestVerificationToken`
- The incoming request has a `Request.Form` entry called `__RequestVerificationToken`
- These cookie and `Request.Form` values match Assuming all is well, the request goes through as normal. But if not, then an authorization failure with message "A required anti-forgery token was not supplied or was invalid".

TITLE	DETAILS
<b>Component</b>	Web API
<b>SDL Phase</b>	Build
<b>Applicable Technologies</b>	MVC5, MVC6
<b>Attributes</b>	Identity Provider - ADFS, Identity Provider - Azure AD
<b>References</b>	<a href="#">Secure a Web API with Individual Accounts and Local Login in ASP.NET Web API 2.2</a>
<b>Steps</b>	If the Web API is secured using OAuth 2.0, then it expects a bearer token in Authorization request header and grants access to the request only if the token is valid. Unlike cookie based authentication, browsers do not attach the bearer tokens to requests. The requesting client needs to explicitly attach the bearer token in the request header. Therefore, for ASP.NET Web APIs protected using OAuth 2.0, bearer tokens are considered as a defense against CSRF attacks. Please note that if the MVC portion of the application uses forms authentication (i.e., uses cookies), anti-forgery tokens have to be used by the MVC web app.

## Example

The Web API has to be informed to rely ONLY on bearer tokens and not on cookies. It can be done by the following configuration in `WebApiConfig.Register` method:

```
config.SuppressDefaultHostAuthentication();
config.Filters.Add(new HostAuthenticationFilter(OAuthDefaults.AuthenticationType));
```

The SuppressDefaultHostAuthentication method tells Web API to ignore any authentication that happens before the request reaches the Web API pipeline, either by IIS or by OWIN middleware. That way, we can restrict Web API to authenticate only using bearer tokens.