Table of Contents

Azure security overview

11/22/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil

The Microsoft Azure Security Information site on Azure.com is your place for technical answers to your Azure security questions. If you are interested in Azure compliance and how Azure secures its own infrastructure, visit the Microsoft Trust Center.

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 the wide array of security tools and capabilities available. These tools and capabilities help make it possible to create secure solutions on the secure Azure platform.

This page, and this site, will evolve as new Azure services come online and existing services are updated. Our security offerings constantly improve and we'll share the technical details of those improvements here. We recommend that you return to this site regularly and see what's new!

Using the Azure Security Information Site

When you look on the left side of the page, you see a collection of categories of security topics. Click or tap the topic to see the articles we have for that category.

Each category has an Overview article. The goal of the Overview article is to provide you a summary of what Azure has to offer in terms of security for that category. The Overview articles help you understand the products, services, and features available in Azure that you can use to enhance the security of your deployments. These Overview articles contain numerous links to technical content so that you can learn more about the topics and how to implement them.

The Azure Security Information site currently addresses the following categories:

- Resources
- Network security
- Storage security
- Virtual machine security
- Endpoint protection
- Security management and monitoring
- Identity management
- PaaS services
- Industry specific
- Architecture

> **NOTE**
> All the articles we have in the Azure Security Information site appear in the navigation on the left side of the page. If you want to see all the articles, click or tap **See More** at the bottom of the categories list.

We want your input! If there are categories you want added, let us know! If there are articles you want added to an

existing category, let us know that too! You can use the Comments section below, or write to us at azsecinfosite@microsoft.com and we'll incorporate your recommendations into our plans.

Getting started with Microsoft Azure security

11/15/2016 • 16 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Ralph Squillace • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • ShawnJackson

---

When you build 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.

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.

In this overview article on Azure security, we'll look at:

- Azure services and features you can use to help secure your services and data within Azure.
- How Microsoft secures the Azure infrastructure to help protect your data and applications.

Identity and access management

Controlling access to IT infrastructure, data, and applications is critical. Microsoft Azure delivers these capabilities by services such as Azure Active Directory (Azure AD), Azure Storage, and support for numerous standards and APIs.

Azure AD is an identity repository and engine that provides authentication, authorization, and access control for an organization's users, groups, and objects. Azure AD also offers developers an effective way to integrate identity management in their applications. Industry-standard protocols such as SAML 2.0, WS-Federation, and OpenID Connect make sign-in possible on platforms such as .NET, Java, Node.js, and PHP.

The REST-based Graph API enables developers to read and write to the directory from any platform. Through support for OAuth 2.0, developers can build mobile and web applications that integrate with Microsoft and third-party web APIs, and build their own secure web APIs. Open-source client libraries are available for .Net, Windows Store, iOS, and Android, with additional libraries under development.

How Azure enables identity and access management

Azure AD can be used as a standalone cloud directory for your organization or as an integrated solution with your existing on-premises Active Directory. Some integration features include directory sync and single sign-on (SSO). These extend the reach of your existing on-premises identities into the cloud and improve the admin and user experience.

Some other capabilities for identity and access management include:

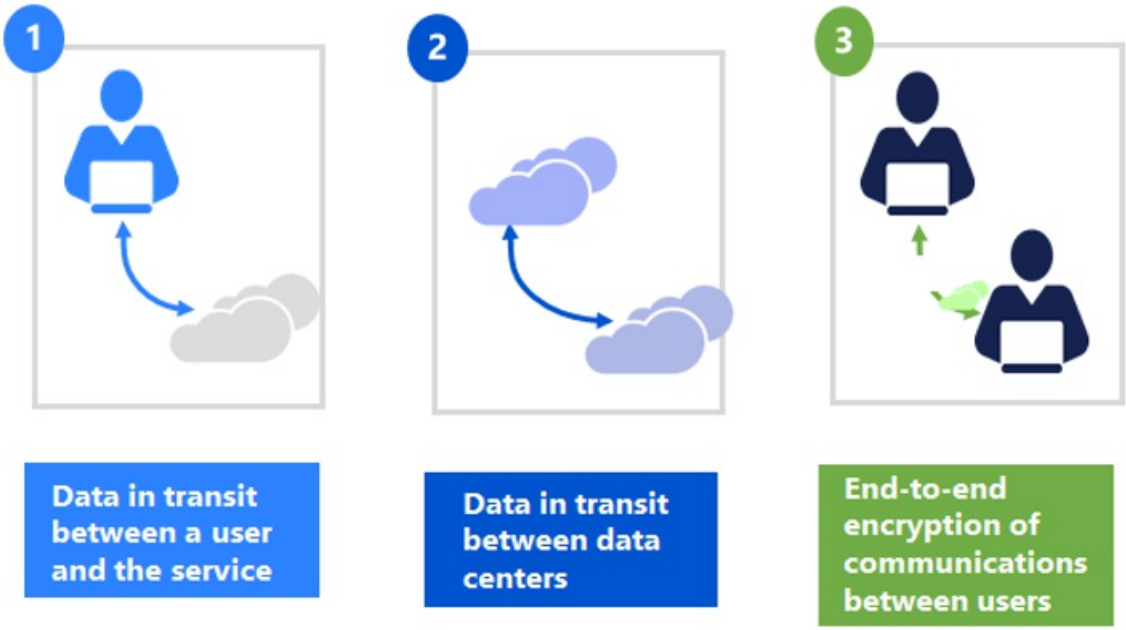
- Azure AD enables SSO to SaaS applications, regardless of where they are hosted. Some applications are federated with Azure AD, and others use password SSO. Federated applications can also support user provisioning and password vaulting.
- Access to data in Azure Storage is controlled via authentication. 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 a 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 new credentials.

Data access control and encryption

Microsoft employs the principles of Separation of Duties and Least Privilege throughout Azure operations. Access to data by Azure support personnel requires your explicit permission and is granted on a "just-in-time" basis that is logged and audited, then revoked after completion of the engagement.

Azure also provides multiple capabilities for protecting data in transit and at rest. This includes encryption for data, files, applications, services, communications, and drives. You can encrypt information before placing it in Azure, and also store keys in your on-premises datacenters.



Azure encryption technologies

You can gather details on administrative access to your subscription environment by using Azure AD Reporting. You can configure BitLocker Drive Encryption on VHDs containing sensitive information in Azure.

Other capabilities in Azure that will assist you to keep your data secure include:

- Application developers can build encryption into the applications they deploy in Azure by using the Windows CryptoAPI and .NET Framework.
- Completely control the keys with client-side encryption for Azure Blob storage. The storage service never sees the keys and is incapable of decrypting the data.
- Azure Rights Management (Azure RMS) (with the RMS SDK) provides file and data-level encryption and data-leak prevention through policy-based access management.
- Azure supports table-level and column-level encryption (TDE/CLE) in SQL Server virtual machines, and it

supports third-party on-premises key management servers in datacenters.

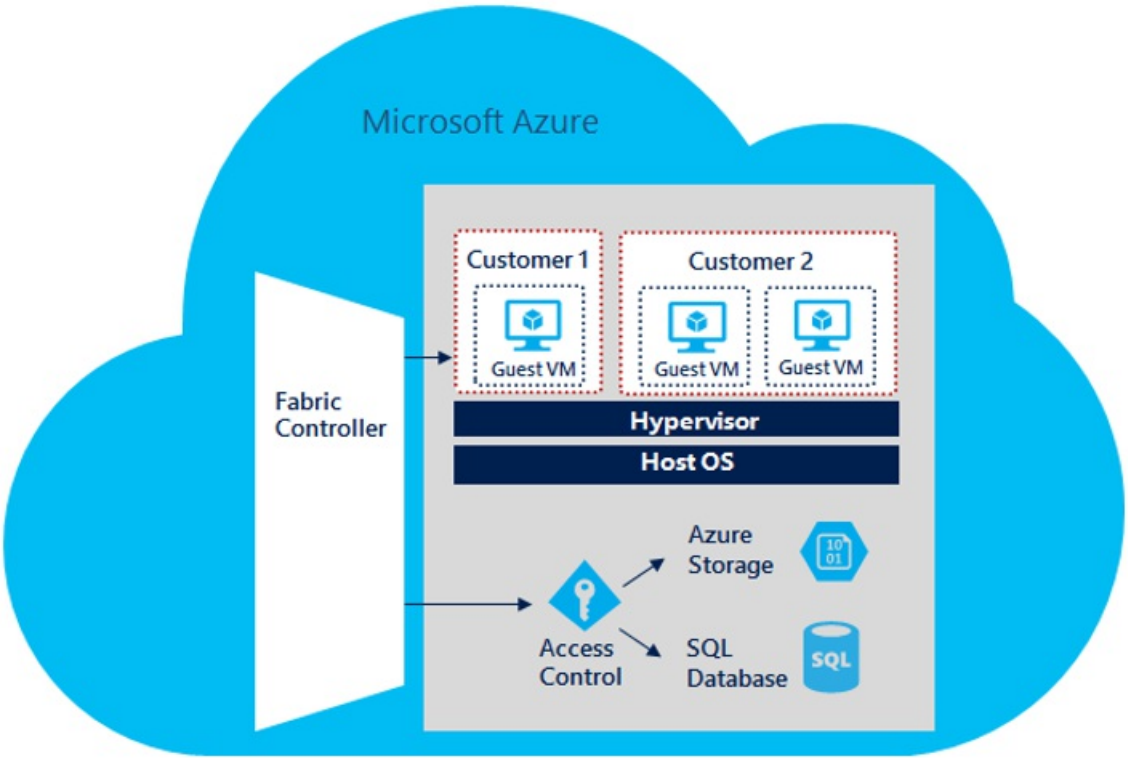
- Storage Account Keys, Shared Access Signatures, management certificates, and other keys are unique to each Azure tenant.
- Azure StorSimple hybrid storage encrypts data via a 128-bit public/private key pair before uploading it to Azure Storage.
- Azure supports and uses numerous encryption mechanisms, including SSL/TLS, IPsec, and AES, depending on the data types, containers, and transports.

Virtualization

The Azure platform uses a virtualized environment. User instances operate as standalone virtual machines that do not have access to a physical host server, and this isolation is enforced by using physical processor (ring-0/ring-3) privilege levels.

Ring 0 is the most privileged and 3 is the least. The guest OS runs in a lesser-privileged Ring 1, and applications run in the least privileged Ring 3. This virtualization of physical resources leads to a clear separation between guest OS and hypervisor, resulting in additional security separation between the two.

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.



How Azure implements virtualization

Azure uses a hypervisor firewall (packet filter) that is implemented in the hypervisor and configured by a fabric controller agent. This helps protect tenants from unauthorized access. 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 here:

- **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 cluster and the OS activation

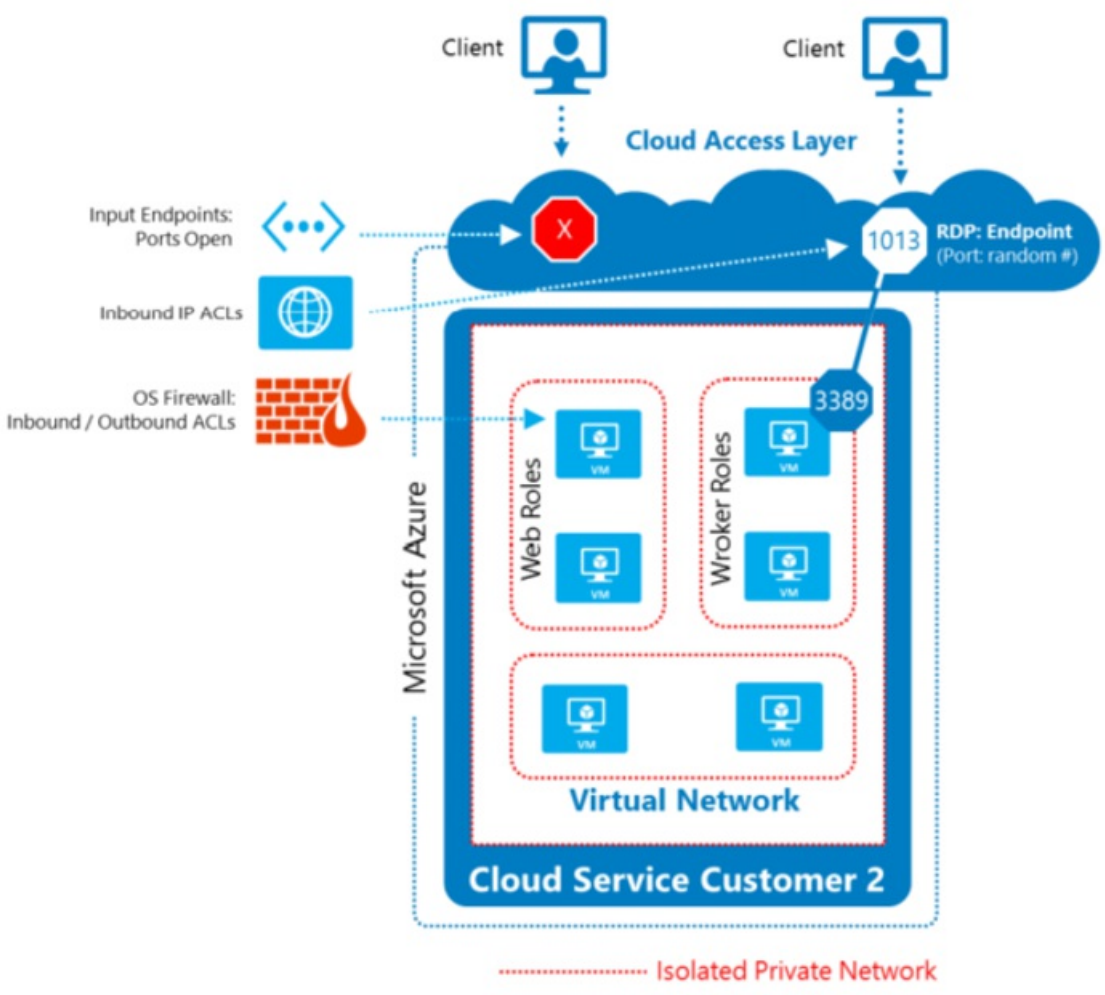
server. The virtual machines' list of allowed outgoing destinations does not include Azure router subnets, Azure management back end, and other Microsoft properties.

- **Role configuration file**: This defines the inbound Access Control Lists (ACLs) based on the tenant's service model. For example, if a tenant has a Web front end on port 80 on a certain virtual machine, then Azure opens TCP port 80 to all IPs if you're configuring an endpoint in the Azure classic deployment model. If the virtual machine has a back end or worker role running, then it opens the worker role only to the virtual machine within the same tenant.

Isolation

Another important cloud security requirement is separation to prevent unauthorized and unintentional transfer of information between deployments in a shared multi-tenant architecture.

Azure implements network access control and segregation through VLAN isolation, ACLs, load balancers, and IP filters. It restricts external traffic inbound to ports and protocols on your virtual machines that you define. Azure implements network filtering to prevent spoofed traffic and restrict incoming and outgoing traffic to trusted platform components. Traffic flow policies are implemented on boundary protection devices that deny traffic by default.



Network Address Translation (NAT) is used to separate internal network traffic from external traffic. Internal traffic is not externally routable. Virtual IP addresses that are externally routable are translated into internal Dynamic IP addresses that are only routable within Azure.

External traffic to Azure virtual machines is firewalled via ACLs on routers, load balancers, and Layer 3 switches. Only specific known protocols are permitted. ACLs are in place to limit traffic originating from guest virtual machines to other VLANs used for management. In addition, traffic filtered via IP filters on the host OS further limits the traffic on both data link and network layers.

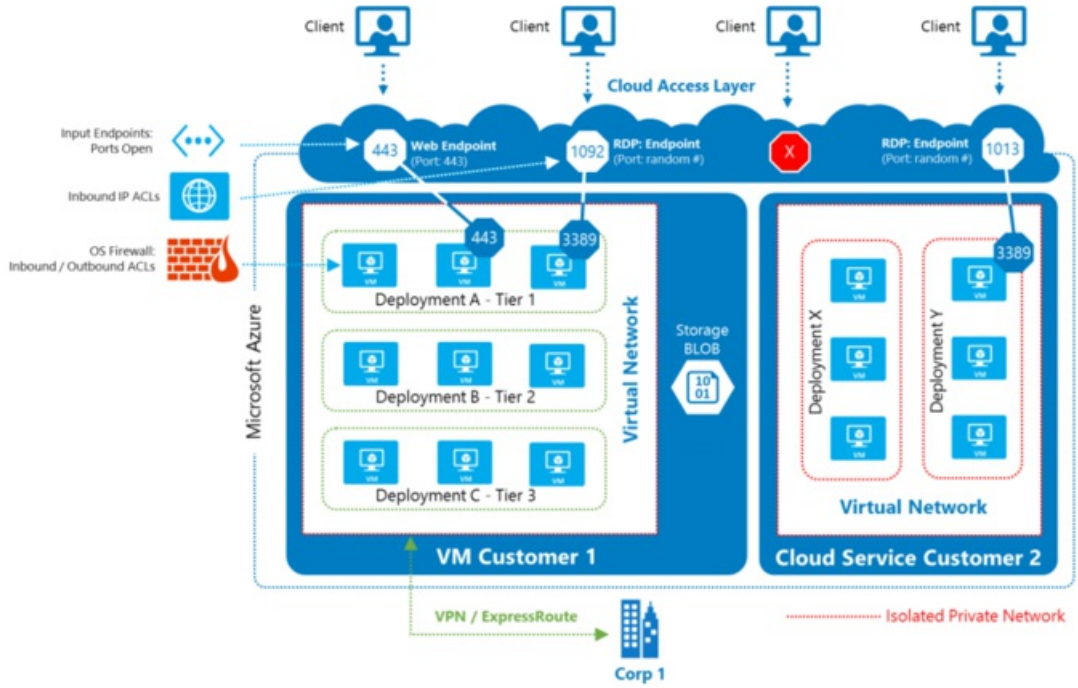
How Azure implements isolation

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 Azure hypervisor enforces memory and process separation between virtual machines, and it securely routes network traffic to guest OS tenants. Azure also implements isolation for tenants, storage, and virtual networks.

- Each Azure AD tenant is logically isolated by using security boundaries.
- Azure storage accounts are unique to each subscription, and access must be authenticated by using a storage account key.
- Virtual networks are logically isolated through a combination of unique private IP addresses, firewalls, and IP ACLs. Load balancers route traffic to the appropriate tenants based on endpoint definitions.

Virtual networks and firewalls

The distributed and virtual networks in Azure help ensure that your private network traffic is logically isolated from traffic on other Azure virtual networks.



Your subscription can contain multiple isolated private networks (and include firewall, load balancing, and network address translation).

Azure provides three primary levels of network segregation in each Azure cluster to logically segregate traffic. Virtual local area networks (VLANs) are used to separate customer traffic from the rest of the Azure network. Access to the Azure network from outside the cluster is restricted through load balancers.

Network traffic to and from virtual machines must pass through the hypervisor virtual switch. The IP filter component in the root OS isolates the root virtual machine from the guest virtual machines and the guest virtual machines from one another. It performs filtering of traffic to restrict communication between a tenant's nodes and the public Internet (based on the customer's service configuration), segregating them from other tenants.

The IP filter helps prevent guest virtual machines from:

- Generating spoofed traffic.
- Receiving traffic not addressed to them.
- Directing traffic to protected infrastructure endpoints.
- Sending or receiving inappropriate broadcast traffic.

You can place your virtual machines onto Azure virtual networks. These virtual networks are similar to the networks you configure in on-premises environments, where they are typically associated with a virtual switch. Virtual machines connected to the same virtual network can communicate with one another without additional configuration. You can also configure different subnets within your virtual network.

You can use the following Azure Virtual Network technologies to help secure communications on your virtual network:

- **Network Security Groups (NSGs)**. You can use an NSG to control traffic to one or more virtual machine instances 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.
- **User-defined routing**. You can control the routing of packets through a virtual appliance by creating user-defined routes that specify the next hop for packets flowing to a specific subnet to go to a virtual network security appliance.
- **IP forwarding**. A virtual network security appliance must be able to receive incoming traffic that is not addressed to itself. To allow a virtual machine to receive traffic addressed to other destinations, you enable IP forwarding for the virtual machine.
- **Forced tunneling**. Forced tunneling lets you redirect or "force" all Internet-bound traffic generated by your virtual machines in a virtual network back to your on-premises location via a site-to-site VPN tunnel for inspection and auditing
- **Endpoint ACLs**. You can control which machines are allowed inbound connections from the Internet to a virtual machine on your virtual network by defining endpoint ACLs.
- **Partner network security solutions**. There are a number of partner network security solutions that you can access from the Azure Marketplace.

How Azure implements virtual networks and firewalls

Azure implements packet-filtering firewalls on all host and guest virtual machines by default. Windows OS images from the Azure Marketplace also have Windows Firewall enabled by default. Load balancers at the perimeter of Azure public-facing networks control communications based on IP ACLs managed by customer administrators.

If Azure moves a customer's data as part of normal operations or during a disaster, it does so over private, encrypted communications channels. Other capabilities employed by Azure to use in virtual networks and firewalls are:

- **Native host firewall**: Azure Service Fabric and Azure Storage run on a native OS that has no hypervisor. Hence the windows firewall is configured with the previous two sets of rules. Storage runs native to optimize performance.
- **Host firewall**: The host firewall is to protect the host operating system that runs the hypervisor. The rules are programmed to allow only the Service Fabric controller and jump boxes to talk to the host OS on a specific port. The other exceptions are to allow DHCP response and DNS Replies. Azure uses a machine configuration file that has the template of firewall rules for the host OS. The host itself is protected from external attack by a Windows firewall configured to permit communication only from known, authenticated sources.
- **Guest firewall**: Replicates the rules in the virtual machine Switch packet filter but programmed in different software (such as the Windows Firewall piece of the guest OS). The guest virtual machine firewall can be configured to restrict communications to or from the guest virtual machine, even if the communication is permitted by configurations at the host IP Filter. For example, you may choose to use the guest virtual machine firewall to restrict communication between two of your VNets that have been configured to connect to one another.
- **Storage firewall (FW)**: The firewall on the storage front end filters traffic to be only on ports 80/443 and other necessary utility ports. The firewall on the storage back end restricts communications to come only from storage front-end servers.
- **Virtual Network Gateway**: The Azure Virtual Network Gateway serves as the cross-premises gateway connecting your workloads in Azure Virtual Network to your on-premises sites. It is required to connect to on-

premises sites through IPsec site-to-site VPN tunnels, or through ExpressRoute circuits. For IPsec/IKE VPN tunnels, the gateways perform IKE handshakes and establish the IPsec S2S VPN tunnels between the virtual networks and on-premises sites. Virtual network gateways also terminate point-to-site VPNs.

Secure remote access

Data stored in the cloud must have sufficient safeguards enabled to prevent exploits and maintain confidentiality and integrity while in-transit. This includes network controls that tie in with an organization's policy-based, auditable identity and access management mechanisms.

Built-in cryptographic technology enables you to encrypt communications within and between deployments, between Azure regions, and from Azure to on-premises datacenters. Administrator access to virtual machines through remote desktop sessions, remote Windows PowerShell, and the Azure portal is always encrypted.

To securely extend your on-premises datacenter to the cloud, Azure provides both site-to-site VPN and point-to-site VPN, plus dedicated links with ExpressRoute (connections to Azure Virtual Networks over VPN are encrypted).

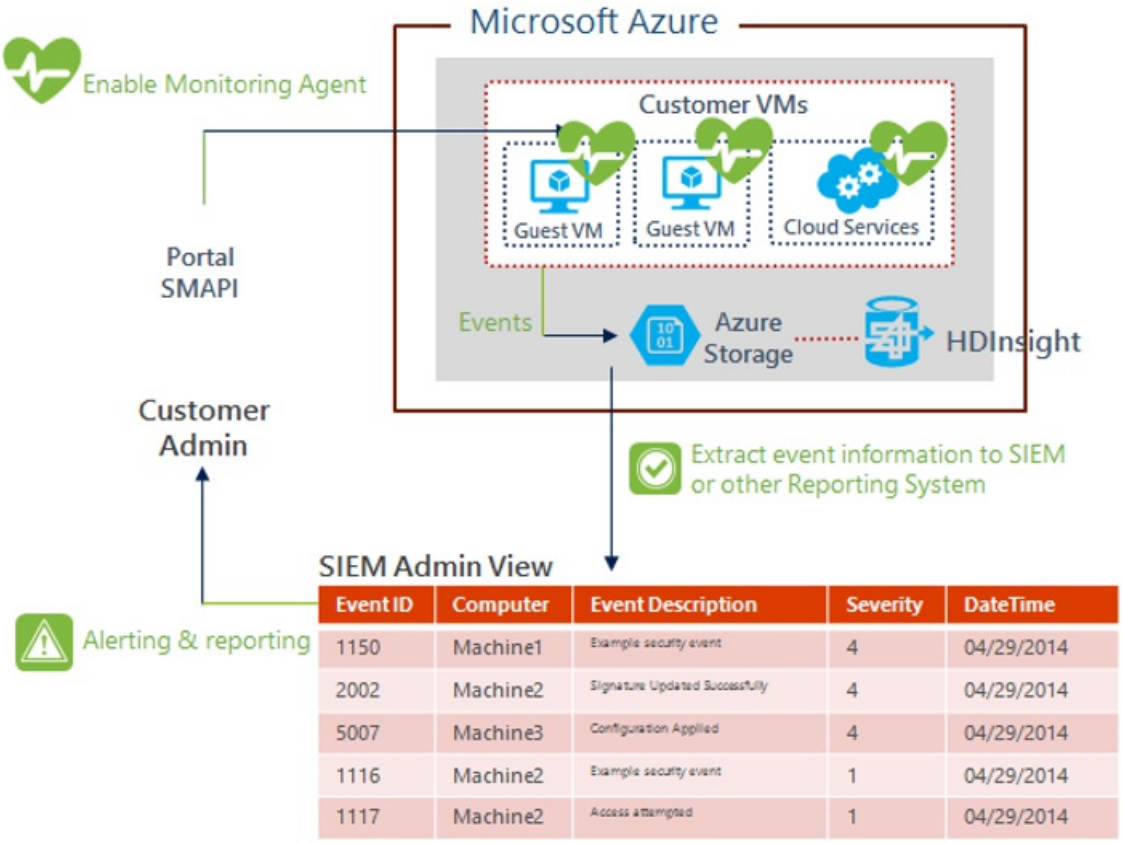
How Azure implements secure remote access

Connections to the Azure portal must always be authenticated, and they require SSL/TLS. You can configure management certificates to enable secure management. Industry-standard security protocols such as SSTP and IPsec are fully supported.

Azure ExpressRoute lets you create private connections between Azure datacenters and infrastructure that's on your premises or in a co-location environment. ExpressRoute connections do not go over the public Internet. They offer more reliability, faster speeds, lower latencies, and higher security than typical Internet-based links. In some cases, transferring data between on-premises locations and Azure by using ExpressRoute connections can also yield significant cost benefits.

Logging and monitoring

Azure provides authenticated logging of security-relevant events that generate an audit trail, and it is engineered to be resistant to tampering. This includes system information, such as security event logs in Azure infrastructure virtual machines and Azure AD. Security event monitoring includes collecting events such as changes in DHCP or DNS server IP addresses; attempted access to ports, protocols, or IP addresses that are blocked by design; changes in security policy or firewall settings; account or group creation; and unexpected processes or driver installation.



Audit logs recording privileged user access and activities, authorized and unauthorized access attempts, system exceptions, and information security events are retained for a set period of time. The retention of your logs is at your discretion because you configure log collection and retention to your own requirements.

How Azure implements logging and monitoring

Azure deploys Management Agents (MA) and Azure Security Monitor (ASM) agents to each compute, storage, or fabric node under management whether they are native or virtual. Each Management Agent is configured to authenticate to a service team storage account with a certificate obtained from the Azure certificate store and forward pre-configured diagnostic and event data to the storage account. These agents are not deployed to customers' virtual machines.

Azure administrators access logs through a web portal for authenticated and controlled access to the logs. An administrator can parse, filter, correlate, and analyze logs. The Azure service team storage accounts for logs are protected from direct administrator access to help prevent against log tampering.

Microsoft collects logs from network devices using the Syslog protocol, and from host servers using Microsoft Audit Collection Services (ACS). These logs are placed into a log database from which alerts for suspicious events are generated. The administrator can access and analyze these logs.

Azure Diagnostics is a feature of Azure that enables you to collect diagnostic data from an application running in Azure. This is diagnostic data for debugging and troubleshooting, measuring performance, monitoring resource usage, traffic analysis, capacity planning, and auditing. After the diagnostic data is collected, it can be transferred to an Azure storage account for persistence. Transfers can either be scheduled or on demand.

Threat mitigation

In addition to isolation, encryption, and filtering, Azure employs a number of threat mitigation mechanisms and processes to protect infrastructure and services. These include internal controls and technologies used to detect and remediate advanced threats such as DDoS, privilege escalation, and the OWASP Top-10.

The security controls and risk management processes Microsoft has in place to secure its cloud infrastructure reduce the risk of security incidents. In the event an incident occurs, the Security Incident Management (SIM) team

within the Microsoft Online Security Services and Compliance (OSSC) team is ready to respond at any time.

How Azure implements threat mitigation

Azure has security controls in place to implement threat mitigation and also to help customers mitigate potential threats in their environments. The following list summarizes the threat mitigation capabilities offered by Azure:

- Azure Antimalware is enabled by default on all infrastructure servers. You can optionally enable it within your own virtual machines.
- Microsoft maintains continuous monitoring across servers, networks, and applications to detect threats and prevent exploits. Automated alerts notify administrators of anomalous behaviors, allowing them to take corrective action on both internal and external threats.
- You can deploy third-party security solutions within your subscriptions, such as web application firewalls from Barracuda.
- Microsoft's approach to penetration testing includes "Red-Teaming," which involves Microsoft security professionals attacking (non-customer) live production systems in Azure to test defenses against real-world, advanced, persistent threats.
- Integrated deployment systems manage the distribution and installation of security patches across the Azure platform.

Next steps

Azure Trust Center

Azure Security Team Blog

Microsoft Security Response Center

Active Directory Blog

Azure Security Services and Technologies

11/22/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Barclay Neira

---

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?"

We understand that when you're evaluating your cloud service provider technical options, it's helpful to have such a list available that you can use to dig down deeper when the time is right for you.

The following is our initial effort at providing a list. Over time, this list will change and grow, just as Azure does. The list is categorized, and the list of categories will also grow over time. Make sure to check this page on a regular basis to stay up-to-date on our security-related services and technologies.

Azure Security - General

- Azure Security Center
- Azure Key Vault
- Azure Disk Encryption
- Log Analytics
- Azure Dev/Test Labs

Azure Storage Security

- Azure Storage Service Encryption
- StorSimple Encrypted Hybrid Storage
- Azure Client-Side Encryption
- Azure Storage Shared Access Signatures
- Azure Storage Account Keys
- Azure File Shares with SMB 3.0 Encryption
- Azure Storage Analytics

Azure Database Security

- Azure SQL Firewall
- Azure SQL Cell Level Encryption
- Azure SQL Connection Encryption
- Azure SQL Authentication
- Azure SQL Always Encryption
- Azure SQL Column Level Encryption
- Azure SQL Transparent Data Encryption
- Azure SQL Database Auditing

Azure Identity and Access Management

- Azure Role Based Access Control
- Azure Active Directory
- Azure Active Directory B2C
- Azure Active Directory Domain Services
- Azure Multi-Factor Authentication

Backup and Disaster Recovery

- Azure Backup
- Azure Site Recovery

Azure Networking

- Network Security Groups
- Azure VPN Gateway
- Azure Application Gateway
- Azure Load Balancer
- Azure ExpressRoute
- Azure Traffic Manager
- Azure Application Proxy

Azure Network Security Overview

11/22/2016 • 16 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil

---

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 goal of this article is to make it easier for you to understand what Microsoft Azure has to offer in the area of network security. Here we provide basic explanations for core network security concepts and requirements. We also provide you information on what Azure has to offer in each of these areas. There are numerous links to other content that will enable you to get a deeper understanding for the areas in which you're interested.

This Azure Network Security Overview article will focus on the following:

- Azure networking
- Network access control
- Secure remote access and cross-premises connectivity
- Availability
- Logging
- Name resolution
- DMZ architecture
- Azure Security Center

Azure 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 helps insure that network traffic in your deployments is not accessible to other Microsoft 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 an Azure Virtual Network. 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. Access controls are based on allow or deny decisions for connections to and from your virtual machine or service.

Azure supports several types of network access control. These include:

- 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 make sure that your virtual machines and the network services that run on those virtual machines can communicate only with other networked devices that they need to communicate with and all other connection attempts are blocked.

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. NSGs do not provide application layer inspection or authenticated access controls.

Learn more:

- Network Security Groups

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. If routing is configured incorrectly, applications and services hosted on your virtual machine may connect to devices you don't want them to connect to, including devices owned and operated by potential attackers.

Azure networking supports the ability to customize the routing behavior for network traffic on your Azure Virtual Networks. This enables you to alter the default routing table entries in your Azure 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 Azure Virtual Network through a specific location.

For example, you might have a virtual network security appliance on your Azure Virtual Network. You want to make sure that all traffic to and from your Azure Virtual Network goes through that virtual security appliance. 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. Note that this is different from accepting incoming connections and then responding to them. Front-end web servers need to respond to request 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 may represent a security risk because these connections could be used to download malware. Even if you do wish 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 so that you can 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 User Defined Routes.

Learn more:

- What are User Defined Routes and IP Forwarding

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 levels higher than the network.

For example, your security requirements might include:

- Authentication and authorization prior to 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 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".

Secure Remote Access and Cross Premises Connectivity

Setup, configuration and management of your Azure resources needs to be done remotely. In addition, you may 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 an Azure Virtual Network
- Connect your on-premises network to an Azure Virtual Network with a VPN
- Connect your on-premises network to an Azure Virtual Network with a dedicated WAN link
- Connect Azure Virtual Networks to each other

Connect Individual Workstations to an Azure Virtual Network

There may be times when you want to enable individual developers or operations personnel to manage virtual machines and services in Azure. For example, you need access to a virtual machine on an Azure Virtual Network and 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 uses the SSTP VPN protocol to enable you to set up a private and secure connection between the user and the Azure Virtual Network. Once the VPN connection is established, the user will be able to RDP or SSH over the VPN link into any virtual machine on the Azure Virtual Network (assuming that the user can authenticate and is authorized).

Learn more:

- Configure a Point-to-Site Connection to a Virtual Network using PowerShell

Connect Your On-Premises Network to an Azure Virtual Network with a VPN

You may want to connect your entire corporate network, or portions of it, to an Azure Virtual Network. This is common in hybrid IT scenarios where companies extend their on-premises datacenter into Azure. In many cases companies will host parts of a service in Azure and parts on-premises, such as when a solution includes front-end web servers in Azure and back-end databases on-premises. These kind 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 a point-to-site VPN connects a single device to an Azure Virtual Network, while a site-to-site VPN connects an entire network (such as your on-premises network) to an Azure Virtual Network. Site-to-site VPNs to an Azure 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 an Azure 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 Azure Virtual Networks may be considered bandwidth constrained for some applications and purposes, as they max out at around 200Mbps.

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 an Azure Virtual Network. This is enabled through Azure ExpressRoute.

Learn more:

- ExpressRoute technical overview

Connect Azure Virtual Networks to Each Other

It is possible for you to use many Azure Virtual Networks for your deployments. There are many reasons why you might do this. One of the reasons might be to simplify management; another might be for security reasons. Regardless of the motivation or rationale for putting resources on different Azure Virtual Networks, there may be times when you want resources on each of the networks to connect with one another.

One option would be for services on one Azure Virtual Network to connect to services on another Azure Virtual Network by "looping back" through the Internet. The connection would start on one Azure Virtual Network, go through the Internet, and then come back to the destination Azure Virtual Network. This option exposes the connection to the security issues inherent to any Internet-based communication.

A better option might be to create an Azure Virtual Network-to-Azure Virtual Network site-to-site VPN. This Azure Virtual Network-to-Azure Virtual Network site-to-site VPN uses the same IPsec tunnel mode protocol as the cross-premises site-to-site VPN connection mentioned above.

The advantage of using an Azure Virtual Network-to-Azure Virtual Network site-to-site VPN is that the VPN connection is established over the Azure network fabric; it does not connect 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

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:

- Increase availability – when you load balance connections across multiple devices, one or more of the devices can become unavailable and the services running on the remaining online devices can continue to serve the

content from the service

- Increase performance – when you load balance connections across multiple devices, a single device doesn't have to take the processor hit. 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 to help insure adequate levels of performance and high availability. In contrast to traditional network-based load balancers, the load balancing decisions made by HTTP-based load balancers are based on characteristics of the HTTP protocol, not on the network and transport layer protocols.

To provide you HTTP-based load balancing for your web-based services, Azure provides you the Azure Application Gateway. The Azure Application Gateway supports:

- HTTP-based load balancing – load balancing decisions are made based on characteristic special to the HTTP protocol
- 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 insures stability of transactions.
- SSL offload – when a client connection is established with the load balancer, that session between the client and the load balancer is encrypted using the HTTPS (SSL/) protocol. However, in order to increase performance, you have the option to have the connection between the load balancer and the web server behind the load balancer use the HTTP (unencrypted) protocol. This is referred to as "SSL offload" because the web servers behind the load balancer don't experience the processor overhead involved with encryption, and therefore should be able to service requests more quickly.
- URL-based content routing – this feature makes it possible for the load balancer to make decisions on 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 load balancing 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 the Azure Load Balancer. Some key characteristics of the Azure 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 will 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 data centers located throughout the world, it's possible for an entire geopolitical region to become unavailable and still have the application up and running.

In addition to the availability advantages you get by hosting applications in globally distributed datacenters, you also can get performance benefits. These performance benefits can be obtained by using a mechanism that directs requests for the service to the datacenter that is nearest to the device that is making the request.

Global load balancing can provide you both of these benefits. In Azure, you can gain the benefits of global load balancing by using Azure Traffic Manager.

Learn more:

- What is Traffic Manager?

Logging

Logging at a network level is a key function for any network security scenario. In Azure, you can log information obtained for Network Security Groups to get network level logging information. With NSG logging, you get information from:

- Audit logs – these logs are used to view all operations submitted to your Azure subscriptions. These logs are enabled by default and can be used within the Azure portal.
- 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)

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 – internal name resolution is used by services on your Azure 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 – external name resolution is used by people and devices outside of your on-premises and Azure 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:

- An Azure Virtual Network DNS server – when you create a new Azure Virtual Network, a DNS server is created for you. This DNS server can resolve the names of the machines located on that Azure Virtual Network. This DNS server is not configurable and is managed by the Azure fabric manager, thus making it a secure name resolution solution.
- Bring your own DNS server – you have the option of putting a DNS server of your own choosing on your Azure Virtual Network. This DNS server could 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 (VNet)

For external DNS 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 will 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 a highly available and performant 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

DMZ Architecture

Many enterprise organizations use DMZs to segment their networks to create a buffer-zone between the Internet and their services. The DMZ 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 DMZ segment and another network interface connected to a network that has virtual machines and services that accept inbound connections from the Internet.

There are a number of variations of DMZ design and the decision to deploy a DMZ, and then what type of DMZ to use if you decide to use one, is based on your network security requirements.

Learn more:

- Microsoft Cloud Services and Network Security

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.

Azure 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

Azure Network Security Best Practices

11/15/2016 • 17 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear • Yuri Diogenes

---

Microsoft Azure enables you to connect virtual machines and appliances to other networked devices by placing them on Azure Virtual Networks. An Azure Virtual Network is a virtual network construct that allows you to connect virtual network interface cards to a virtual network to allow TCP/IP-based communications between network enabled devices. Azure Virtual Machines connected to an Azure Virtual Network are able to connect to devices on the same Azure Virtual Network, different Azure Virtual Networks, on the Internet or even on your own on-premises networks.

In this article we will discuss 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, 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 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.

Azure Network security best practices discussed in this article include:

- Logically segment subnets
- Control routing behavior
- Enable Forced Tunneling
- Use Virtual network appliances
- Deploy DMZs for security zoning
- Avoid exposure to the Internet with dedicated WAN links
- Optimize uptime and performance
- Use global load balancing
- Disable RDP Access to Azure Virtual Machines
- Enable Azure Security Center
- Extend your datacenter into Azure

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.

Similar to what you do on-premises, you'll want to segment the larger address space into subnets. You can use CIDR based subnetting principles to create your subnets.

Routing between subnets will happen automatically and you do not need to manually configure routing tables. However, the default setting is that there are no network access controls between the subnets you create on the Azure Virtual Network. In order to create network access controls between subnets, you'll need to put something between the subnets.

One of the things you can use to accomplish this task is 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 can allow or deny traffic to and from single IP address, to and from multiple IP addresses or even to and from entire subnets.

Using NSGs for network access control between subnets enables you to put resources that belong to the same security zone or role in their own subnets. For example, think of a simple 3-tier application that has a web tier, an application logic tier and a database tier. You put virtual machines that belong to each of these tiers into their own subnets. Then you use NSGs to control traffic between the subnets:

- Web tier virtual machines can only initiate connections to the application logic machines and can only accept connections from the Internet
- Application logic virtual machines can only initiate connections with database tier and can only accept connections from the web tier
- Database tier virtual machines cannot initiate connection with anything outside of their own subnet and can only accept connections from the application logic tier

To learn more about Network Security Groups and how you can use them to logically segment your Azure Virtual Networks, please read the article What is a Network Security Group (NSG).

Control routing behavior

When you put a virtual machine on an Azure Virtual Network, you'll notice that the virtual machine can connect to any other virtual machine on the same Azure Virtual Network, even if the other virtual machines are on different subnets. The reason why this is possible is that there is a collection of system routes that are enabled by default that allow this type of communication. These default routes allow virtual machines on the same Azure Virtual Network to initiate connections with each other, and with the Internet (for outbound communications to the Internet only).

While the default system routes are useful for many deployment scenarios, there are times when you want to customize the routing configuration for your deployments. These customizations will allow you to configure the next hop address to reach specific destinations.

We recommend that you configure User Defined Routes when you deploy a virtual network security appliance, which we'll talk about in a later best practice.

> **NOTE**
> user Defined Routes are not required and the default system routes will work in most instances.

You can learn more about User Defined Routes and how to configure them by reading the article What are User Defined Routes and IP Forwarding.

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 VPN connections. Imagine that you establish a VPN connection from your

hotel room to your corporate network. This connection allows you to connect to resources on your corporate network and all communications to resources on 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 and therefore recommend that split tunneling be disabled and all connections, those destined for the Internet and those destined for corporate resources, go through the VPN tunnel. The advantage of doing this is that connections to the Internet are then forced through the corporate network security devices, which wouldn't be the case if the VPN client connected to the Internet outside of the VPN tunnel.

Now let's bring this back to virtual machines on an Azure Virtual Network. The default routes for an Azure Virtual Network allow virtual machines to initiate traffic to the Internet. This too can represent a security risk, as these outbound connections could increase the attack surface of a virtual machine and be leveraged by attackers. For this reason, we recommend that you enable forced tunneling on your virtual machines when you have cross-premises connectivity between your Azure Virtual Network and your on-premises network. We will talk about cross premises connectivity later in this Azure networking best practices document.

If you do not have a cross premises connection, make sure you take advantage of Network Security Groups (discussed earlier) or Azure virtual network security appliances (discussed next) to prevent outbound connections to the Internet from your Azure Virtual Machines.

To learn more about forced tunneling and how to enable it, please read the article Configure Forced Tunneling using PowerShell and Azure Resource Manager.

Use virtual network appliances

While Network Security Groups and User Defined Routing can provide a certain measure of network security at the network and transport layers of the OSI model, there are going to be situations where you'll 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 significantly enhanced levels of security over what is provided by network level controls. Some of the network security capabilities provided by virtual network security appliances include:

- Firewalling
- Intrusion detection/Intrusion Prevention
- Vulnerability management
- Application control
- Network-based anomaly detection
- Web filtering
- Antivirus
- Botnet protection

If you require a higher level of network security than you can obtain with network level access controls, then we recommend that you investigate and deploy Azure virtual network security appliances.

To learn about what Azure virtual network security appliances are available, and about their capabilities, please visit the Azure Marketplace and search for "security" and "network security".

Deploy DMZs for security zoning

A DMZ or "perimeter network" is a physical or logical network segment that is designed to provide an additional layer of security between your assets and the Internet. The intent of the DMZ is to place specialized network access control devices on the edge of the DMZ network so that only desired traffic is allowed past the network security

device and into your Azure Virtual Network.

DMZs 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 would typically enable 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.

While this is the basic design of a DMZ, there are many different DMZ designs, such as back-to-back, tri-homed, multi-homed, and others.

We recommend for all high security deployments that you consider deploying a DMZ to enhance the level of network security for your Azure resources.

To learn more about DMZs and how to deploy them in Azure, please read the article Microsoft Cloud Services and Network Security.

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 will be running in Azure while other components remain on-premises.

In the hybrid IT scenario, there is usually some type of cross-premises connectivity. This cross-premises connectivity allows the company to connect their on-premises networks to Azure Virtual Networks. There are two cross-premises connectivity solutions available:

- Site-to-site VPN
- ExpressRoute

Site-to-site VPN represents a virtual private connection between your on-premises network and an Azure Virtual Network. This connection takes place over the Internet and allows you to "tunnel" information inside an encrypted link between your network and Azure. Site-to-site VPN is a secure, mature technology that has been deployed by enterprises of all sizes for decades. Tunnel encryption is performed using IPsec tunnel mode.

While site-to-site VPN is a trusted, reliable, and established technology, traffic within the tunnel does traverse the Internet. In addition, bandwidth is relatively constrained to a maximum of about 200Mbps.

If you require an exceptional level of security or performance for your cross-premises connections, we recommend that you use Azure 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 inherent in Internet communications.

To learn more about how Azure ExpressRoute works and how to deploy, please read the article ExpressRoute Technical Overview.

Optimize uptime and performance

Confidentiality, integrity and availability (CIA) comprise the triad of today's most influential security model. Confidentiality is about encryption and privacy, integrity is about making sure that data is not changed by unauthorized personnel, and availability is about making sure that authorized individuals are able to access the information they are authorized to access. Failure in any one of these areas represents a potential breach in security.

Availability can be thought of as being about uptime and performance. If a service is down, information can't be accessed. If performance is so poor as to make the data unusable, then we can consider the data to be inaccessible.

Therefore, from a security perspective, we need to do whatever we can to make sure our services have optimal uptime and performance. A popular and effective method used to enhance availability and performance is to use 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 will stop sending traffic to that server and redirect traffic 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. We'll address appropriateness in the following sections. At the Azure Virtual Network level, Azure provides you with three primary load balancing options:

- HTTP-based load balancing
- External load balancing
- Internal load balancing

HTTP-based Load Balancing

HTTP-based load balancing bases decisions about what server to send connections using characteristics of the HTTP protocol. Azure has an HTTP load balancer that goes by the name of Application Gateway.

We recommend that you us Azure Application Gateway when:

- Applications that require requests from the same user/client session to reach the same back-end virtual machine. Examples of this would be shopping cart apps and web mail servers.
- Applications that want to free web server farms from SSL termination overhead by taking advantage of Application Gateway's SSL offload feature.
- Applications, such as a content delivery network, that require multiple HTTP requests on the same long-running TCP connection to be routed or load balanced to different back-end servers.

To learn more about how Azure Application Gateway works and how you can use it in your deployments, please read the article Application Gateway Overview.

External Load Balancing

External load balancing takes place when incoming connections from the Internet are load balanced among your servers located in an Azure Virtual Network. The Azure External Load balancer can provide you this capability and we recommend that you use it when you don't require the sticky sessions or SSL offload.

In contrast to HTTP-based load balancing, the External Load Balancer uses information at the network and transport layers of the OSI networking model to make decisions on what server to load balance connection to.

We recommend that you use External Load Balancing whenever you have stateless applications accepting incoming requests from the Internet.

To learn more about how the Azure External Load Balancer works and how you can deploy it, please read the article Get Started Creating an Internet Facing Load Balancer in Resource Manager using PowerShell.

Internal Load Balancing

Internal load balancing is similar to external load balancing and uses the same mechanism to load balance connections to the servers behind them. The only difference is that the load balancer in this case is accepting connections from virtual machines 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.

We recommend that you use internal load balancing for scenarios that will benefit from this capability, such as when you need to load balance connections to SQL Servers or internal web servers.

To learn more about how Azure Internal Load Balancing works and how you can deploy it, please read the article Get Started Creating an Internal Load Balancer using PowerShell.

Use global load balancing

Public cloud computing makes it possible to deploy globally distributed applications that have components located in datacenters all over the world. This is possible on Microsoft Azure due to Azure's global datacenter presence. In contrast to the load balancing technologies mentioned earlier, global load balancing makes it possible to make services available even when entire datacenters might become unavailable.

You can get this type of global load balancing in Azure by taking advantage of Azure Traffic Manager. Traffic Manager makes is possible to load balance connections to your services based on the location of the user.

For example, if the user is making 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.

On the availability side, global load balancing makes sure that your service is available even if an entire datacenter should become available.

For example, if an Azure datacenter should become unavailable due to environmental reasons or due to outages (such as regional network failures), connections to your service would be rerouted to the nearest online datacenter. This global load balancing is accomplished by taking advantage of DNS policies that you can create in Traffic Manager.

We recommend that you use Traffic Manager for any cloud solution you develop that has a widely distributed scope across multiple regions and requires the highest level of uptime possible.

To learn more about Azure Traffic Manager and how to deploy it, please read the article What is Traffic Manager.

Disable RDP/SSH Access to Azure Virtual Machines

It is possible to reach Azure Virtual Machines using the Remote Desktop Protocol (RDP)and the Secure Shell (SSH) protocols. These protocols make it possible to manage virtual machines 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 various brute force techniques to gain access to Azure Virtual Machines. Once the attackers gain access, they can use your virtual machine as a launch point for compromising other machines on your Azure Virtual Network or even attack networked devices outside of Azure.

Because of this, 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 you can use to access these virtual machines for remote management:

- Point-to-site VPN
- Site-to-site VPN
- ExpressRoute

Point-to-site VPN is another term for a remote access VPN client/server connection. A point-to-site VPN enables a single user to connect to an Azure Virtual Network over the Internet. After the point-to-site connection is established, the user will be able to use RDP or SSH to connect to any virtual machines 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 virtual machines.

Point-to-site VPN is more secure than direct RDP or SSH connections because the user has to authenticate twice before connecting to a virtual machine. 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.

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. If you deploy a site-to-site VPN, users on your on-premises network will be able to connect to virtual machines on your Azure Virtual Network by using the RDP or SSH protocol over the site-to-site VPN connection and does not require you to allow direct RDP or SSH access over the Internet.

You can also use a dedicated WAN link to provide functionality similar to the site-to-site VPN. The main differences are 1. the dedicated WAN link doesn't traverse the Internet, and 2. dedicated WAN links are typically more stable and performant. Azure provides you a dedicated WAN link solution in the form of ExpressRoute.

Enable 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 broad ecosystem of security solutions.

Azure 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

We highly recommend that you enable Azure Security Center for all of your Azure deployments.

To learn more about Azure Security Center and how to enable it for your deployments, please read the article Introduction to Azure Security Center.

Securely extend your datacenter into Azure

Many enterprise IT organizations are looking to expand into the cloud instead of growing their on-premises datacenters. This expansion represents an extension of existing IT infrastructure into the public cloud. By taking advantage of cross-premises connectivity options it's possible to treat your Azure Virtual Networks as just another subnet on your on-premises network infrastructure.

However, there is a lot of planning and design issues that need to be addressed first. This is especially important in the area of network security. One of the best ways to understand how you approach such a design is to see an example.

Microsoft has created the Datacenter Extension Reference Architecture Diagram and supporting collateral to help you understand what such a datacenter extension would look like. This provides an example reference implementation that you can use to plan and design a secure enterprise datacenter extension to the cloud. We recommend that you review this document to get an idea of the key components of a secure solution.

To learn more about how to securely extend your datacenter into Azure, please view the video Extending Your Datacenter to Microsoft Azure.

Microsoft cloud services and network security

11/15/2016 • 36 min to read • Edit on GitHub

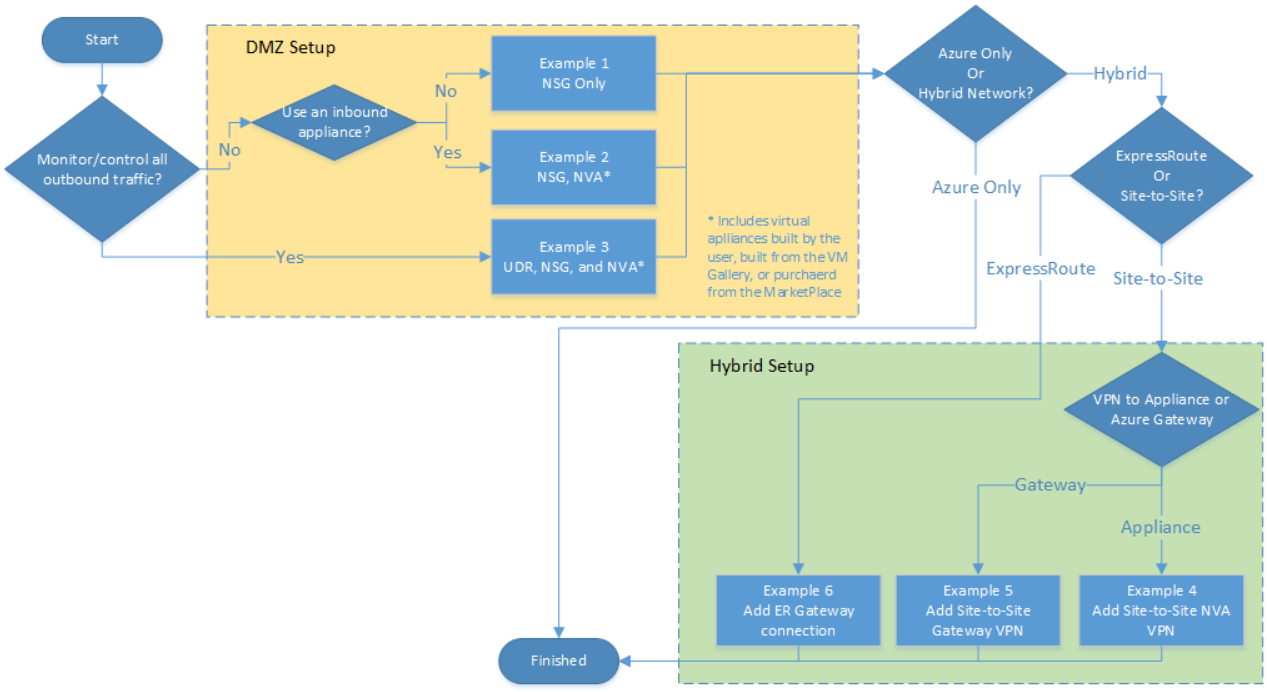
Contributors

Jon Ormond • Joseph Molnar • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Olivier Martin • James Dunn

Microsoft cloud services deliver hyperscale 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 multitier 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 more complete explanations, continue reading through the paper.



Example 1: Build a perimeter network (also known as DMZ, demilitarized zone, and 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 gateway VPN.
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

Microsoft has taken a leadership position supporting compliance initiatives required by enterprise customers. The following are some of the compliance certifications for Azure:

Global

- ISO/IEC 27001
- SOC 1
- SOC 2
- PCI DSS L1 version 3
- Cloud Security Alliance Cloud Security Matrix
- ISO / IEC 27018

United States

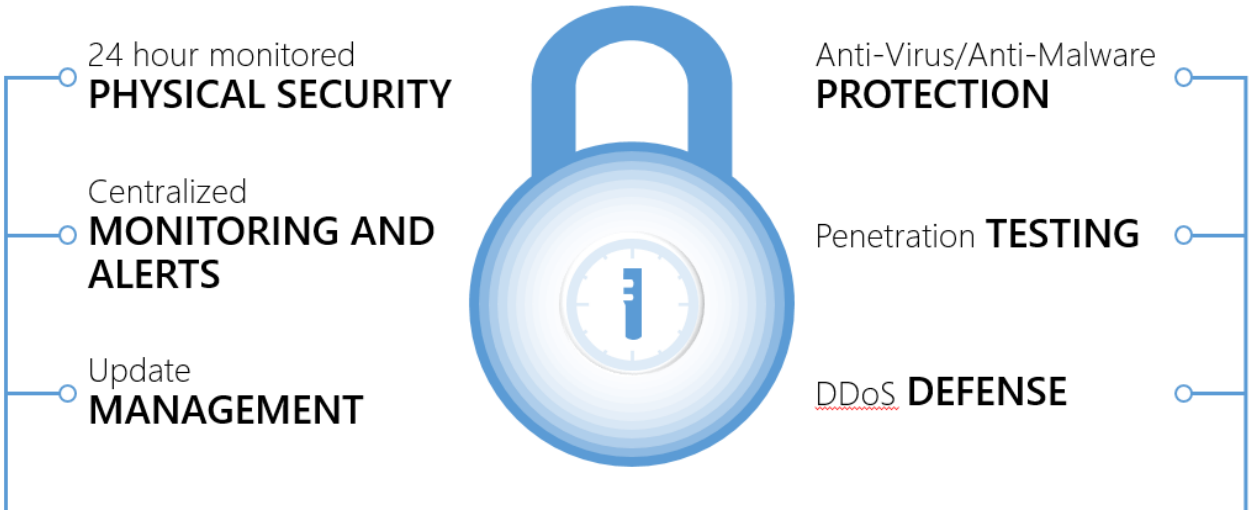
- FedRAMP
- HIPAA (Healthcare)
- FIPS 140-2
- Life Sciences GxP
- Family Educational Rights and Privacy Act
- Criminal Justice Information System

Regional

- European Union Model Clause
- United Kingdom G-Cloud
- China Multi Layer Protection Scheme
- China CCCPPF
- Singapore Multi-Tier Cloud Security
- Australian Signals Directorate I-RAP Assessment

For more details, see the compliance information on the Microsoft Trust Center.

Microsoft has a comprehensive approach to protect cloud infrastructure needed to run hyperscale global services. Microsoft cloud infrastructure includes hardware, software, networks, and administrative and operations staff, in addition to the physical datacenters.



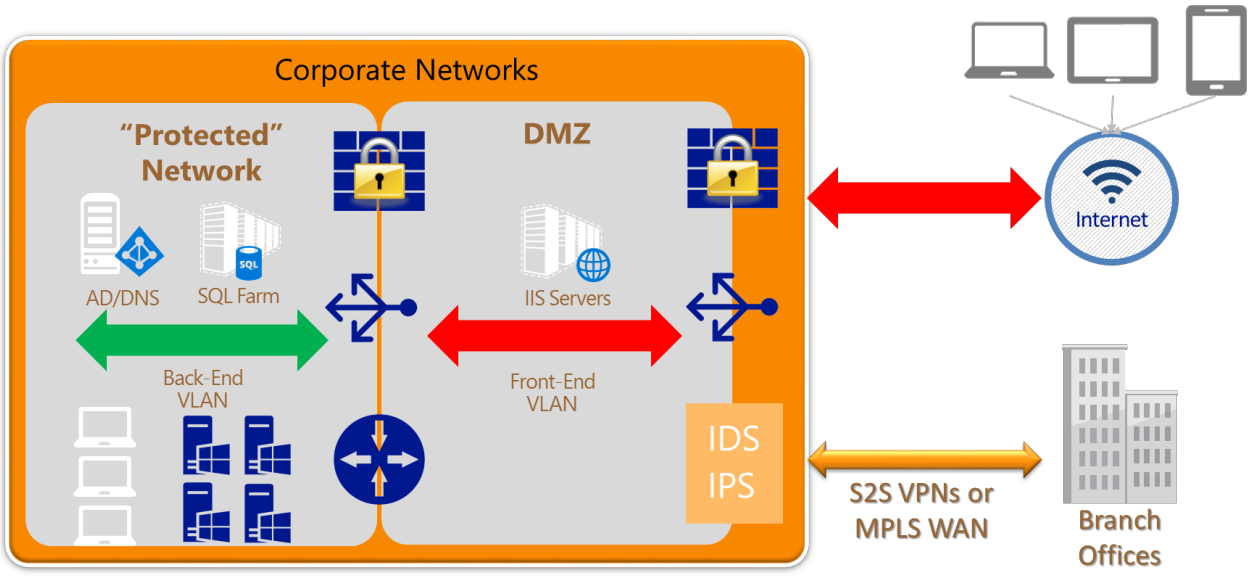
This approach provides a more secure foundation for customers to deploy their services in the Microsoft cloud. The next step is for customers to design and create a security architecture to protect these services.

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. Devices could include the following: firewalls, Distributed Denial of Service (DDoS) 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.

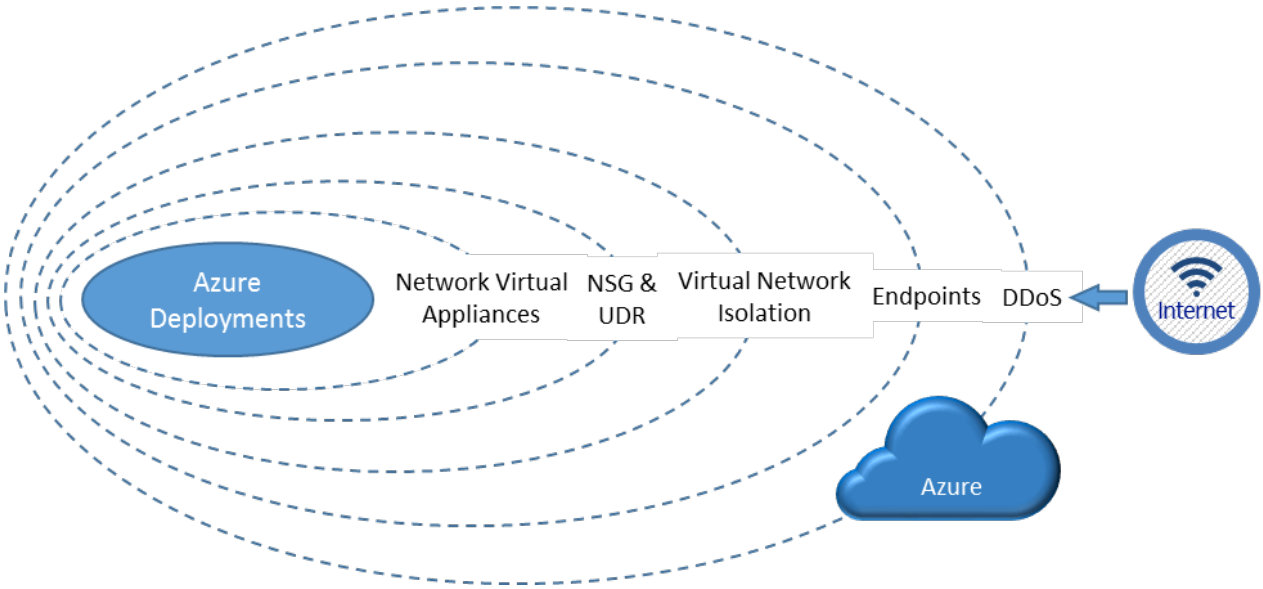


There are many architectures used to implement a perimeter network, from a simple load balancer in front of a web farm, to a multiple subnet perimeter network with varied mechanisms at each boundary to block traffic and protect the deeper layers of the corporate network. How the perimeter network is 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 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 Internet connectivity. This DDoS protection layer has no user configurable attributes and is not accessible to customer. This protects Azure as a platform from large-scale attacks, but will not directly protect individual customer application. Additional layers of resilience can be configured by the customer against a localized attack. For example, if customer A was attacked with a large-scale DDoS attack on a public endpoint, Azure blocks connections to that service. Customer A could fail over to another virtual network or service endpoint not involved with the attack to restore service. It should be noted that although customer A might be affected on that endpoint, no other services outside that endpoint would be affected. In addition, other customers and services would see no impact from that attack.
2. **Service endpoints**: Endpoints allow cloud services or resource groups to have public Internet IP addresses and ports exposed. The endpoint will use Network Address Translation (NAT) to route traffic to the internal address and port on the Azure virtual network. This is the primary path for external traffic to pass into the virtual network. The service endpoints are user configurable to determine which traffic is passed in, and how and where it's translated to on the virtual network.

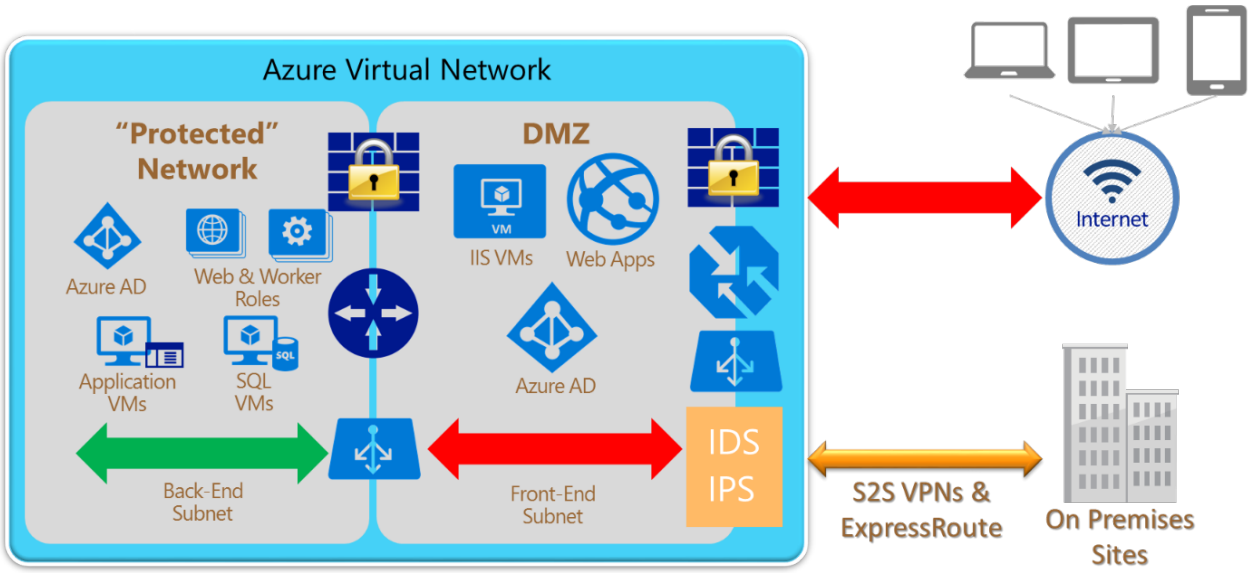
Once traffic reaches the virtual network, there are many features that come into play. Azure virtual networks are the foundation for customers to attach their workloads and where basic network-level security applies. It is a private network (a virtual network overlay) in Azure for customers with the 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. This is a critical property that ensures customer VMs and communication remains private within a virtual network.

- **Multitier topology**: Virtual networks allow customers to define multitier 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 do this, customers can use Azure VPN Gateways or third-party network virtual appliances. 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 multitiered secure network environment.

With these features and capabilities, one example of how a perimeter network architecture could be constructed in Azure is the following:



Perimeter network characteristics and requirements

The perimeter network is designed to be 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 IPs, 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. (That is, one 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 ensures the traffic between the perimeter network and other internal or private subnet tiers flows through a firewall or IDS/IPS virtual appliance on the subnet boundaries with user-defined routes.
- **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 this is not always 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:80 and TCP: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:** In order 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 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 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.

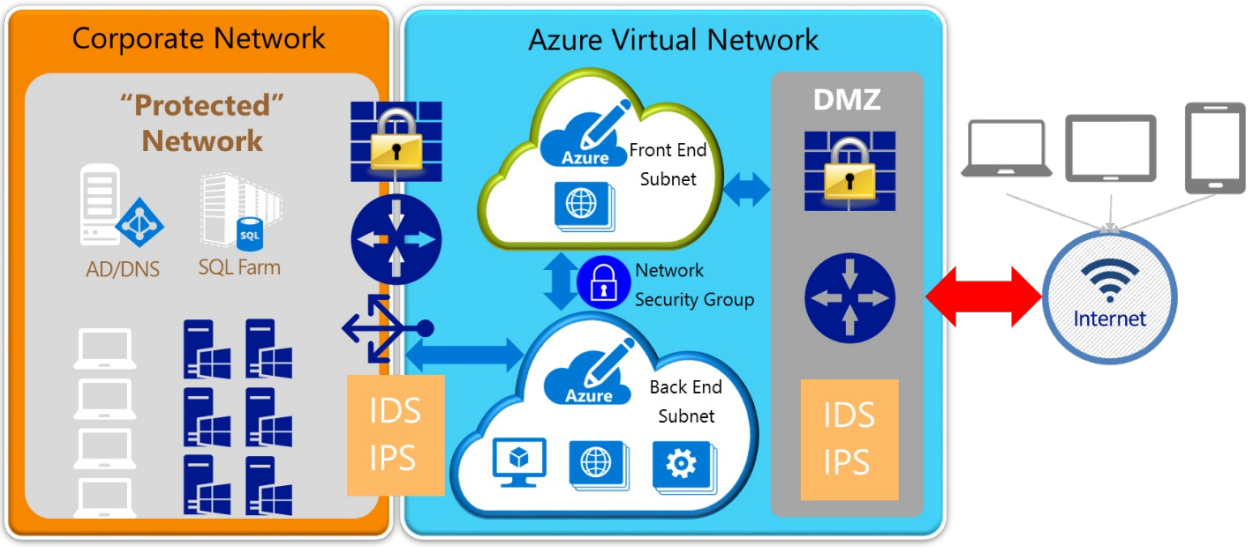
1) How many boundaries are needed?

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 really any number of security boundaries that can be applied in a given network.

The number and type of boundaries needed will vary based on a company's risk tolerance and the specific scenario being implemented. This is often a joint decision made by 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 the more difficult operations and troubleshooting can be, 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 which 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).

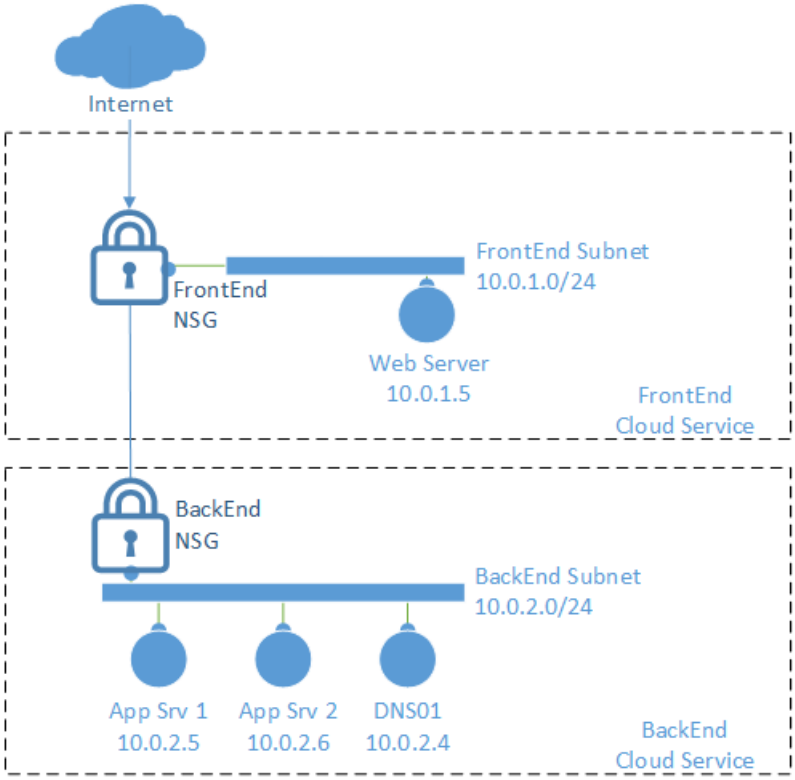
Additionally, traffic between virtual networks within Azure may be needed. These scenarios will be added at a later date.

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:

- Two cloud services: "FrontEnd001" and "BackEnd001"
- A virtual network, "CorpNetwork", 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")

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 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).

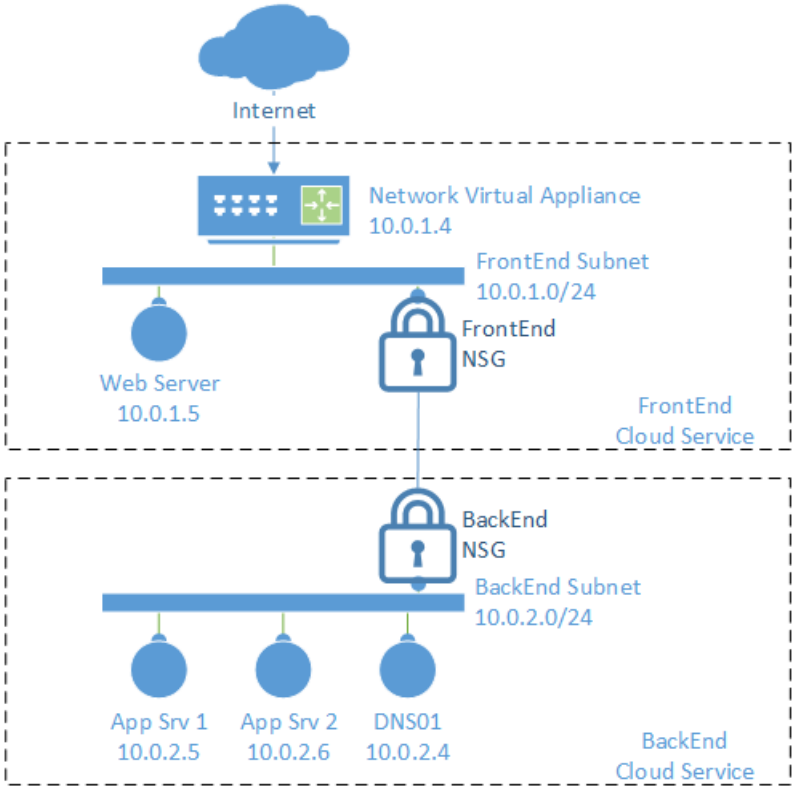
Conclusion

This 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 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:

- Two cloud services: “FrontEnd001” and “BackEnd001”
- A virtual network, “CorpNetwork”, 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 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.

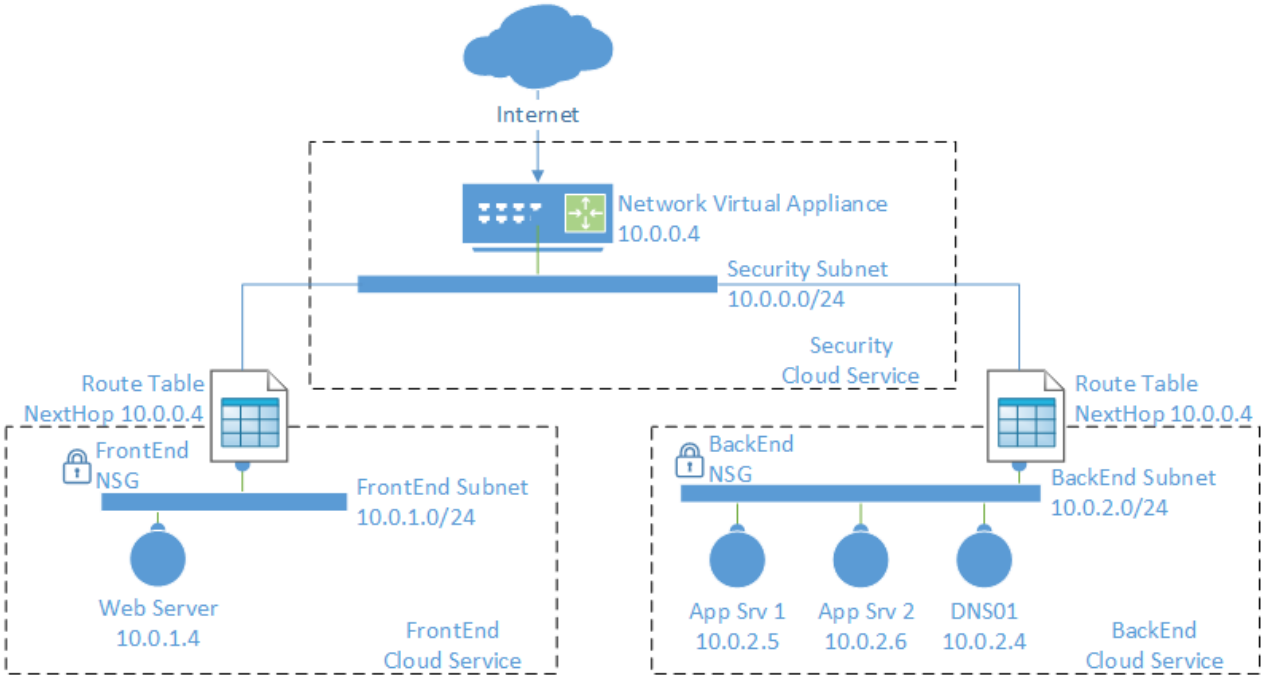
Conclusion

This 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 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:

- Three cloud services: “SecSvc001”, “FrontEnd001”, and “BackEnd001”
- A virtual network, “CorpNetwork”, 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 the defined address prefix(es) of 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. For the purpose of 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 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 will break local subnet communications.
>
> - In our example, 10.0.1.0/24 pointing to VNETLocal is critical as otherwise, 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 over to the NVA, which will send it to the subnet, and the subnet will re-send it to the NVA and so on.
> - Chances of a routing loop are typically higher on multi-nic appliances that are directly connected to each subnet they are communicating with, which is often of traditional, on-premises, appliances.

Once the routing tables are created, they are bound to their subnets. The front-end subnet routing table, once created and bound to the subnet, would look like this:

```
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. This 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 traffic from AppVM01 makes a request to the DNS01 server, UDR would route this 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 in conjunction 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 as an

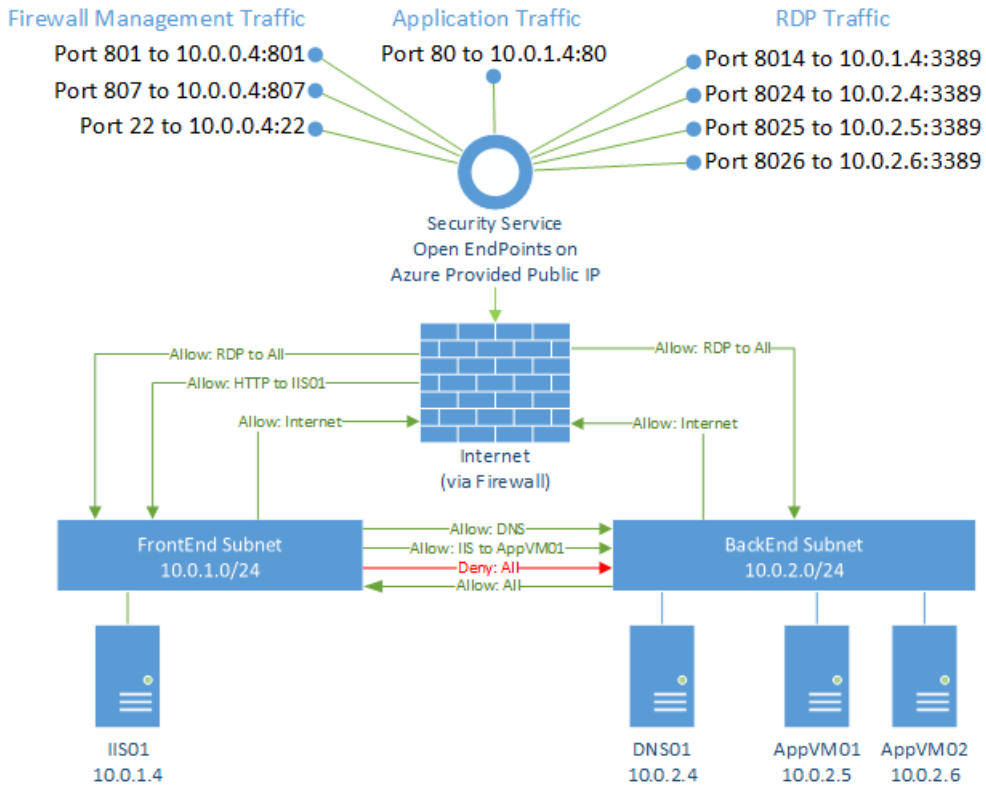
asymmetric flow and would drop the outbound traffic. Thus there are three layers of security protecting the subnets:

- No open endpoints on the FrontEnd001 and BackEnd001 cloud services.
- 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 will flow 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 will hit 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 will 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. This is because the firewall is the only resource on that subnet, and this 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 somewhat 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. This could also be bundled 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).
     - 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 application(s). 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.
     - 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 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 will be dropped by this rule. This is a default rule and usually activated. No modifications are generally needed.

> **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 all of the previous rules are created, it's important to review the priority of each rule to ensure traffic will be allowed or denied as desired. For this example, the rules are in priority order.

Conclusion

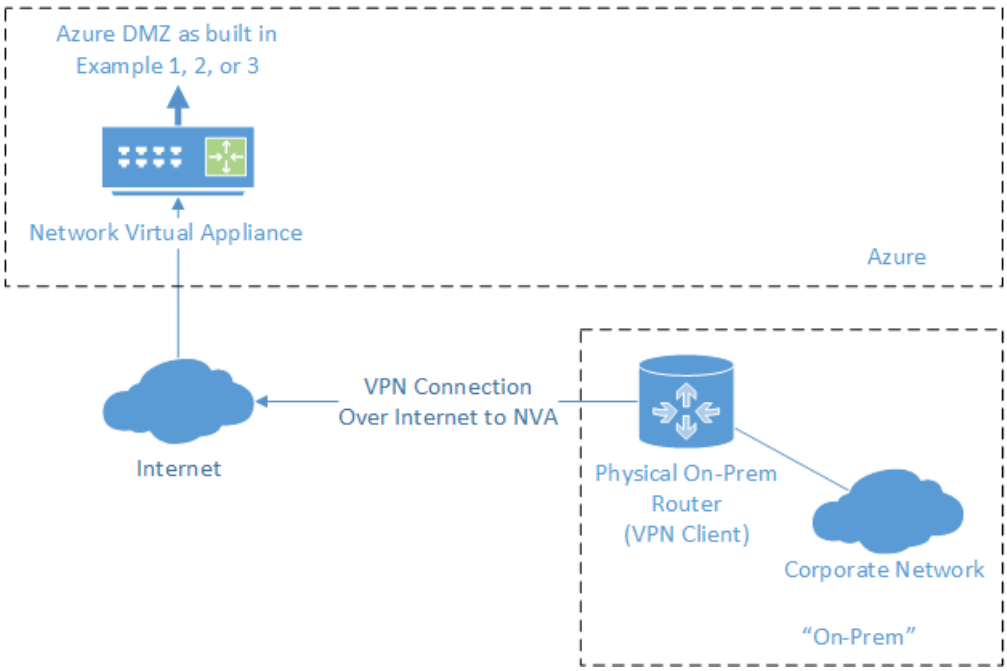
This is a more complex but more complete way of protecting and isolating the network than the previous examples. (Example 2 protects just the application, and Example 1 just isolates subnets). This design allows for monitoring traffic in both directions, and protects not just the inbound application 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 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 virtual private network

Back to Fast start | Detailed build instructions available soon



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.

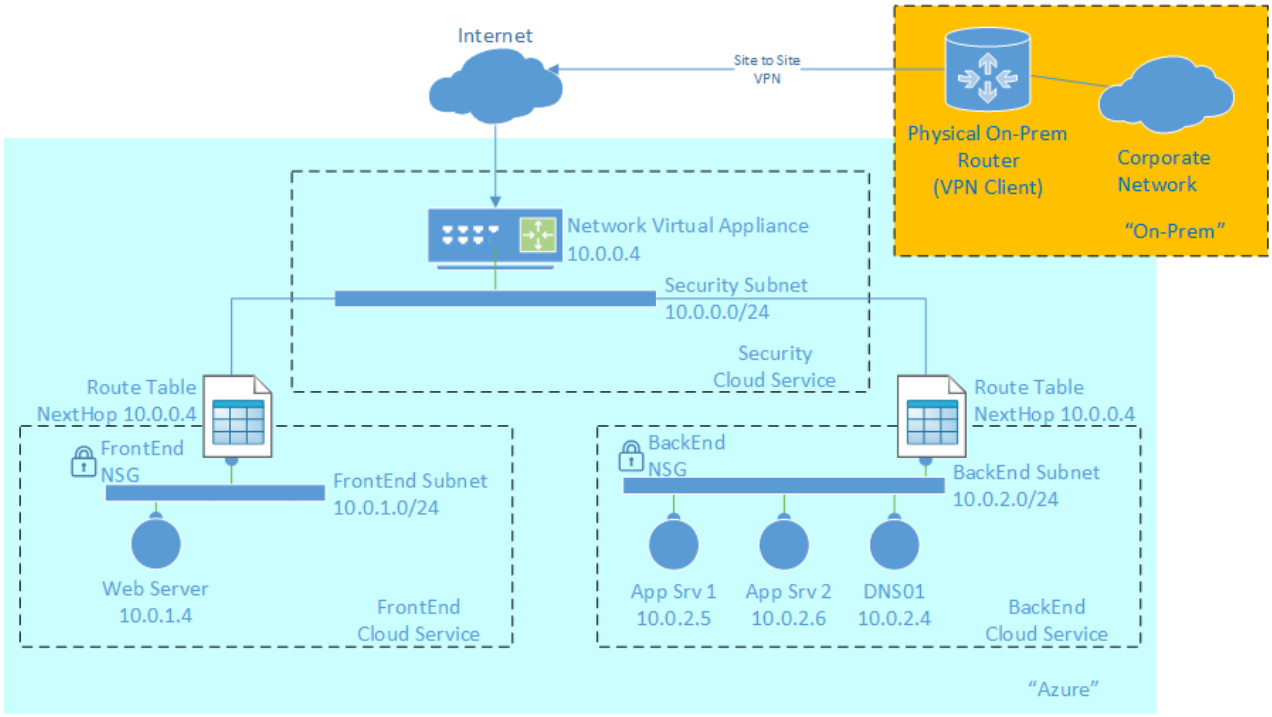
As shown in the previous figure, a VPN connection over the Internet (site-to-site) is used to connect an on-premises network to an Azure virtual network via an NVA.

> **NOTE**
> If you use ExpressRoute with the Azure Public Peering option enabled, a static route should be created. This 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.

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).

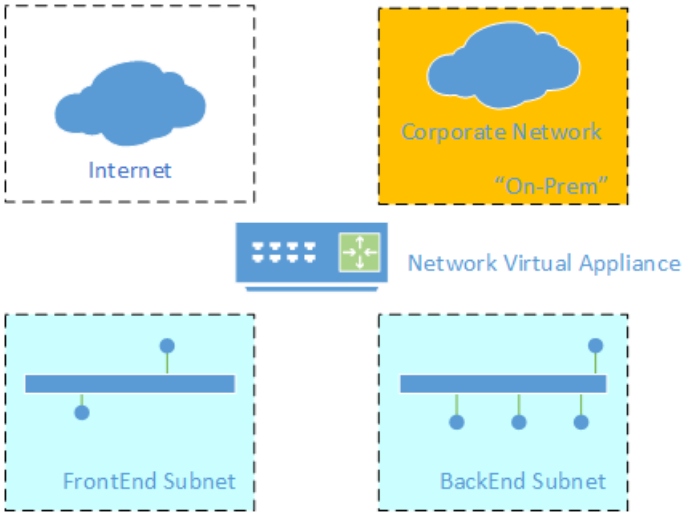
Traffic flows should be considered carefully, as they can be optimized or degraded by this design pattern, depending on the specific use case.

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:



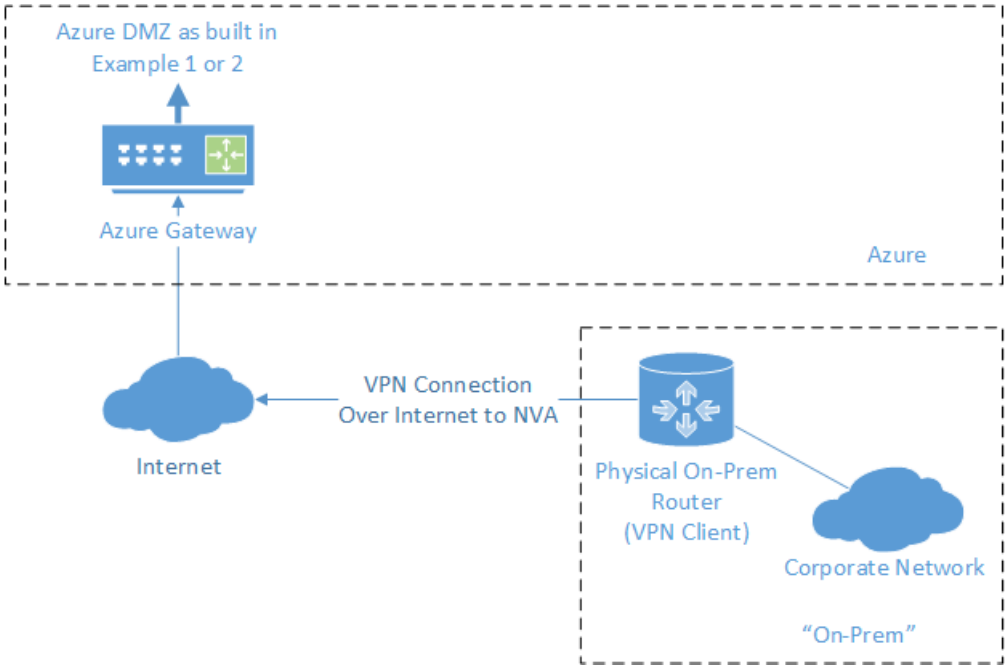
Conclusion

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 gateway VPN

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.

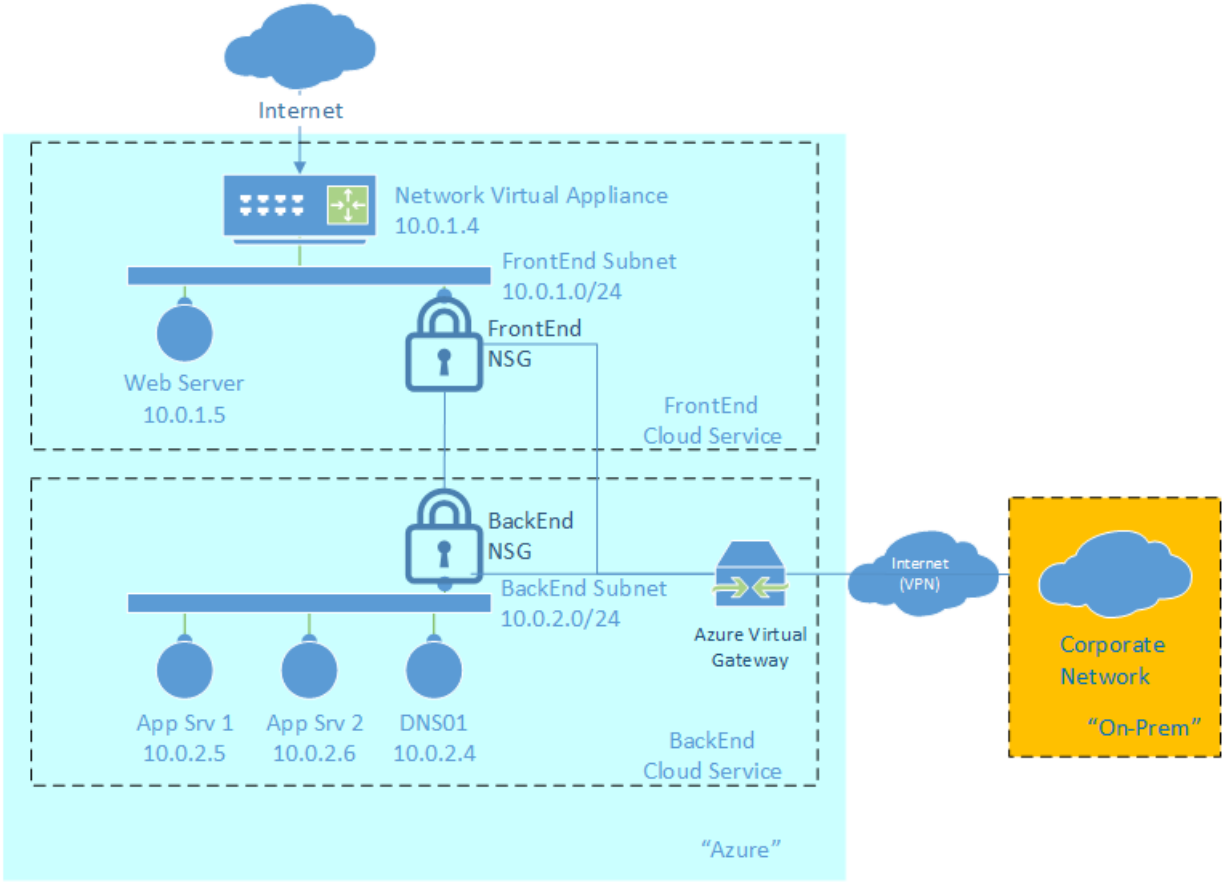
As shown in the preceding figure, a VPN connection over the Internet (site-to-site) is used to connect an on-premises network to an Azure virtual network via an Azure VPN gateway.

> **NOTE**
>
> If you use ExpressRoute with the Azure Public Peering option enabled, a static route should be created. This 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 option. 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 completely 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:



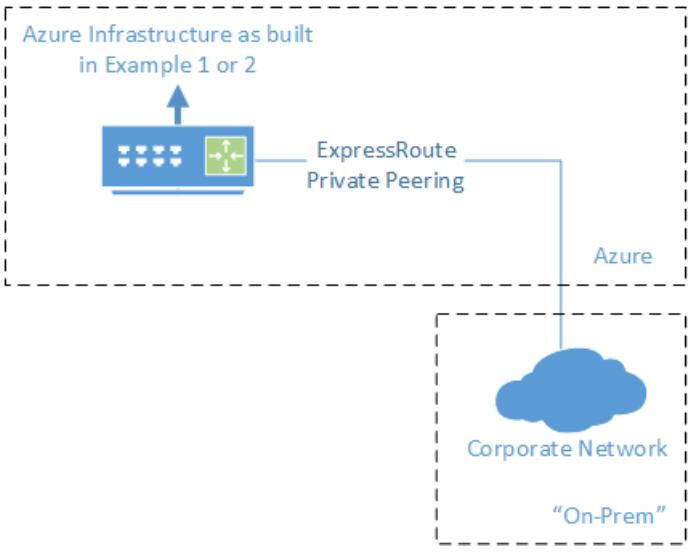
Conclusion

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 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.

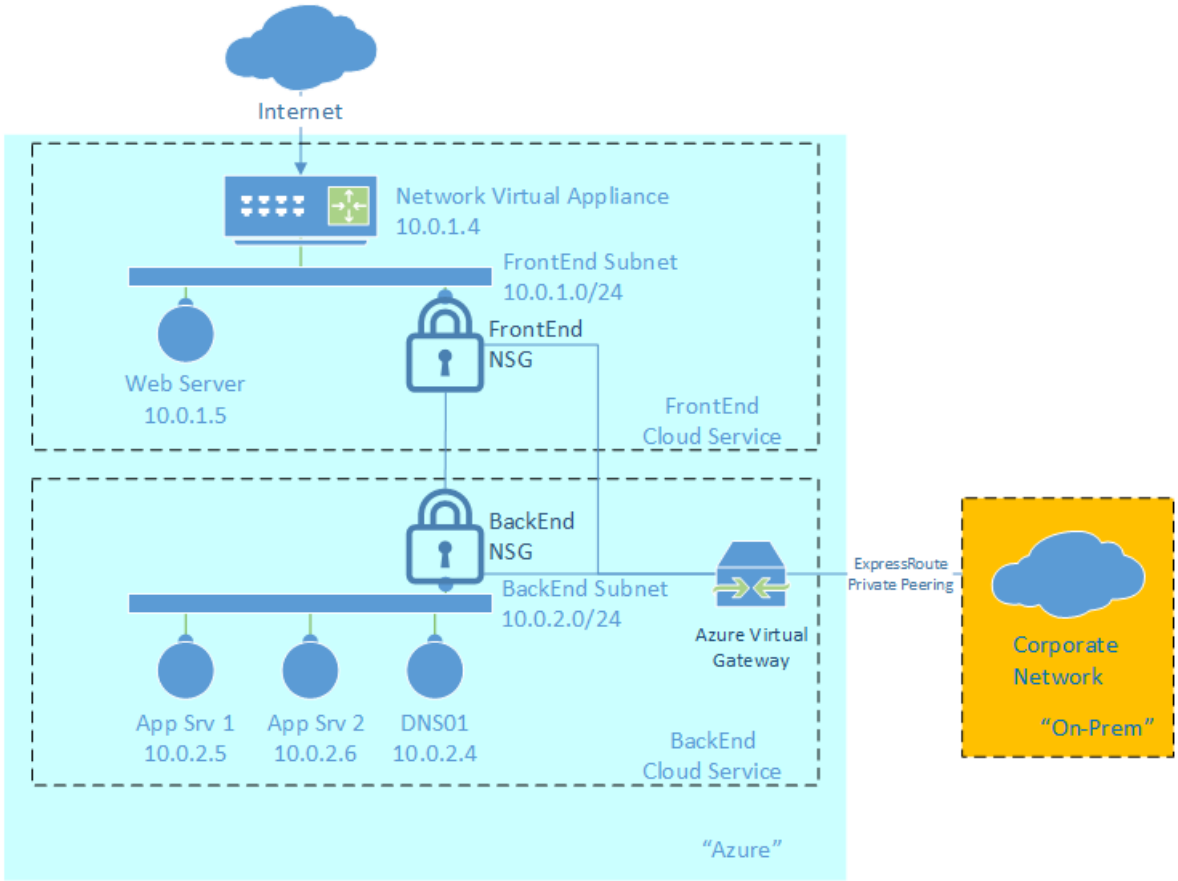
> **NOTE**
>
> There are certain restrictions when using UDR with ExpressRoute, due to the complexity of dynamic routing used in the Azure virtual gateway. These are as follows:
>
> - UDR should not be applied to the gateway subnet on which the ExpressRoute linked Azure virtual gateway is connected.
> - The ExpressRoute linked Azure virtual gateway cannot be the NextHop device for other UDR bound subnets.

> **TIP**
>
> Using ExpressRoute keeps corporate network traffic off of the Internet for better security and significantly increased performance. It also allows for service level agreements from your ExpressRoute provider. The Azure Gateway can pass up to 2 Gb/s with ExpressRoute, whereas with site-to-site VPNs, the Azure Gateway maximum throughput is 200 Mb/s.

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 completely 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:
- Accessing Azure with PowerShell: https://azure.microsoft.com/documentation/articles/powershell-install-configure/
- Virtual networking documentation: https://azure.microsoft.com/documentation/services/virtual-network/
- Network security group documentation: https://azure.microsoft.com/documentation/articles/virtual-networks-nsg/
- User defined routing documentation: https://azure.microsoft.com/documentation/articles/virtual-networks-udr-overview/
- Azure virtual gateways: https://azure.microsoft.com/documentation/services/vpn-gateway/
- Site-to-Site VPNs: https://azure.microsoft.com/documentation/articles/vpn-gateway-create-site-to-site-rm-powershell
- ExpressRoute documentation (be sure to check out the “Getting Started” and “How To” sections): https://azure.microsoft.com/documentation/services/expressroute/

Implementing a DMZ between Azure and your on-premises datacenter

11/15/2016 • 12 min to read • Edit on GitHub

Contributors

REDMOND\telmos • Theano Petersen • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Joseph Molnar • Mike Wasson • Yuri Diogenes • Cody Mansfield • victorar • Paulo Teixeira • JohnPWSharp

patterns & practices
proven practices for predictable results

This article describes best practices for implementing a secure hybrid network that extends an on-premises network to Azure. This reference architecture implements a DMZ between an on-premises network and an Azure virtual network using user defined routes (UDRs). The DMZ includes highly available network virtual appliances (NVAs) that implement security functionality such as firewalls and packet inspection. All outgoing traffic from the VNet is force-tunneled to the Internet through the on-premises network so it can be audited.

This architecture requires a connection to your on-premises datacenter implemented using either a VPN gateway, or an ExpressRoute connection.

> **NOTE**
> Azure has two different deployment models: Resource Manager and classic. This reference architecture uses Resource Manager, which Microsoft recommends for new deployments.

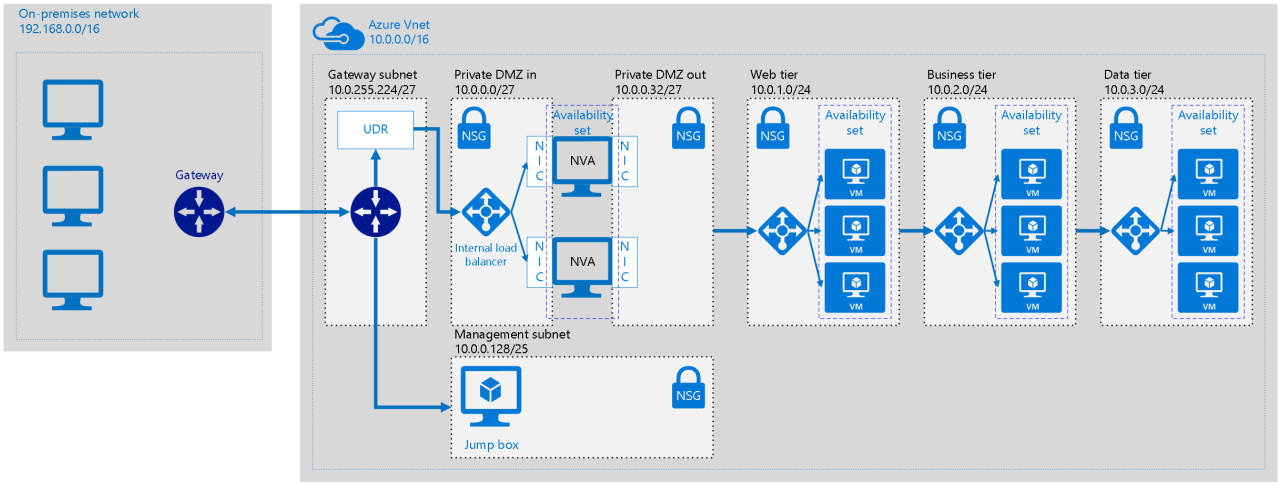
Typical use cases for this architecture include:

- Hybrid applications where workloads run partly on-premises and partly in Azure.
- Azure infrastructure that routes incoming traffic from on-premises and the internet.
- Applications required to audit outgoing traffic. This is often a regulatory requirement of many commercial systems and can help to prevent public disclosure of private information.

Architecture diagram

The following diagram highlights the important components in this architecture:

> A Visio document that includes this architecture diagram is available for download at the Microsoft download center. This diagram is on the "DMZ - Private" page.



- **On-premises network.** This is a network of computers and devices connected through a private local-area network implemented in an organization.
- **Azure virtual network (VNet).** The VNet hosts the application and other resources running in the cloud.
- **Gateway.** The gateway provides connectivity between the routers in the on-premises network and the VNet.
- **Network virtual appliance (NVA).** NVA is a generic term that describes a VM performing tasks such as allowing or denying access as a firewall, optimizing WAN operations (including network compression), custom routing, or other network functionality.
- **Web tier, business tier, and data tier subnets.** These are subnets hosting the VMs and services that implement an example 3-tier application running in the cloud. See Running Windows VMs for an N-tier architecture on Azure for more information.
- **User defined routes (UDR).** User defined routes define the flow of IP traffic within Azure VNets.

> **NOTE**
> Depending on the requirements of your VPN connection, you can configure Border Gateway Protocol (BGP) routes as an alternative to to using UDRs to implement the forwarding rules that direct traffic back through the on-premises network.

- **Management subnet.** This subnet contains VMs that implement management and monitoring capabilities for the components running in the VNet.

Recommendations

Azure offers many different resources and resource types, so this reference architecture can be provisioned many different ways. We have provided an Azure Resource Manager template to install the reference architecture that follows these recommendations. If you choose to create your own reference architecture you should follow these recommendations unless you have a specific requirement that a recommendation does not fit.

RBAC recommendations

Create several RBAC roles to manage the resources in your application. Consider creating a DevOps custom role with permissions to administer the infrastructure for the application. Consider creating a centralized IT administrator custom role to manage network resources, and a separate security IT administrator custom role to manage secure network resources such as the NVAs.

The DevOps role should include permissions to deploy the application components as well as monitor and restart VMs. The centralized IT administrator role should include permissions to monitor network resources. Neither of these roles should have access to the NVA resources as this should be restricted to the security IT administrator role.

Resource group recommendations

Azure resources such as VMs, VNets, and load balancers can be easily managed by grouping them together into resource groups. You can then assign the roles above to each resource group to restrict access.

We recommend the creation of the following:

- A resource group containing the subnets (excluding the VMs), NSGs, and the gateway resources for connecting to the on-premises network. Assign the centralized IT administrator role to this resource group.
- A resource group containing the VMs for the NVAs (including the load balancer), the jump box and other management VMs, and the UDR for the gateway subnet that forces all traffic through the NVAs. Assign the security IT administrator role to this resource group.
- Separate resource groups for each application tier that contain the load balancer and VMs. Note that this resource group shouldn't include the subnets for each tier. Assign the DevOps role to this resource group.

Virtual network gateway recommendations

On-premises traffic passes to the VNet through a virtual network gateway. We recommend an Azure VPN gateway or an Azure ExpressRoute gateway.

NVA recommendations

NVAs provide different services for managing and monitoring network traffic. The Azure Marketplace offers several third-party vendor NVAs, including:

- Barracuda Web Application Firewall and Barracuda NextGen Firewall
- Cohesive Networks VNS3 Firewall/Router/VPN
- Fortinet FortiGate-VM
- SecureSphere Web Application Firewall
- DenyAll Web Application Firewall
- Check Point vSEC

If none of these third-party NVAs meet your requirements, you can create a custom NVA using VMs. For an example of creating custom NVAs, see the DMZ in this reference architecture that implements the following functionality:

- Traffic is routed using IP forwarding on the NVA NICs.
- Traffic is permitted to pass through the NVA only if it is appropriate to do so. Each NVA VM in the reference architecture is a simple Linux router with inbound traffic arriving on network interface *eth0*, and outbound traffic matching rules defined by custom scripts dispatched through network interface *eth1*.
- Traffic routed to the management subnet does not pass through the NVAs and the NVAs can only be configured from the management subnet. If traffic to the management subnet is required to be routed through the NVAs, there is no route to the management subnet to fix the NVAs if they should fail.
- The VMs for the NVA are included in an availability set behind a load balancer. The UDR in the gateway subnet directs NVA requests to the load balancer.

Another recommendation to consider is connecting multiple NVAs in series with each NVA performing a specialized security task. This allows each security function to be managed on a per-NVA basis. For example, an NVA implementing a firewall could be placed in series with an NVA running identity services. The tradeoff for ease of management is the addition of extra network hops that may increase latency, so ensure that this doesn't affect your application's performance.

NSG recommendations

The VPN gateway exposes a public IP address for the connection to the on-premises network. We recommend creating a network security group (NSG) for the inbound NVA subnet implementing rules to block all traffic not originating from the on-premises network.

We also recommend that you implement NSGs for each subnet to provide a second level of protection against

inbound traffic bypassing an incorrectly configured or disabled NVA. For example, the web tier subnet in the reference architecture implements an NSG with a rule to ignore all requests other than those received from the on-premises network (192.168.0.0/16) or the VNet, and another rule that ignores all requests not made on port 80.

Internet access recommendations

Force-tunnel all outbound internet traffic through your on-premises network using the site-to-site VPN tunnel and route to the internet using network address tranlation (NAT). This will both prevent accidental leakage of any confidential information stored in your data tier and also allow inspection and auditing of all outgoing traffic.

> **NOTE**
> Don't completely block Internet traffic from the web, business and application tiers. If these tiers use Azure PaaS services they rely on public IP addresses for VM diagnostics logging, download of VM extensions, and other functionality. Azure diagnostics also requires that components can read and write to an internet-dependent Azure storage account.

We further recommend that you verify outbound internet traffic is force-tunneled correctly. If you're using a VPN connection with the routing and remote access service on an on-premises server, use a tool such as WireShark or Microsoft Message Analyzer.

Management subnet recommendations

The management subnet contains a jump box that executes management and monitoring functionality. Implement the following recommendations for the jump box:

- Do not create a public IP address for the jump box.
- Create one route to access the jump box through the incoming gateway and implement an NSG in the management subnet to only respond to requests from the allowed route.
- Restrict execution of all secure management tasks to the jump box.

NVA recommendations

Include a layer 7 NVA to terminate application connections at the NVA level and maintain affinity with the backend tiers. This guarantees symmetric connectivity in which response traffic from the backend tiers returns through the NVA.

Scalability considerations

The reference architecture implements a load balancer directing on-premises network traffic to a pool of NVA devices. As discussed earlier, the NVA devices are VMs executing network traffic routing rules and are deployed into an availability set. This design allows you to monitor the throughput of the NVAs over time and add NVA devices in response to increases in load.

The standard SKU VPN gateway supports sustained throughput of up to 100 Mbps. The High Performance SKU provides up to 200 Mbps. For higher bandwidths, consider upgrading to an ExpressRoute gateway. ExpressRoute provides up to 10 Gbps bandwidth with lower latency than a VPN connection.

> **NOTE**
> The articles Implementing a Hybrid Network Architecture with Azure and On-premises VPN and Implementing a hybrid network architecture with Azure ExpressRoute describe issues surrounding the scalability of Azure gateways.

Availability considerations

The reference architecture implements a load balancer distributing requests from on-premises to a pool of NVA devices in Azure. The NVA devices are VMs executing network traffic routing rules and are deployed into an

availability set. The load balancer regularly queries a health probe implemented on each NVA and will remove any unresponsive NVAs from the pool.

If you're using Azure ExpressRoute to provide connectivity between the VNet and on-premises network, configure a VPN gateway to provide failover if the ExpressRoute connection becomes unavailable.

For specific information on maintaining availability for VPN and ExpressRoute connections, see the articles Implementing a Hybrid Network Architecture with Azure and On-premises VPN and Implementing a hybrid network architecture with Azure ExpressRoute.

Manageability considerations

All application and resource monitoring should be performed by the jump box in the management subnet. Depending on your application requirements, you may need to add additional monitoring resources in the management subnet, but again any of these additional resources should be accessed via the jump box.

If gateway connectivity from your on-premises network to Azure is down, you can still reach the jump box by deploying a PIP, adding it to the jump box, and remoting in from the internet.

Each tier's subnet in the reference architecture is protected by NSG rules, and it may be necessary to create a rule to open port 3389 for RDP access on Windows VMs or port 22 for SSH access on Linux VMs. Other management and monitoring tools may require rules to open additional ports.

If you're using ExpressRoute to provide the connectivity between your on-premises datacenter and Azure, use the Azure Connectivity Toolkit (AzureCT) to monitor and troubleshoot connection issues.

> **NOTE**
> You can find additional information specifically aimed at monitoring and managing VPN and ExpressRoute connections in the articles Implementing a Hybrid Network Architecture with Azure and On-premises VPN and Implementing a hybrid network architecture with Azure ExpressRoute.

Security considerations

This reference architecture implements multiple levels of security:

Routing all on-premises user requests through the NVA

The UDR in the gateway subnet blocks all user requests other than those received from on-premises. The UDR passes allowed requests to the NVAs in the private DMZ subnet, and these requests are passed on to the application if they are allowed by the NVA rules. Other routes can be added to the UDR, but ensure they don't inadvertently bypass the NVAs or block administrative traffic intended for the management subnet.

The load balancer in front of the NVAs also acts as a security device by ignoring traffic on ports that are not open in the load balancing rules. The load balancers in the reference architecture only listen for HTTP requests on port 80 and HTTPS requests on port 443. Any additional rules added to the load balancers must be documented and the traffic should be monitored to ensure there are no security issues.

Using NSGs to block/pass traffic between application tiers

Each of the web, business, and data tiers restrict traffic between them using NSGs. That is, the business tier uses an NSG to block all traffic that doesn't originate in the web tier, and the data tier uses an NSG to block all traffic that doesn't originate in the business tier. If you have a requirement to expand the NSG rules to allow broader access to these tiers, weigh these requirements against the security risks. Each new inbound pathway represents an opportunity for accidental or purposeful data leakage or application damage.

DevOps access

Restrict the operations that DevOps can perform on each tier using RBAC to manage permissions. When granting permissions, use the principle of least privilege. Log all administrative operations and perform regular audits to ensure any configuration changes were planned.

> **NOTE**
>
> For more extensive information, examples, and scenarios about managing network security with Azure, see Microsoft cloud services and network security. For detailed information about protecting resources in the cloud, see Getting started with Microsoft Azure security. For additional details on addressing security concerns across an Azure gateway connection, see Implementing a Hybrid Network Architecture with Azure and On-premises VPN and Implementing a hybrid network architecture with Azure ExpressRoute.

Solution deployment

A deployment for a reference architecture that implements these recommendations is available on Github. This reference architecture includes a virtual network (VNet), network security group (NSG), load balancer, and two virtual machines (VMs).

The reference architecture can be deployed either with Windows or Linux VMs by following the directions below:

1. Right click the button below and select either "Open link in new tab" or "Open link in new window":

2. Once the link has opened in the Azure portal, you must enter values for some of the settings:

   - The **Resource group** name is already defined in the parameter file, so select **Create New** and enter `ra-private-dmz-rg` in the text box.
   - Select the region from the **Location** drop down box.
   - Do not edit the **Template Root Uri** or the **Parameter Root Uri** text boxes.
   - Review the terms and conditions, then click the **I agree to the terms and conditions stated above** checkbox.
   - Click on the **Purchase** button.

3. Wait for the deployment to complete.
4. The parameter files include hard-coded administrator user name and password for all VMs, and it is strongly recommended that you immediately change both. For each VM in the deployment, select it in the Azure portal and then click on **Reset password** in the **Support + troubleshooting** blade. Select **Reset password** in the **Mode** dropdown box, then select a new **User name** and **Password**. Click the **Update** button to persist.

Next steps

- Learn how to implement a DMZ between Azure and the Internet.
- Learn how to implement a highly available hybrid network architecture.

Azure storage security overview

11/15/2016 • 4 min to read • Edit on GitHub

Contributors

TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Matthew Baldwin

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:

- The storage account can be secured using Role-Based Access Control and Azure Active Directory.
- Data can be secured in transit between an application and Azure by using Client-Side Encryption, HTTPS, or SMB 3.0.
- Data can be set to be automatically encrypted when written to Azure Storage using Storage Service Encryption.
- OS and Data disks used by virtual machines can be set to be encrypted using Azure Disk Encryption.
- Delegated access to the data objects in Azure Storage can be granted using Shared Access Signatures.
- The authentication method used by someone when they access storage can be tracked using Storage analytics.

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 such as storage account keys, data encryption in transit and at rest, and storage analytics.

This article provides an overview of Azure security features that can be used with Azure Storage. Links are provided to articles that give details of each feature so you can learn more.

Here are the core features to be covered in this article:

- Role-Based Access Control
- Delegated access to storage objects
- Encryption in transit
- Encryption at rest/Storage Service Encryption
- Azure Disk Encryption
- Azure Key Vault

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.

Learn more:

- Azure Active Directory Role-based Access Control

Delegated access to storage objects

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 of time 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 pass in the SAS to the appropriate constructor or method.

Learn more:

- Understanding the SAS model
- Create and use a SAS with Blob storage

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.

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 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.

Learn more about Storage Service Encryption:

- Azure Storage Service Encryption is available for Azure Blob Storage. For details on other Azure storage types, see File, Disk (Premium Storage), Table, and Queue.
- Azure Storage Service Encryption for Data at Rest

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.

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

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, while ensuring that all data in the virtual machine disks are encrypted at rest in your 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 Data Security and Encryption Best Practices

11/15/2016 • 11 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear

---

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 the purpose of Azure data security and encryption best practices the recommendations will 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.

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.

Azure data security and encryption best practices discussed in this article include:

- Enforce multi-factor authentication
- Use role based access control (RBAC)
- Encrypt Azure virtual machines
- Use hardware security models
- Manage with Secure Workstations
- Enable SQL data encryption
- Protect data in transit
- Enforce file level data encryption

Enforce Multi-factor Authentication

The first step in data access and control in Microsoft Azure is to authenticate the user. Azure Multi-Factor Authentication (MFA) is a method of verifying user's identity by using another method than just a username and password. This authentication method helps safeguard access to data and applications while meeting user demand for a simple sign-in process.

By enabling Azure MFA for your users, you are adding a second layer of security to user sign-ins and transactions. In this case, a transaction might be accessing a document located in a file server or in your SharePoint Online. Azure MFA also helps IT to reduce the likelihood that a compromised credential will have access to organization's data.

For example: if you enforce Azure MFA for your users and configure it to use a phone call or text message as verification, if the user's credential is 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 this extra layer 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 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.

For more information on Azure MFA, please read the article Getting started with Azure Multi-Factor Authentication in the cloud.

Use Role Based Access Control (RBAC)

Restrict access based on the need to know and least privilege security principles. This is imperative for organizations that want to enforce security policies for data access. Azure Role-Based Access Control (RBAC) can be used 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 leverage built-in RBAC roles in Azure to assign privileges to users. Consider using *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 needs to manage VMs and storage account, consider adding them to *Virtual Machine Contributor* role.

Organizations that do not enforce data access control by leveraging capabilities such as RBAC may be giving more privileges than necessary for their users. This can lead to data compromise by having some users having access to data that they shouldn't have in the first place.

You can learn more about Azure RBAC by reading the article Azure Role-Based Access Control.

Encrypt Azure Virtual Machines

For many organizations, data encryption at rest is a mandatory step towards data privacy, compliance and data sovereignty. Azure Disk Encryption enables IT administrators to encrypt Windows and Linux IaaS Virtual Machine (VM) disks. Azure 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 the data disks.

You can leverage Azure Disk Encryption to help protect and safeguard your data to meet your organizational security and compliance requirements. Organizations should also consider using encryption to help mitigate risks related to unauthorized data access. It is also recommended that you encrypt drives prior to writing sensitive data to them.

Make sure to encrypt your VM's data volumes and boot volume in order to protect data at rest in your Azure storage account. Safeguard the encryption keys and secrets by leveraging Azure Key Vault.

For your on-premises Windows Servers, consider the following encryption best practices:

- Use BitLocker for data encryption
- Store recovery information in AD DS.
- If there is any concern that BitLocker keys have been compromised, we recommend that you either format the drive to remove all instances of the BitLocker metadata from the drive or that you decrypt and encrypt the entire drive again.

Organizations that do not enforce data encryption are more likely to be exposed to data integrity issues, such as malicious or rogue users stealing data and compromised accounts gaining unauthorized access to data in clear format. Besides these risks, companies that have to comply with industry regulations, must prove that they are diligent and are using the correct security controls to enhance data security.

You can learn more about Azure Disk Encryption by reading the article Azure Disk Encryption for Windows and Linux IaaS VMs.

Use Hardware Security Modules

Industry encryption solutions use secret keys to encrypt data. Therefore, it is critical that these keys are safely stored. Key management becomes an integral part of data protection, since it will be leveraged to store secret keys that are used to encrypt data.

Azure disk encryption uses Azure Key Vault to help you control and manage disk encryption keys and secrets in your key vault subscription, while ensuring that all data in the virtual machine disks are encrypted at rest in your Azure storage. You should use Azure Key Vault to audit keys and policy usage.

There are many inherent risks related to not having appropriate security controls in place to protect the secret keys that were used to encrypt your data. If attackers have access to the secret keys, they will be able to decrypt the data and potentially have access to confidential information.

You can learn more about general recommendations for certificate management in Azure by reading the article Certificate Management in Azure: Do's and Don'ts.

For more information about Azure Key Vault, read Get started with Azure Key Vault.

Manage with Secure Workstations

Since the vast majority of the attacks target the end user, the endpoint becomes one of the primary points of attack. If an attacker compromises the endpoint, he can leverage the user's credentials to gain access to organization's data. Most endpoint attacks are able to take advantage of the fact that end users are administrators in their local workstations.

You can reduce these risks by using a secure management workstation. We recommend that you use a Privileged Access Workstations (PAW) to reduce the attack surface in workstations. These secure management workstations can help you mitigate some of these attacks help ensure your data is safer. Make sure to use PAW to harden and lock down your workstation. This is an important step to provide high security assurances for sensitive accounts, tasks and data protection.

Lack of endpoint protection may put your data at risk, make sure to enforce security policies across all devices that are used to consume data, regardless of the data location (cloud or on-premises).

You can learn more about privileged access workstation by reading the article Securing Privileged Access.

Enable SQL data encryption

Azure SQL Database transparent data encryption (TDE) helps protect against the threat of malicious activity by performing real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application. TDE 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 is very important to also encrypt your database itself. This is an implementation of the defense in depth approach for data protection. If you are using Azure SQL Database and wish 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 which meets the requirements of many industry standards (e.g., HIPAA,

PCI).

It's important to understand that files related to buffer pool extension (BPE) are not encrypted when a database is encrypted using TDE. You must use file system level encryption tools like BitLocker or the Encrypting File System (EFS) for BPE related files.

Since an authorized user such as a security administrator or a database administrator can access the data even if the database is encrypted with TDE, you should also follow the recommendations below:

- SQL authentication at the database level
- Azure AD authentication using RBAC roles
- Users and applications should use separate accounts to authenticate. This way you can limit the permissions granted to users and applications and reduce the risks 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

Organizations that are not using database level encryption may be more susceptible for attacks that may compromise data located in SQL databases.

You can learn more about SQL TDE encryption by reading the article Transparent Data Encryption with Azure SQL Database.

Protect data in transit

Protecting data in transit should be essential part of your data protection strategy. Since data will be 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

Another layer of protection that can increase the level of security for your data is encrypting the file itself, regardless of the file location.

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.

You can learn more about Azure RMS by reading the article Getting Started with Azure Rights Management.

Azure Storage security guide

11/22/2016 • 43 min to read • Edit on GitHub

Contributors

Robin Shahan • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • tfitzmac • Tamra Myers • Ralph Squillace • Yuri Diogenes • Natalia

Overview

Azure Storage provides a comprehensive set of security capabilities which together enable developers to build secure applications. The storage account itself can be secured using Role-Based Access Control and Azure Active Directory. Data can be secured in transit between an application and Azure by using Client-Side Encryption, HTTPS, or SMB 3.0. Data can be set to be automatically encrypted when written to Azure Storage using Storage Service Encryption (SSE). OS and Data disks used by virtual machines can be set to be encrypted using Azure Disk Encryption. Delegated access to the data objects in Azure Storage can be granted using Shared Access Signatures.

This article will provide 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. In this section, we'll talk about the Azure Resource Manager deployment model and how to use Role-Based Access Control (RBAC) to control access to your storage accounts. We will also talk about 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), and how you can enable it for a storage account, resulting in your block blobs, page blobs, and append blobs being automatically encrypted when written to Azure Storage. 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 which 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 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:

- 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 has to 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 is the reference for the APIs you can use to manage your storage account programmatically.

- Developer's guide to auth with Azure Resource Manager API

  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, e.g. blobs, entities within a table, queue messages, and files on an Azure Files 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 re-sync 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. Note that 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: it is recommended to use 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

- About Azure Storage Accounts

  This article gives an overview of storage accounts and discusses viewing, copying, and regenerating storage access keys.

- Azure Storage Resource Provider REST API Reference

  This article contains links to specific articles about retrieving the storage account keys and regenerating the storage account keys for an Azure Account using the REST API. Note: This is for Resource Manager storage accounts.

- Operations on storage accounts

  This article in the Storage Service Manager REST API Reference contains links to specific articles on retrieving and regenerating the storage account keys using the REST API. Note: This is for the Classic storage accounts.

- Say goodbye to key management – manage access to Azure Storage data using Azure AD

  This article shows how to use Active Directory to control access to your Azure Storage keys in Azure Key Vault. It also shows how to use an Azure Automation job to regenerate the keys on an hourly basis.

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 allowing access to the objects?

There are basically two methods for controlling access to the data objects themselves. The first is by controlling access to the storage account keys, and the second is using Shared Access Signatures to grant access to specific data objects for a specific amount of time.

One exception to note is that 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.

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%2FRHIX5Xcg0Mq2rqI3OlWTjEg2tYkboXr1P9ZUXDtkk%3D (signature used for the authentication of the SAS)
```

How the Shared Access Signature is authenticated 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 an SAS URI

1. You can create an ad hoc URI on demand, defining all of the query parameters each time.

   This is really 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 5 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 URI's, you have three options. You can issue SAS tokens with short expiration policies and simply 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 big 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, please 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.
- Limiting access by IP Address (IP ACLs)
  - What is an endpoint Access Control List (ACLs)?
  - Constructing a Service SAS

    This is the reference article for service-level SAS; it includes an example of IP ACLing.
  - Constructing an Account SAS

    This is the reference article for account-level SAS; it includes an example of IP ACLing.
- 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.

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.

Resources

- Enable HTTPS for an app in Azure App Service

  This article shows you how to enable HTTPS for an Azure Web App.

Using encryption during transit with Azure File Shares

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.

Note that while Azure File Shares can be used with Unix, the Linux SMB client does not yet support encryption, so access is only allowed within an Azure region. Encryption support for Linux is on the roadmap of Linux developers

responsible for SMB functionality. When they add encryption, you will have the same ability for accessing an Azure File Share on Linux as you do for Windows.

Resources

- How to use Azure File Storage with Linux

  This article shows how to mount an Azure File Share on a Linux system and upload/download files.

- Get started with Azure File storage on Windows

  This article gives an overview of Azure File shares and how to mount and use them using PowerShell and .NET.

- Inside Azure File Storage

  This article announces the general availability of Azure File Storage and provides technical details about the SMB 3.0 encryption.

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. The other two – Client-side Encryption and SSE – are both used to encrypt data in Azure Storage. Let's look at each of these, and then do a comparison and see when each one can be used.

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 simply 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 allows you to request that the storage service automatically encrypt the data when writing it to Azure Storage. When you read the data from Azure Storage, it will be decrypted by the storage service before being returned. This enables you to secure your data without having to modify code or add code to any applications.

This is a setting that applies to the whole storage account. You can enable and disable this feature by changing the value of the setting. To do this, you can use the Azure portal, PowerShell, the Azure CLI, the Storage Resource Provider REST API, or the .NET Storage Client Library. By default, SSE is turned off.

At this time, the keys used for the encryption are managed by Microsoft. We generate 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, you will get the ability to manage your own encryption keys, and provide a migration path from Microsoft-managed keys to customer-managed keys.

This feature is available for Standard and Premium Storage accounts created using the Resource Manager deployment model. 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.

Data is only encrypted when SSE is enabled and the data is written to Blob Storage. Enabling or disabling SSE does not impact existing data. In other words, when you enable this encryption, it will not go back and encrypt data that already exists; nor will it decrypt the data that already exists when you disable SSE.

If you want to use this feature with a Classic storage account, you can create a new Resource Manager storage account and use AzCopy to copy the data to the new account.

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. Note that although the data is encrypted in transit, we still recommend using HTTPS to take advantage of the built-in data integrity checks which 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 pretty 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 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:

- 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.

> **NOTE**
>
> For Linux VMs already running in Azure, or new Linux VMs created from images in the Azure Marketplace, encryption of the OS disk is not currently supported. Encryption of the OS Volume for Linux VMs is supported only for VMs that were encrypted on-premises and uploaded to Azure. This restriction only applies to the OS disk; encryption of data volumes for a Linux VM is supported.

The solution supports the following for IaaS VMs for public preview release when enabled in Microsoft Azure:

- Integration with Azure Key Vault
- Standard A, D and G series IaaS VMs
- Enable encryption on IaaS VMs created using Azure Resource Manager model
- All Azure public regions

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 Virtual Machines

  This article discusses the preview release of Azure Disk Encryption and provides a link to download the white paper.

Comparison of Azure Disk Encryption, SSE, and Client-Side Encryption

IaaS VMs and their VHD files

For disks used by IaaS VMs, we recommend using Azure Disk Encryption. You can turn on SSE to encrypt the VHD files that are used to back those disks in Azure Storage, but it only encrypts newly written data. This means if you create a VM and then enable SSE on the storage account that holds the VHD file, only the changes will be encrypted, not the original VHD file.

If you create a VM 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 created from images in the Azure Marketplace if you want them fully encrypted.

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 service management 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 it before transit, and encrypts the data at rest. 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 for your data in transit, and SSE to encrypt the data at rest.

With client-side encryption, you can encrypt table entities, queue messages, and blobs. With SSE, you can only encrypt blobs. If you need table and queue data to be encrypted, you should use client-side encryption.

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 lot of data.

Storage Service Encryption (SSE)

SSE is managed by Azure Storage. Using 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. There is no impact on the performance when using this feature.

You can only encrypt block blobs, append blobs, and page blobs using SSE. If you need to encrypt table data or queue data, you should consider using client-side encryption.

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, enable SSE, 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 and SSE enabled on the storage account holding the VHD files, it will work fine; it will result in any newly-written data being encrypted twice.

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.

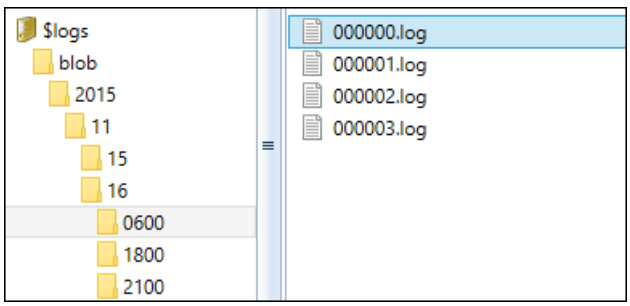
This can be really helpful if you are tightly guarding access to storage. For example, in Blob Storage you can set all of the containers to private and implement the use of an SAS service throughout your applications. Then you can check the logs regularly to see if your blobs are accessed using the storage account keys, which may indicate a breach of security, or if the blobs are public but they shouldn't be.

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 simply 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:
1.0;2015-11-16T06:13:27.2588724Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https:
1.0;2015-11-16T06:14:28.0166751Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https:
1.0;2015-11-16T06:14:29.2558837Z;GetBlobServiceProperties;Success;200;3;3;authenticated;mystorage;mystorage;blob;"https:
1.0;2015-11-16T06:14:43.4307865Z;BlobPreflightRequest;AnonymousSuccess;200;2;2;anonymous;;mystorage;blob;"https://mystor
1.0;2015-11-16T06:14:43.4528051Z;GetBlobServiceProperties;Success;200;1;1;authenticated;mystorage;mystorage;blob;"https:
1.0;2015-11-16T06:15:30.3567270Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https:
1.0;2015-11-16T06:15:29.2735098Z;GetBlobServiceProperties;Success;200;5;5;authenticated;mystorage;mystorage;blob;"https:
1.0;2015-11-16T06:16:32.9445742Z;GetBlobServiceProperties;Success;200;4;3;authenticated;mystorage;mystorage;blob;"https:
1.0;2015-11-16T06:16:44.2766486Z;ListContainers;Success;200;4;4;authenticated;mystorage;mystorage;blob;"https://mystorag
1.0;2015-11-16T06:16:56.0216743Z;CreateContainer;Success;201;10;10;authenticated;mystorage;mystorage;blob;"https://mysto
1.0;2015-11-16T06:16:56.0517020Z;ListContainers;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystorag
1.0;2015-11-16T06:16:59.9423538Z;ListContainers;Success;200;3;3;authenticated;mystorage;mystorage;blob;"https://mystorag
1.0;2015-11-16T06:16:59.9984102Z;ListBlobs;Success;200;3;3;authenticated;mystorage;mystorage;blob;"https://mystorage.blo
1.0;2015-11-16T06:17:23.7717291Z;GetBlobProperties;ClientOtherError;404;3;3;authenticated;mystorage;mystorage;blob;"http
1.0;2015-11-16T06:17:23.8347867Z;PutBlob;Success;201;71;8;authenticated;mystorage;mystorage;blob;"https://mystorage.blob
1.0;2015-11-16T06:17:23.9549008Z;GetBlobProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mysto
1.0;2015-11-16T06:17:31.9243814Z;GetBlobProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mysto
1.0;2015-11-16T06:17:31.9554107Z;GetBlob;Success;206;81;5;authenticated;mystorage;mystorage;blob;"https://mystorage.blob
1.0;2015-11-16T06:17:46.1437305Z;GetContainerACL;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https://mystora
1.0;2015-11-16T06:16:30.3890982Z;GetBlobServiceProperties;Success;200;2;2;authenticated;mystorage;mystorage;blob;"https:
```

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 authenticated, so we need to look for entries with operation-type "Get-Blob", and check the request-status (4th column) and the authorization-type (8th 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 validated using the storage account key.

How are my blobs being authenticated?

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 "SASSuccess" and the authorization-type is "sas".

   1.0;2015-11-16T18:30:05.6556115Z;GetBlob;**SASSuccess**;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, i.e. 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, quick start, 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, please 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 Virtual Machines security overview

11/15/2016 • 7 min to read • Edit on GitHub

Contributors

TerryLanfear • Ralph Squillace • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes

---

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.

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 our 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. We also provide links to articles that give details of each feature so you can learn more.

The core Azure Virtual Machine security capabilities to be covered in this article:

- Antimalware
- Hardware Security Module
- Virtual machine disk encryption
- Virtual machine backup
- Azure Site Recovery
- Virtual networking
- Security policy management and reporting
- Compliance

Antimalware

With Azure, 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. See the Learn More section below to find articles on partner solutions.

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.

Microsoft Antimalware 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.

When you deploy and enable Microsoft Antimalware, 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 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 to Azure telemetry metadata about detected threats and suspicious resources to ensure rapid response and enables 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 certain files, processes, and drives to exclude them 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.

Learn more: To learn more about antimalware software to protect your virtual machines, see:

- 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
- New Antimalware Options for Protecting Azure Virtual Machines – McAfee Endpoint Protection
- Security solutions in the Azure Marketplace

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.

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 that lets you encrypt your Windows and Linux Azure 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, while ensuring that all data in the virtual machine disks are encrypted at rest in your Azure storage.

Learn more:

- Azure Disk Encryption for Windows and Linux IaaS VMs
- Azure Disk Encryption for Linux and Windows Virtual Machines
- Encrypt a virtual machine

Virtual machine backup

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.

Learn more:

- What is Azure Backup?
- Azure Backup Learning Path
- 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 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.

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** — 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 failback 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 datacenter** — 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 application BCDR features. For example, you can use Site Recovery to protect the SQL Server back end of corporate workloads. This includes native support for SQL Server AlwaysOn 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, 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 helps you optimize and monitor virtual machine security by:

- Providing virtual machine security recommendations such as 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

Security best practices for IaaS workloads in Azure

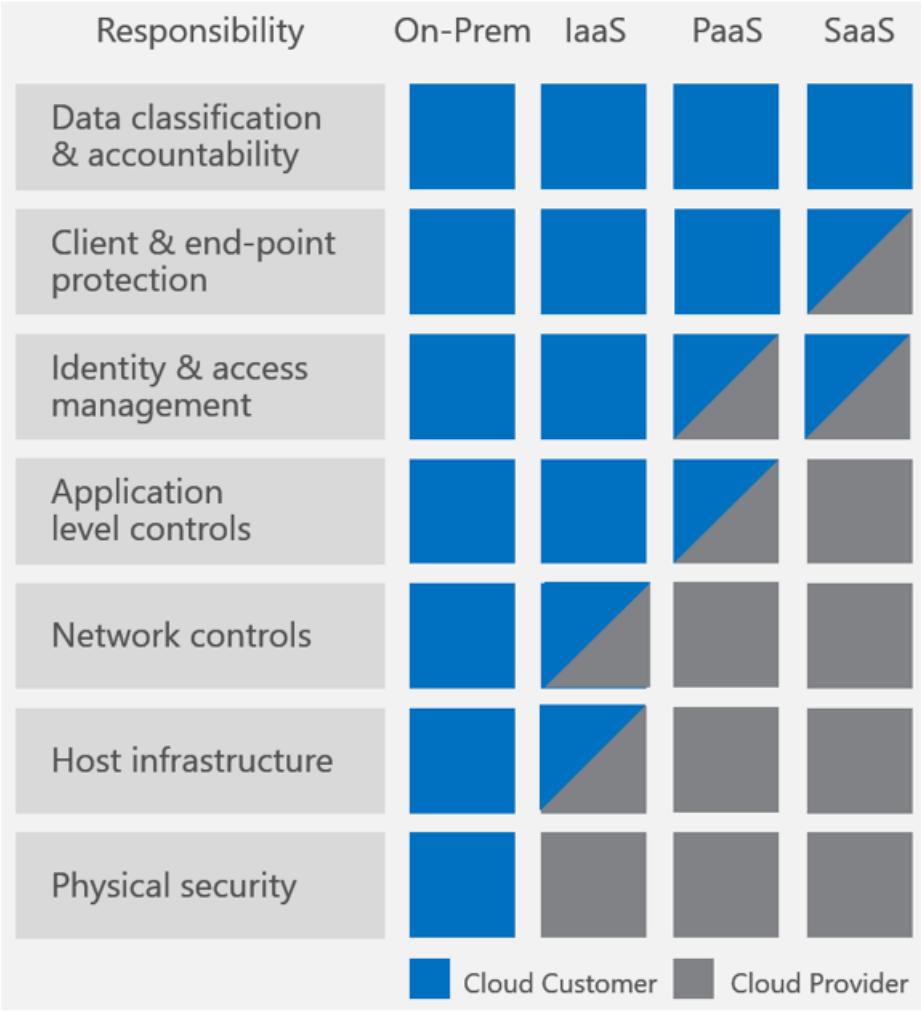
11/22/2016 • 16 min to read • Edit on GitHub

Contributors

Barclay Neira

As you started thinking about moving workloads to Azure IaaS you probably came to the realization that some considerations are familiar. You may already have experience securing virtual environments. The move to Azure IaaS allow you to apply your expertise in securing virtual environments and also brings a new set of options to help you secure your assets.

Before we get going let's start by saying that we should not expect to bring on-premises resources as one-to-one to Azure. The new challenges and the new options bring about an opportunity to reevaluate existing deigns, tools, and processes.



NOTE

Your responsibility for security is based on the type of cloud service. The chart above summarizes the balance of responsibility for both Microsoft and you.

Best practices

We will be discussing some of the options available in Azure that could help you meet your organization's security requirements. While doing this we must keep in mind the different types of workloads and how their security requirements may vary. 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 left by the others.

Use Privileged Access Workstations (PAW)

Organizations often fall prey to cyber-attacks because of administrators performing actions while using accounts with elevated rights. Usually this isn't done maliciously but because existing configuration and processes allow them to do it. Most of these users understand the risk from a conceptual standpoint but still choose to take steps that they would agree are risky.

Doing things like checking email and browsing the Internet seem innocent enough but may expose elevated accounts to compromise by malicious actors who may use browsing activities, specially crafted emails, or other techniques to gain access to your enterprise. The use of secure management workstations for conducting all Azure administration tasks is highly recommended as a way of reducing exposure to accidental compromise.

Privileged Access Workstations (PAWs) provide a dedicated operating system for sensitive tasks that is protected from Internet attacks and threat vectors. Separating these sensitive tasks and accounts from the daily use workstations and devices provides very strong protection from phishing attacks, application and OS vulnerabilities, various impersonation attacks, and credential theft attacks such as keystroke logging, Pass-the-Hash, and Pass-The-Ticket.

The PAW approach is an extension of the well-established recommended practice to use separate admin and user accounts for administrative personnel. This practice uses an individually assigned administrative account that is separate from the user's standard user account. PAW builds on that account separation practice by providing a trustworthy workstation for those sensitive accounts.

For more information on Privileged Access Workstations and guidance for PAW implementation, follow this link:

- Privileged Access Workstations

Use multifactor authentication

In the past, your network perimeter was used to control access to corporate data. In a cloud-first, mobile-first world, identity is the control plane: You use it to control access to IaaS services from any device, and you use it to get visibility and insight into where and how your data is being used. Protecting the digital identity of your Azure users is the cornerstone of protecting your subscriptions from identity theft and other cybercrimes.

One of the most beneficial steps that you can take to secure an account is to enable two factor authentication. Two factor authentication is a way of authenticating by using more than just your password. The second factor is something in addition to the password. This helps mitigate the risk of access by someone who manages to get a hold of someone else's password.

Azure Multi-Factor Authentication helps safeguard access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication with a range of easy verification options—phone call, text message, or mobile app notification—allowing users to choose the method they prefer.

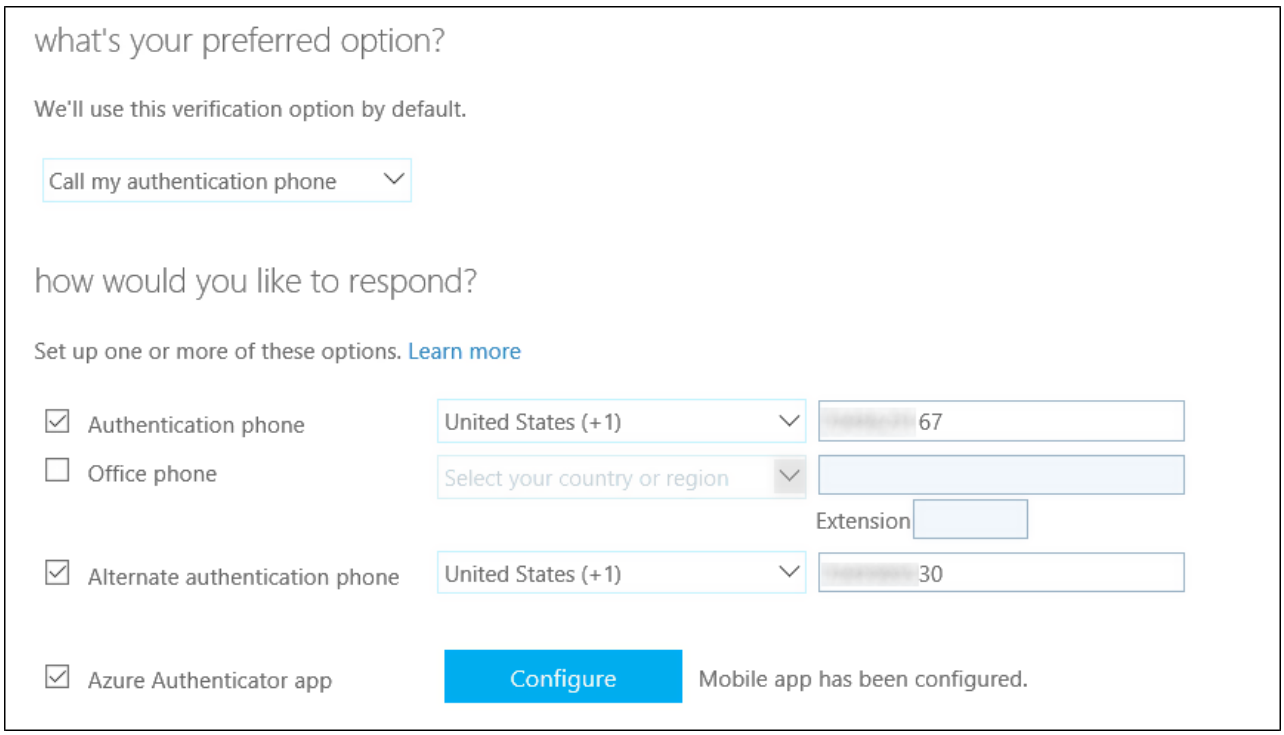
The easiest way to use Azure MFA is the Microsoft Authenticator mobile app that can be used on mobile devices running Windows, IOS and Android. With the latest release of Windows 10 and your on premises Active Directory integrated with Azure AD, Microsoft Hello for Business can be used for seamless single sign-on to Azure resources. In this case, the Windows 10 device will be used as the second factor for authentication.

In the case of Azure, the simplest way to enable two factor authentication is to use Azure Multifactor Authentication (MFA). Azure MFA has an application that can be used on your mobile devices, it can work via phone calls, text messages or via a code generated in the app and it can integrate with your on-premises directory.

For accounts that manage your Azure subscription you should use MFA and for accounts that can logon to the

Virtual Machines you should use MFA when possible. Using MFA for these accounts gives you much greater level of security than just using a password. Using other forms of two factor authentication could work just as well but may be more involved to get deployed if they are not already in production.

The screenshot below shows some of the options available for Azure MFA authentication.



Limit and Constrain Administrative Access

Securing the accounts that can manage your Azure subscription is extremely important. The compromise of any of those accounts effectively negates the value of all the other steps you may take to ensure the confidentiality and integrity of your data. As recently illustrated by the Edward Snowden leak of classified information, internal attacks pose a huge threat to the overall security of any organization.

Individuals who have administrative rights should have been evaluated by following a criteria similar to the one below:

- Are they performing tasks that require administrative privileges?
- How often are the tasks performed?
- Specific reason why the tasks cannot be performed by another administrator on their behalf.
- Document all other known alternative approaches to granting the privilege and why each isn't acceptable.

The use of just in time administration prevents the unnecessary existence of accounts with elevated rights during time periods when those rights are not needed. Accounts have elevated rights for a limited time enabling administrators to do their jobs and then they have those rights removed at the end of a shift or when a task is completed.

PIM allows you to manage, monitor, and control access within your organization. It helps you remain aware of the actions taken by individuals within your organization and brings just-in-time administration to Azure AD by introducing the concept of an eligible admin. These are individuals who have accounts with the potential to be granted admin rights. These types of users can go through an activation process and be granted admin rights for a limited time.

Use DevTest Labs

Using Azure for labs and development environments allows organizations to gain agility in testing and development by taking away the delays introduced by hardware procurement. Unfortunately, there is the risk that a lack of familiarity with Azure or a desire to help expedite its adoption may lead the administrator to be overly

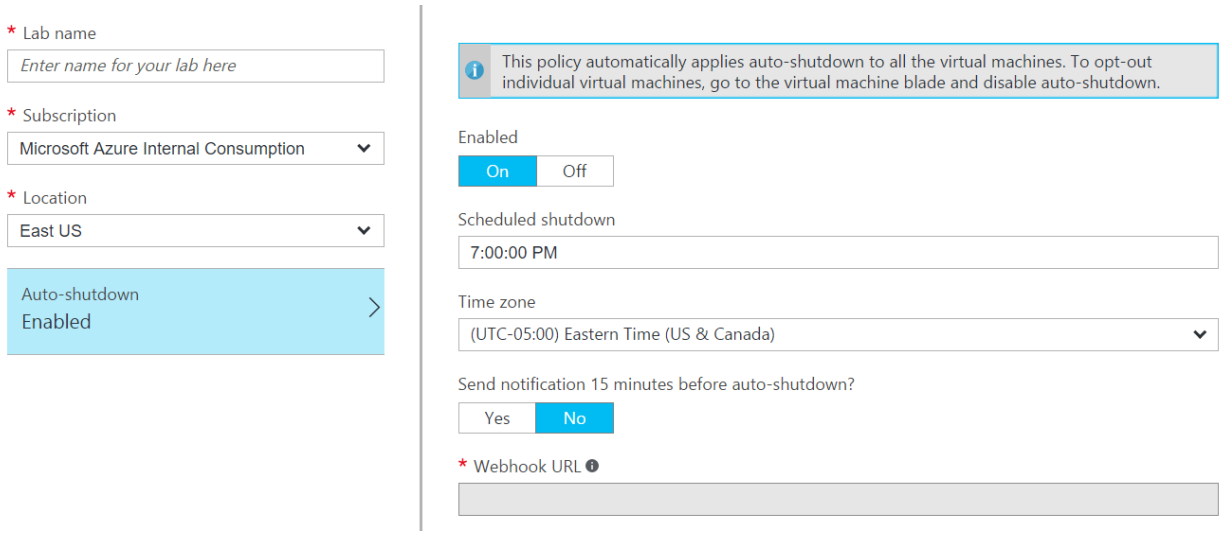
permissive with rights assignment. This may be unintentionally exposing the organization to internal attacks. Some users could be granted a lot more access than they should have.

In Azure we now include a service called DevTest Labs. DevTest Labs uses Azure Role based access control(RBAC)(RBAC). RBAC allows you to segregate duties within your team into roles that grant only the level of access necessary for users to do their jobs. It comes with pre-defined roles (Owner, lab user and contributor). These roles can even be used to assign rights to external partners and greatly simplify collaboration.

Since DevTest Labs uses RBAC, it is possible to create additional custom roles. DevTest Labs not only simplifies the management of permissions, it is also designed to simplify the process of getting environments provisioned and to deal with other typical challenges of teams working on development and test environments. It requires some preparation but in the long term it will make things easier for your team.

Some key Azure DevTest Labs features include:

- Administrative control over the options available to users. Things like allowed VM sizes, maximum number of VMs and when VMs are started and shut down can be centrally managed by the administrator
- Automation of lab environment creation
- Cost tracking
- Simplified distribution of VMs for temporary collaborative work
- Self-service allowing users to provision their labs using templates
- Managing and limiting consumption



There is no additional cost associated with the usage of DevTest Labs. The creation of labs, policy configuration, templates, and artifacts are all free. You only pay for the azure resources used within your labs such as virtual machines, storage accounts and virtual networks.

Control and Limit Endpoint Access

Hosting labs or production systems in Azure means that your systems need to be accessible from the Internet. By default, a new windows virtual machine has the RDP port accessible from the Internet and a Linux virtual machine has the SSH port open. This means that taking steps to 'limit exposed endpoints' is necessary to minimize the risk of unauthorized access.

There are technologies in Azure that can help you limit the access to those administrative endpoints. In Azure you can use Network Security Groups (NSGs). When you use Resource Manager for deployment NSGs are used to limit the access from all networks to just the management endpoints (RDP or SSH). When you think NSGs, think router ACLs. You can use them to tightly control the network communication between various segments of your Azure networks. This is similar to creating networks in DMZs or other isolated networks. They do not inspect the traffic but they will help with network segmentation.

In Azure you can configure a site-to-site VPN from your on-premises network, effectively extending your on-

premises network to the cloud. This would provide you with another opportunity to use NSGs, as you could also modify the NSGs to not allow access from anywhere other than the local network and then require that administration is done by first connecting to the Azure network via VPN.

The site-to-site VPN option may be most attractive in cases where you are hosting production systems that are closely integrated with your on-premises resources in Azure.

Alternatively, the point to site option could be used in situations where you want to manage systems that don't need access to on-premises resources. Those systems can be isolated in their own Azure Virtual Network and administrators could VPN into the Azure hosted environment from their administrative workstation.

> **NOTE**
> Either VPN option would allow you to reconfigure the ACLs on the NSGs to not allow access to management endpoints from the Internet.

Another option worth considering is a Remote Desktop Gateway deployment. You could use this remote desktop gateway deployment to securely connect over HTTPS to remote desktop servers while applying more granular controls to those connections.

Some of the features that you would have access to include:

- Administrator options to limit connections to requests from specific systems.
- Smartcard authentication or Azure MFA
- Control over which systems someone can connect to via the gateway.
- Control over device and disk redirection.

Use a Key Management solution

Secure key management is essential to protecting data in the cloud. With Azure Key Vault, you can securely store encryption keys and small secrets like passwords in hardware security modules (HSMs). For added assurance, you can import or generate keys in HSMs.

If you choose to do this, Microsoft will process your keys in FIPS 140-2 Level 2 validated HSMs (hardware and firmware). Monitor and audit key use with Azure logging—pipe logs into Azure applying or your SIEM for additional analysis and threat detection.

Anyone with an Azure subscription can create and use key vaults. Although Key Vault benefits developers and security administrators, it could be implemented and managed by an organization's administrator responsible for managing Azure services.

Encrypt Virtual Disks and Disk Storage

Azure Disk Encryption addresses the threat of data theft or exposure from unauthorized access achieved by moving a disk. The disk could be attached to another system as a way of bypassing other security controls. Disk encryption uses BitLocker in windows and DM-Crypt in Linux to encrypt operating system and data drives. Azure Disk Encryption integrates with key vault to control and manage the encryption keys and it is available for standard VMs and VMs with premium storage.

For more information, look at the article covering Azure Disk Encryption in Windows and Linux IaaS VMs.

Azure Storage Service Encryption protects your data at rest. It is enabled at the storage account level and it encrypts data as its written in our datacenters and it is automatically decrypted as you access it. It supports the following scenarios:

- Encryption of block blobs, append blobs, and page blobs.
- Encryption of archived VHDs and templates brought to Azure from on-premises.
- Encryption of underlying OS and data disks for IaaS VMs created using your VHDs.

Before you proceed with Azure Storage encryption you should be aware of two notable limitations:

- It is not available on classic storage accounts.
- It only encrypts data written after encryption is enabled.

Use a Centralized Security Management System

Your servers need to be monitored for patching, configuration, events, and activities that may be considered security concerns. To address those concerns you can use Security Center and Operations Management Suite Security and Compliance. Both of these options go beyond the configuration within the operating system and also provide monitoring of the configuration of the underlying infrastructure like network configuration and virtual appliance use.

Operating system management best practices

In an IaaS deployment you are still responsible for the management of the systems that you deploy just like any other server or workstation in your environment. This means that patching, hardening, rights assignments and any other activity related to the maintenance of your system is still your responsibility. For systems that are tightly integrated with your on-premises resources you may want to use the same tools and procedures that you are using on-premises for things like anti-virus, anti-malware, patching, and backup.

Hardening All virtual machines in Azure IaaS should be hardened so that they only expose services endpoints that are required for the applications that are installed. For Windows virtual machines, follow the recommendations that are published by Microsoft as baselines for the Security Compliance Manager solution.

Security Compliance Manager - we recently released version 4.0 - This is a free tool that enables you to quickly configure and manage your desktops, traditional datacenter, private and public cloud using Group Policy and System Center Configuration Manager.

SCM provides ready to deploy policies and DCM configuration packs that are tested. These baselines are based on Microsoft Security guide recommendations and industry best practices, allowing you to manage configuration drift, address compliance requirements, and reduce security threats.

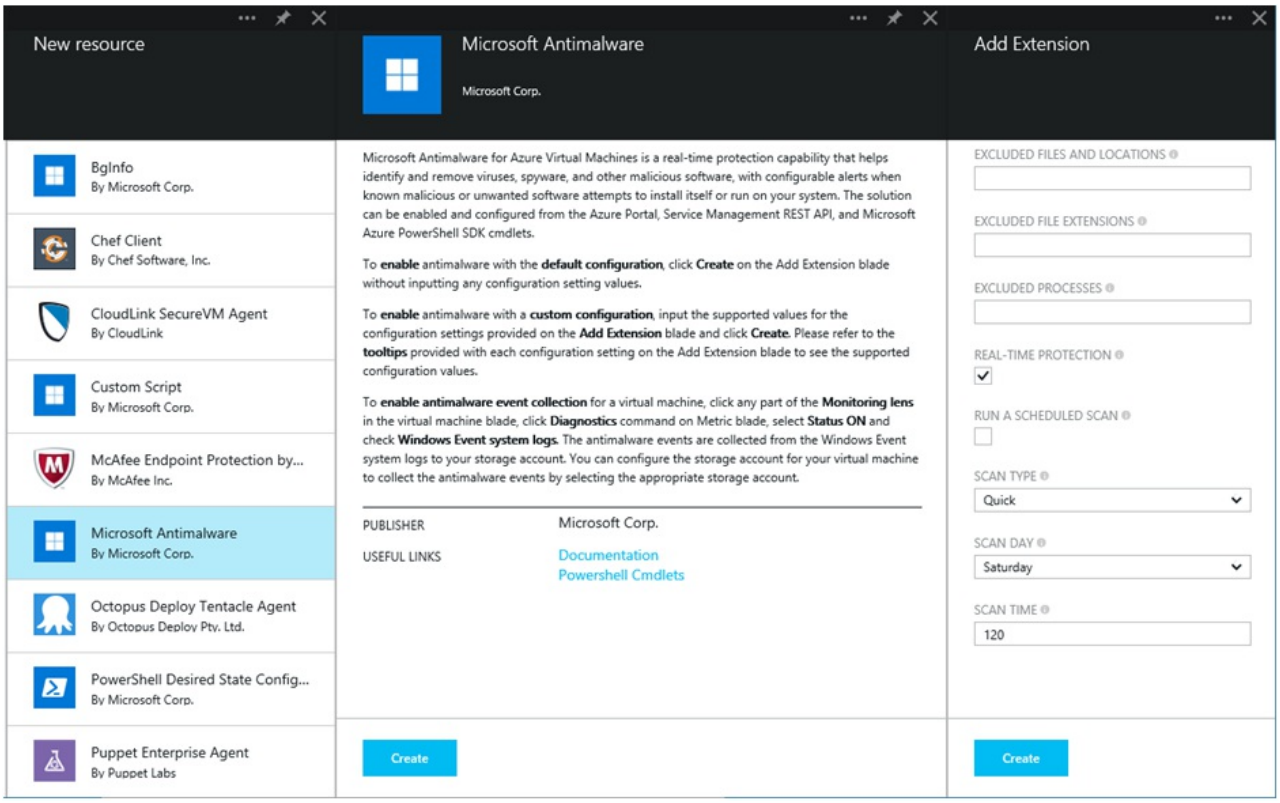
You can leverage SCM to import the current configuration of your computers using two different methods: first, you can import Active Directory-based group policies; second, you can import the configuration of a "golden master" reference machine by using the LocalGPO tool to backup the local group policy which you can then import into SCM.

Compare your standards to industry best practices, customize them , and create new policies and DCM configuration packs. Baselines have been published for all supported operating systems, including Windows 10 Anniversary Update and Windows Server 2016.

Install and manage antimalware

For environments that are hosted separately from your production environment there is an antimalware extension that can be used to protect your virtual machines and cloud services and it integrates with Azure Security Center.

Microsoft Antimalware includes features like real-time protection, scheduled scanning, malware remediation, signature updates, engine updates, samples reporting, exclusion event collection, and PowerShell support.



Install the latest security updates Some of the first workloads we see customers move to Azure are labs and external facing systems. If you are hosting virtual machines in Azure that host applications or services that need to be made accessible to the Internet, you need to be vigilant about patching. Remember that this goes beyond patching the operating system. Unpatched vulnerabilities on third-party applications can also lead to problems that would have been easily avoided if good patch management was in place.

For more information on managing patching in Azure IaaS look at Best practices for software updates on Microsoft Azure IaaS.

Deploy and test a backup solution

Just like security updates, backup needs to be handled the same way 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 are able of providing similar restore capabilities to what users have grown accustomed to based on their experience with on-premise environments.

Production workloads moved to Azure should integrate with existing backup solutions when possible or you can use Azure Backup to help you address your backup requirements.

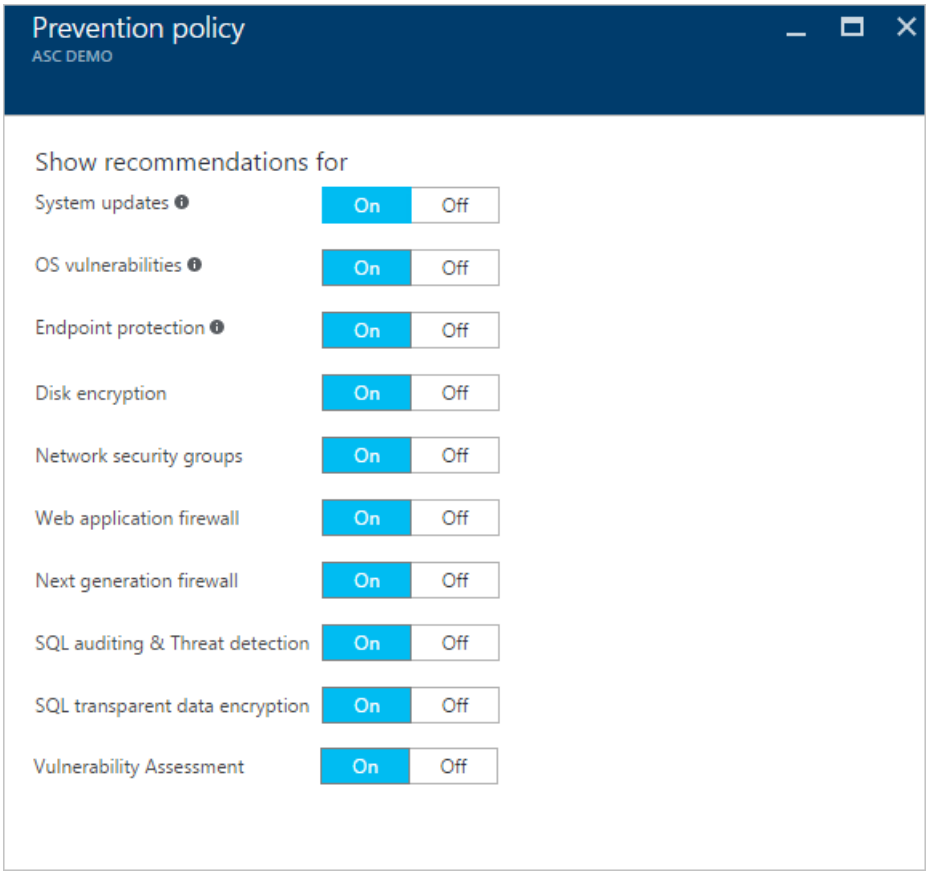
Monitor

Security Center provides ongoing evaluation of 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 virtual machines
- 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

In the image below you can see some of the options available for you to enable in Security Center.



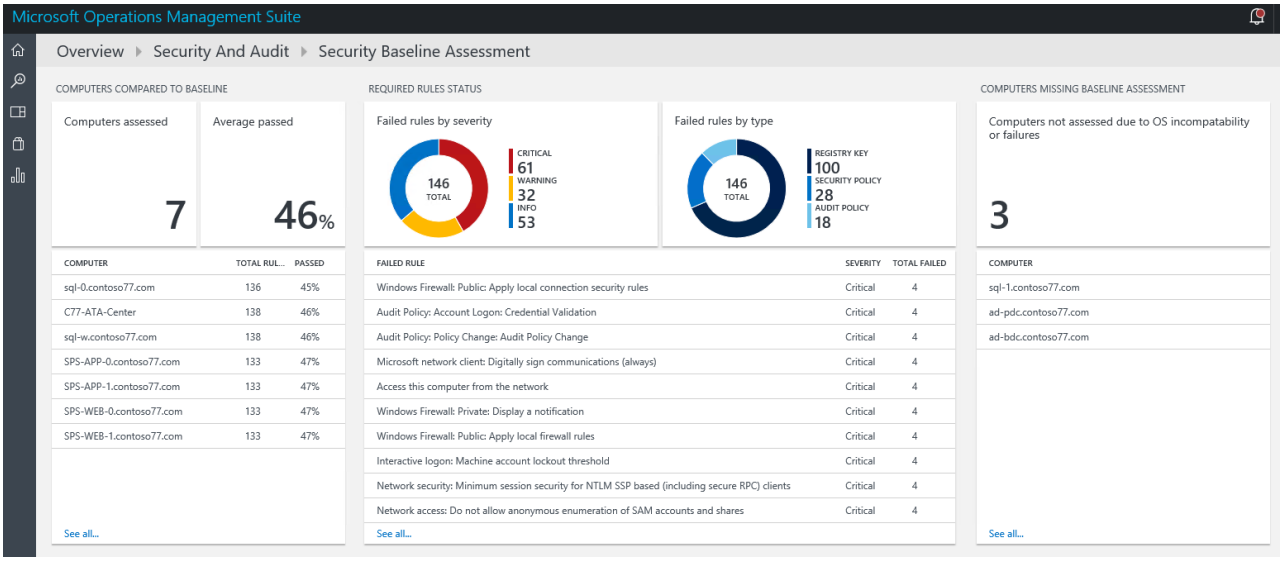
Operations Management Suite is a Microsoft cloud based IT management solution that helps you manage and protect your on-premise and cloud infrastructure. Since OMS is implemented as a cloud based service it can be deployed quickly and with minimal investment in infrastructure resources.

New features are delivered automatically saving you from ongoing maintenance and upgrade costs. It also integrates with System Center Operations Manager. OMS has different components to help you better manage your Azure workloads including a Security and Compliance module.

The security and compliance features in OMS allow you to view information about your resources organized into four major categories:

- Security Domains: in this area you will be able to further explore security records over time, access malware assessment, update assessment, network security, identity and access information, computers with security events and quickly have access to Azure Security Center dashboard.
- Notable Issues: this option will allow you to quickly identify the number of active issues and the severity of these issues.
- Detections (Preview): enables you to identify attack patterns by visualizing security alerts as they take place against your resources.
- Threat Intelligence: enables you to identify attack patterns by visualizing the total number of servers with outbound malicious IP traffic, the malicious threat type, and a map that shows where these IPs are coming from.
- Common security queries: this option provides you a list of the most common security queries that you can use to monitor your environment. When you click in one of those queries, it opens the Search blade with the results for that query

The screenshot below shows an example of the type of information that can be displayed by OMS.



Next Steps

- Azure security Team Blog
- Microsoft Security Response Center
- Azure Security Best Practices and Patterns

Microsoft Antimalware for Azure Cloud Services and Virtual Machines

11/15/2016 • 11 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil

---

The modern threat landscape for cloud environments is extremely dynamic, increasing the pressure on business IT cloud subscribers to maintain effective protection in order to meet compliance and security requirements. Microsoft Antimalware for Azure Cloud Services and Virtual Machines is free 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 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 for Windows 8.0 and higher. 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.

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** - 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 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 certain files, processes, and drives to exclude them from protection and scanning for performance and/or 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.

> **NOTE**
> Microsoft Antimalware can also be deployed using Azure Security Center. Read Install Endpoint Protection in Azure Security Center for more information.

Architecture

The Microsoft Antimalware for Azure Cloud Services and Virtual Machines solution includes the Microsoft Antimalware Client and Service, Antimalware classic deployment model, Antimalware PowerShell cmdlets and Azure Diagnostics Extension. The Microsoft Antimalware solution 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. At this time, Windows Server Technical Preview is not supported and we intend to support it in the future.

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 Websites, the underlying service that hosts the web app has Microsoft Antimalware enabled on it. This is used to protect Azure Websites infrastructure and does not run on customer content.

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:

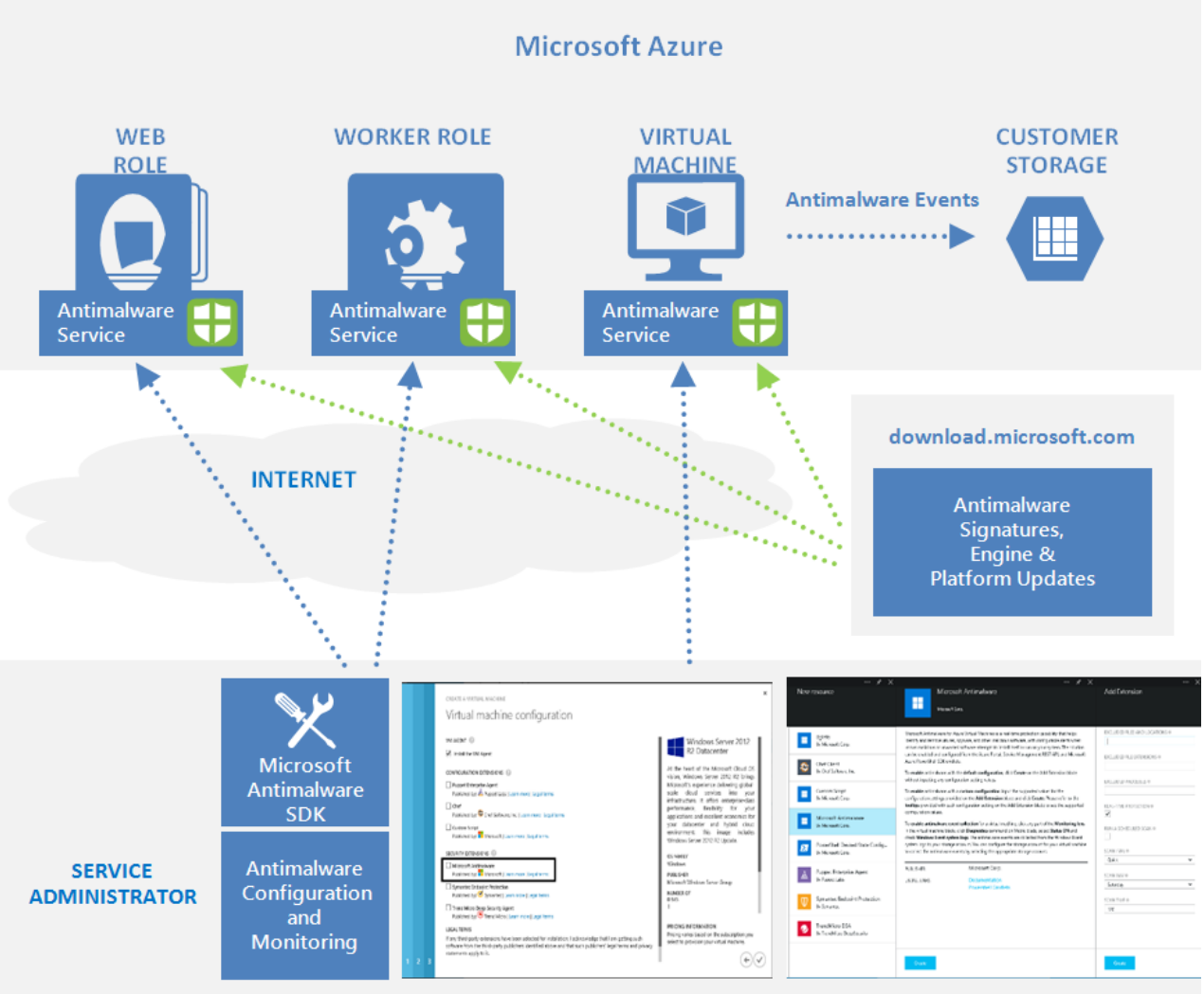
- Virtual Machines – In the Azure portal, under **Security Extensions**
- Virtual Machines – Using the Visual Studio virtual machines configuration in Server Explorer
- Virtual Machines and Cloud Services – Using the Antimalware classic deployment model
- Virtual Machines and Cloud Services – Using Antimalware 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 Cloud Services and Virtual Machines – 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 *vmssextn.py* located here. 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

			... 1380 mins = 11:00 PM Default = 120 mins if ScheduledScanSettings isEnabled = true
ScheduledScanSettings Scan Type	Quick/Full	Quick	Default = Quick if ScheduledScanSettings isEnabled = true
Monitoring	ON OFF	OFF	ON - Enable Antimalware event collection to user subscription storage using Azure Diagnostics extension OFF – Disable Antimalware event collection to user subscription storage by removing antimalware monitoring configuration in Azure Diagnostics extension if it was previously turned ON
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

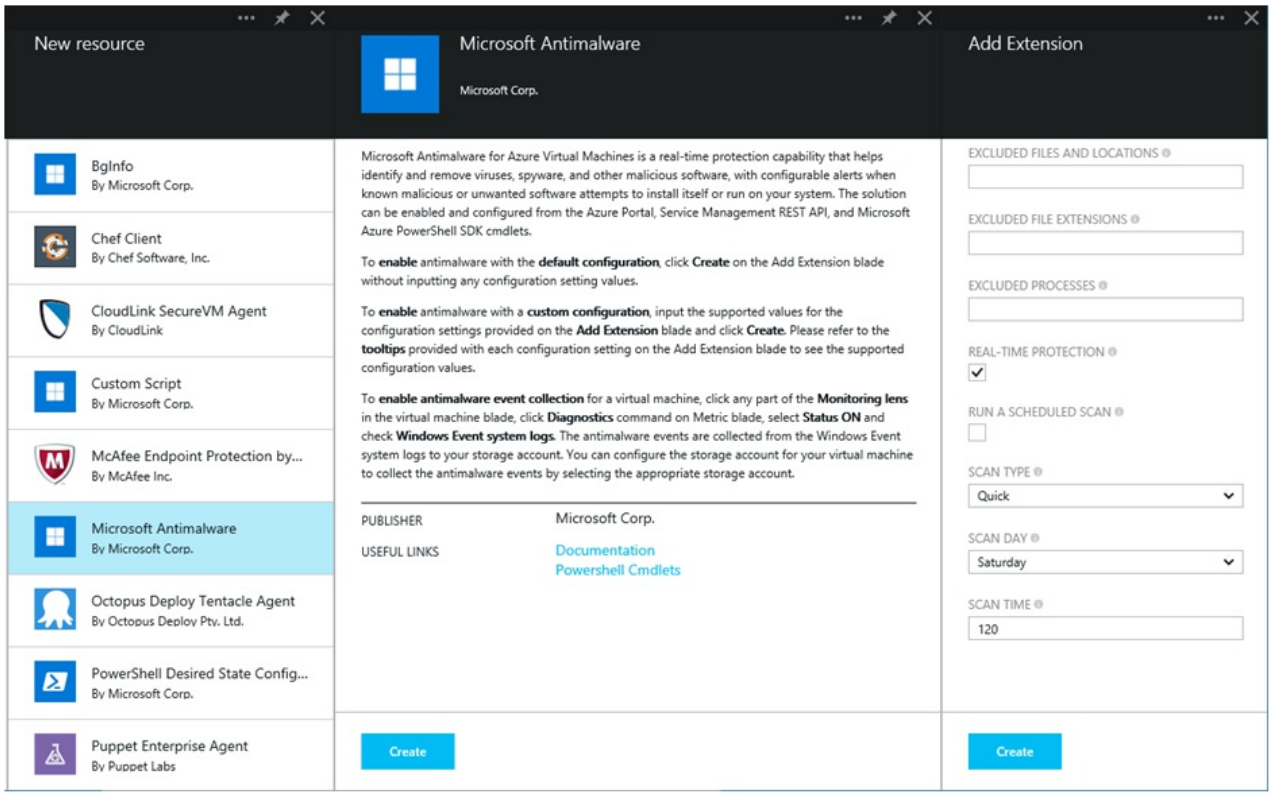
Antimalware Deployment Scenarios

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 using Azure Portal

To enable the Antimalware service, click **Add** on the Extensions blade, select **Microsoft Antimalware** on the New resource blade, click **Create** on the Microsoft Antimalware blade. Click **Create** without inputting any configuration values to enable Antimalware with the default settings, or enter the Antimalware configuration settings for the Virtual Machine configured as shown in Figure 2 below. Please refer to the **tooltips** provided with each configuration setting on the Add Extension blade to see the supported configuration values.



Deployment using the Azure classic 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.Log onto the Azure portal at https://portal.azure.com

2.To create a new virtual machine, click **New**, **Compute**, **Virtual Machine**, **From Gallery** (do not use Quick Create) as shown below:

3.Select the **Microsoft Windows Server** image on the **Choose an Image** page.

4.Click the right arrow and input the Virtual Machine configuration.

5.Check the **Microsoft Antimalware** checkbox under **Security Extensions** on the Virtual Machine configuration page.

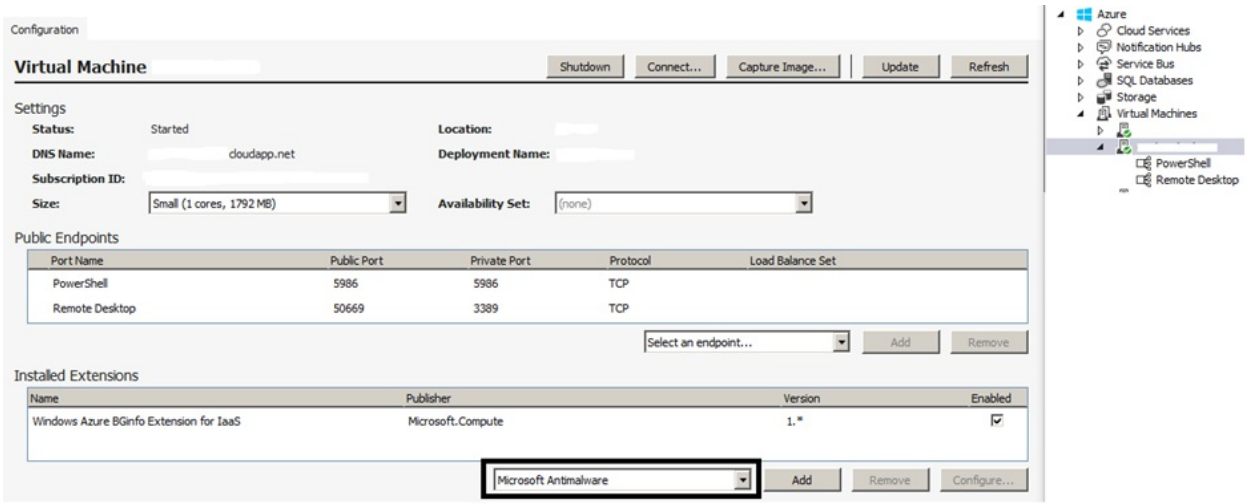
6.Click the Submit button to enable and configure Microsoft Antimalware for Azure Virtual Machines with the default configuration settings.

Deployment Using the Visual Studio virtual machine configuration

To enable and configure the Microsoft Antimalware service using Visual Studio:

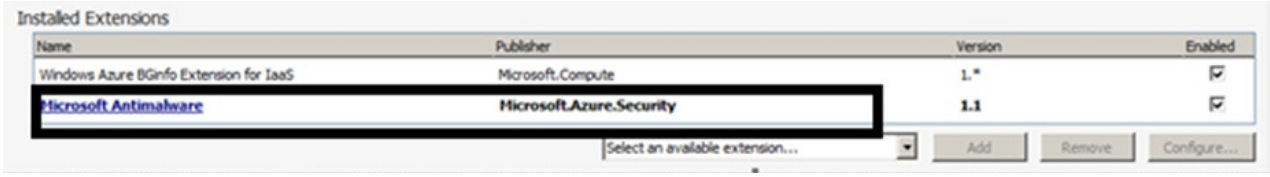
1.Connect to Microsoft Azure in Visual Studio.

2.Choose your Virtual Machine in the **Virtual Machines** node in **Server Explorer**



3.Right click **configure** to view the Virtual Machine configuration page

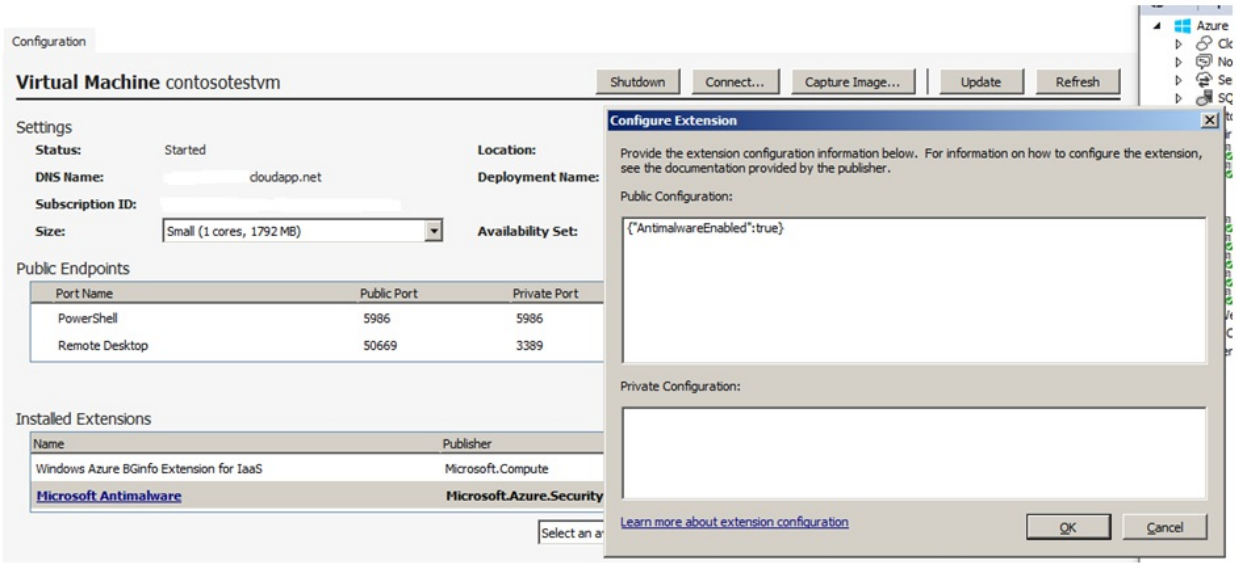
4.Select **Microsoft Antimalware** extension from the dropdown list under **Installed Extensions** and click **Add** to configure with default antimalware configuration.



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 Cloud Services and Virtual Machine - 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 as documented at http://msdn.microsoft.com/library/azure/dn771718.aspx

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 Cloud Services and Virtual Machine - 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-sdk-tools#get-started
2. Use the Set-AzureServiceAntimalwareExtension Antimalware cmdlet to enable and configure Microsoft Antimalware for your Cloud Service as documented at http://msdn.microsoft.com/library/azure/dn771718.aspx

The Antimalware XML configuration settings template is included in the Microsoft Antimalware For Azure Cloud

Services and Virtual Machine - 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-sdk-tools#get-started
2. **For Virtual Machines**: Use the Get-AzureVMMicrosoftAntimalwareExtension Antimalware cmdlet to get the antimalware configuration as documented at http://msdn.microsoft.com/library/azure/dn771719.aspx
3. **For Cloud Services**: Use the Get-AzureServiceAntimalwareConfig Antimalware cmdlet to get the Antimalware configuration as documented at http://msdn.microsoft.com/library/azure/dn771722.aspx

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.

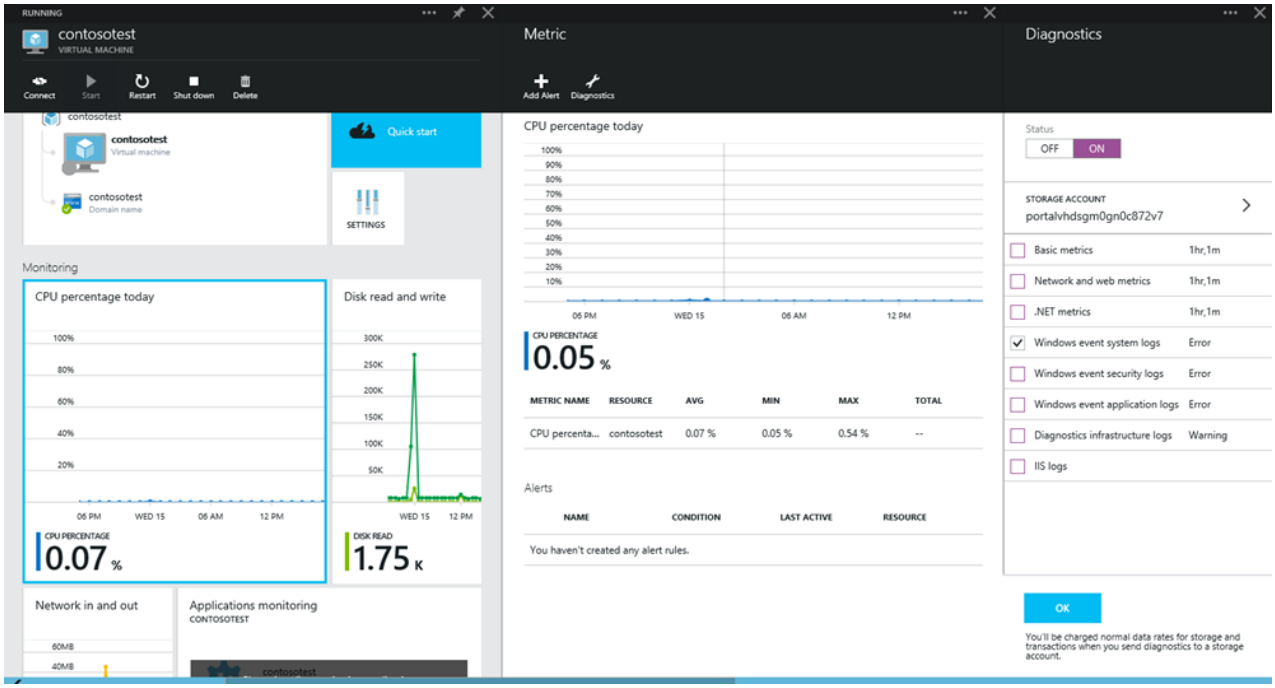
To remove Microsoft Antimalware using PowerShell cmdlets:

1. Set up your PowerShell environment - Refer to the documentation at https://github.com/Azure/azure-sdk-tools#get-started
2. **For Virtual Machines**: Use the Remove-AzureVMMicrosoftAntimalwareExtension Antimalware cmdlet as documented at http://msdn.microsoft.com/library/azure/dn771720.aspx
3. **For Cloud Services:** Use the Remove-AzureServiceAntimalwareExtension Antimalware cmdlet as documented at http://msdn.microsoft.com/library/azure/dn771717.aspx

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-sdk-tools#get-started
2. **For Virtual Machines** - Use the Set-AzureVMMicrosoftAntimalwareExtension Antimalware cmdlet with Monitoring ON option as documented at http://msdn.microsoft.com/library/azure/dn771716.aspx
3. **For Cloud Services** - Use the Set-AzureServiceAntimalwareExtension Antimalware cmdlet with Monitoring ON option as documented at http://msdn.microsoft.com/library/azure/dn771718.aspx

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 details, including sample code on how to extract Antimalware events from your storage account, refer to Microsoft Antimalware For Azure Cloud Services and Virtual Machine - Code Samples.

Azure Disk Encryption for Windows and Linux IaaS VMs

11/15/2016 • 42 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Kamran Khan • Devendra Tiwari • unknown • microsoftman • Victor Ashik • Cody Mansfield • Carolyn Gronlund • Simon Rolfe

---

Microsoft Azure is strongly committed to ensuring your data privacy, data sovereignty and enables you to control your Azure hosted data through a range of advanced technologies to encrypt, control and manage encryption keys, control & audit access of data. This provides Azure customers the flexibility to choose the solution that best meets their business needs. In this paper, we will introduce you to a new technology solution "Azure Disk Encryption for Windows and Linux IaaS VM's" to help protect and safeguard your data to meet your organizational security and compliance commitments. The paper provides detailed guidance on how to use the Azure disk encryption features including the supported scenarios and the user experiences.

NOTE: Certain recommendations contained herein may result in increased data, network, or compute resource usage resulting in additional license or subscription costs.

Overview

Azure Disk Encryption is a new capability that lets you encrypt your Windows and Linux IaaS virtual machine disks. Azure 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 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, while ensuring that all data in the virtual machine disks are encrypted at rest in your Azure storage.

Azure disk encryption for Windows and Linux IaaS VMs is now in **General Availability** in all Azure public regions for Standard VMs and VMs with premium storage.

Encryption Scenarios

The Azure Disk Encryption solution supports the following customer scenarios:

- Enable encryption on new IaaS VMs created from pre-encrypted VHD and encryption keys
- Enable encryption on new IaaS VMs created from the Azure Gallery images
- Enable encryption on existing IaaS VMs running in Azure
- Disable encryption on Windows IaaS VMs
- Disable encryption on data drives for Linux IaaS VMs

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

- Integration with Azure Key Vault
- Standard tier VMs - A, D, DS, G, GS etc series IaaS VMs
- Enable encryption on Windows and Linux IaaS VMs
- Disable encryption on OS and data drives for Windows IaaS VMs
- Disable encryption on data drives for Linux IaaS VMs
- Enable encryption on IaaS VMs running Windows Client OS
- Enable encryption on volumes with mount paths

- Enable encryption on Linux VMs configured with Software-based RAID system
- Enable encryption on Windows VMs configured with 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
- Disable encryption on OS drive for Linux IaaS VMs
- IaaS VMs created using classic VM creation method
- Integration with your on-premises Key Management Service
- Windows Server 2016 Technical Preview is not supported in this release
- Azure Files (Azure file share), Network file system (NFS), Dynamic volumes, Windows VMs configured with Software-based RAID systems

Encryption Features

When you enable and deploy Azure disk encryption for Azure IaaS VMs, the following capabilities are enabled, depending on the configuration provided:

- Encryption of OS volume to protect boot volume at rest in customer storage
- Encryption of Data volume/s to protect the data volumes at rest in customer storage
- Disable encryption on OS and data drives for Windows IaaS VMs
- Disable encryption on data drives for Linux IaaS VMs
- Safeguarding the encryption keys and secrets in customer Azure key vault subscription
- Reporting encryption status of the encrypted IaaS VM
- Removal of disk encryption configuration settings from the IaaS virtual machine

The Azure disk encryption for IaaS VMS for Windows and Linux solution includes the disk encryption extension for Windows, disk encryption extension for Linux, disk encryption PowerShell cmdlets, disk encryption CLI cmdlets and disk encryption Azure Resource Manager templates. The Azure disk encryption solution is supported on IaaS VMs running Windows or Linux OS. For more details on the supported Operating Systems, see prerequisites section below.

NOTE: There is no additional charge for encrypting VM disks with Azure Disk Encryption.

Value Proposition

The Azure Disk Encryption Management solution enables the following business needs in the cloud:

- IaaS VM's are secured at rest using industry standard encryption technology to address organizational security and compliance requirements.
- IaaS VM's boot under customer controlled keys and policies, and they can audit their usage in Key Vault.

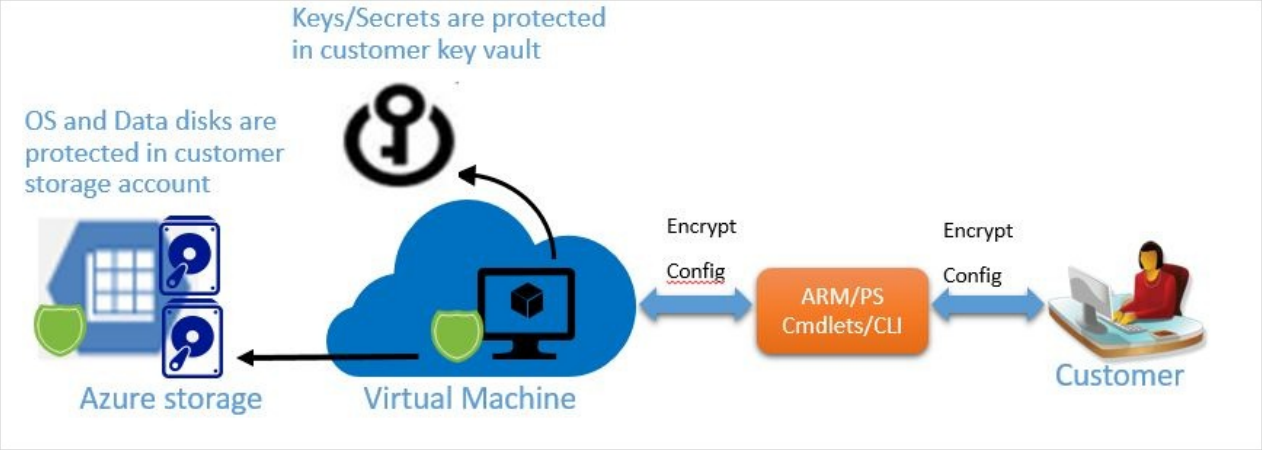
Encryption Workflow

The high level steps required to enable disk encryption for Windows and Linux VM's are:

1. Customer chooses a encryption scenario from the above encryption scenarios
2. Customer opts into enabling disk encryption via the Azure disk encryption Resource Manager template or PS cmdlets or CLI command and specifies the encryption configuration
    - For the customer encrypted VHD scenario, the customer uploads the encrypted VHD to their storage account and encryption key material to their key vault and provide the encryption configuration to enable encryption on a new IaaS VM
    - For the new VM's created from the Azure gallery and existing VM's already running in Azure, customer provide the encryption configuration to enable encryption on the IaaS VM
3. Customer grants access to Azure platform to read the encryption key material (BitLocker Encryption Keys for

Windows systems and Passphrase for Linux) from their key vault to enable encryption on the IaaS VM

4. Customer provide Azure AD application identity to write the encryption key material to their key vault to enable encryption on the IaaS VM for scenarios mentioned in #2 above
5. Azure updates the VM service model with encryption and key vault configuration and provisions encrypted VM for the customer



Decryption Workflow

The high level steps required to disable disk encryption for IaaS VM's are:

1. Customer chooses to disable encryption (decryption) on a running IaaS VM in Azure via the Azure disk encryption Resource Manager template or PS cmdlets and specifies the decryption configuration.
2. The disable encryption step disables encryption of the OS or data volume or both on the running Windows IaaS VM. However disabling OS disk encryption for Linux is not supported as mentioned in the documentation above. The disable step is allowed only for data drives on Linux VMs.
3. Azure updates the VM service model and the IaaS VM is marked decrypted. The contents of the VM are not encrypted at rest anymore.
4. The disable encryption operation does not delete the customer key vault and the encryption key material, - BitLocker Encryption Keys for Windows or Passphrase for Linux.

Prerequisites

The following are prerequisites to enable Azure Disk Encryption on Azure IaaS VMs for the supported scenarios called out in the overview section

- User must have a valid active Azure subscription to create resources in Azure in the regions supported
- Azure Disk Encryption is supported on the following Windows server SKU's - Windows Server 2008 R2, Windows Server 2012 and Windows Server 2012 R2. Windows Server 2016 Technical Preview is not supported in this release.
- Azure Disk Encryption is supported on the following Windows client SKU's - Windows 8 Client and Windows 10 Client.

Note: For Windows Server 2008 R2, .Net framework 4.5 MUST be installed before enabling encryption in Azure. You can install it from Windows update by installing the optional update "Microsoft .NET Framework 4.5.2 for Windows Server 2008 R2 x64-based Systems (KB2901983)"

- Azure Disk Encryption is supported on the following Linux server SKUs - Ubuntu, CentOS, SUSE and SUSE Linux Enterprise Server (SLES) and Red Hat Enterprise Linux.

Note: Linux OS disk encryption is currently supported on the following Linux distributions - RHEL 7.2, CentOS 7.2, Ubuntu 16.04

- All resources (Ex: Key Vault, Storage account, VM, etc.,) must belong to the same Azure region and subscription.

Note: Azure disk encryption requires that the Key Vault and the VMs reside in the same Azure region. Configuring them in separate region will cause failure in enabling Azure disk encryption feature.

- To set up and configure Azure Key Vault for Azure disk encryption usage, see section **Setting and Configuring Azure Key Vault for Azure disk encryption usage** in the *Prerequisites* section of this article.
- To set up and configure Azure AD application in Azure Active directory for Azure disk encryption usage, see section **Setup the Azure AD Application in Azure Active Directory** in the *Prerequisites* section of this article.
- To set up and configure Key Vault Access policy for the Azure AD Application, see section **Setting Key Vault Access policy for the Azure AD Application** in the *Prerequisites* section of this article.
- To prepare a pre-encrypted Windows VHD, see section **Preparing a pre-encrypted Windows VHD** in the Appendix of this article.
- To prepare a pre-encrypted Linux VHD, see section **Preparing a pre-encrypted Linux VHD** in the Appendix of this article.
- Azure platform needs access to the encryption keys or secrets in customer Azure Key Vault in order to make them available to the virtual machine to boot and decrypt the virtual machine OS volume. To grant permissions to Azure platform to access the customer Key Vault, **enabledForDiskEncryption** property must be set on the Key Vault for this requirement. Refer to section **Setting and Configuring Azure Key Vault for Azure disk encryption usage** in the Appendix of this article for more details.
- The Key Vault secret and key encryption key (KEK) URLs must be versioned. Azure enforces this restriction of versioning. See below examples for valid secret and KEK URL:
  - Example of valid secret URL: *https://contosovault.vault.azure.net/secrets/BitLockerEncryptionSecretWithKek/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
  - Example of valid KRK KEK: *https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
- Azure disk encryption does not support port numbers being specified as part of Key Vault secret and KEK URLs. See below examples for supported Key Vault URL:
  - Unaccepted Key Vault URL *https://contosovault.vault.azure.net:443/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
    - Accepted Key Vault URL: *https://contosovault.vault.azure.net/secrets/contososecret/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
- To enable Azure Disk Encryption feature, the IaaS VMs must meet the following network endpoint configuration requirements:
  - The IaaS VM must be able to connect to Azure Active Directory endpoint [Login.windows.net] to get a token to connect to Azure key vault
  - The IaaS VM must be able to connect to Azure Key Vault endpoint to write the encryptions keys to customer key vault
  - The IaaS VM must be able to connect to Azure storage endpoint which hosts the Azure extension repository and Azure storage account which hosts the VHD files

Note: If your security policy limits access from Azure VMs to Internet, you can resolve the above URI to which you need connectivity and configure a specific rule to allow outbound connectivity to the IPs.

- Use the latest version of Azure PowerShell SDK version to configure Azure Disk Encryption. Download the latest version of Azure PowerShell release

Note: Azure Disk Encryption is not supported on Azure PowerShell SDK version 1.1.0. If you are receiving an error related to using Azure PowerShell 1.1.0, please see the article Azure Disk Encryption Error Related to Azure PowerShell 1.1.0.

- To run any of the Azure CLI commands and associate it with your Azure subscription, you must first install Azure CLI version:

    - To install Azure CLI and associate it with your Azure subscription, see How to install and configure Azure CLI
    - Using the Azure CLI for Mac, Linux, and Windows with Azure Resource Manager, see here
- Azure disk encryption solution use BitLocker external key protector for Windows IaaS VMs. If your VMs are domain joined, do not push any group policies that enforce TPM protectors. Refer to this article for details on the group policy for "Allow BitLocker without a compatible TPM".
- The Azure disk encryption prerequisite PowerShell script to create Azure AD application, create new key vault or setup existing key vault and enable encryption is located here.

Setup the Azure AD Application in Azure Active Directory

When encryption needs 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 Key Vault needs Azure AD authentication.

For this purpose, an Azure AD application should be created. Detailed steps for registering an application can be found here, in the section "Get an Identity for the Application" section in this blog post. This post also contains a number of helpful examples on provisioning and configuring your Key Vault. For authentication purposes, either client secret based authentication or client certificate-based Azure AD authentication can be used.

Client secret based authentication for Azure AD

The sections that follow have the necessary steps to configure a client secret based authentication for Azure AD.

Create a new Azure AD app using Azure PowerShell

Use the PowerShell cmdlet below to create a new Azure AD app:

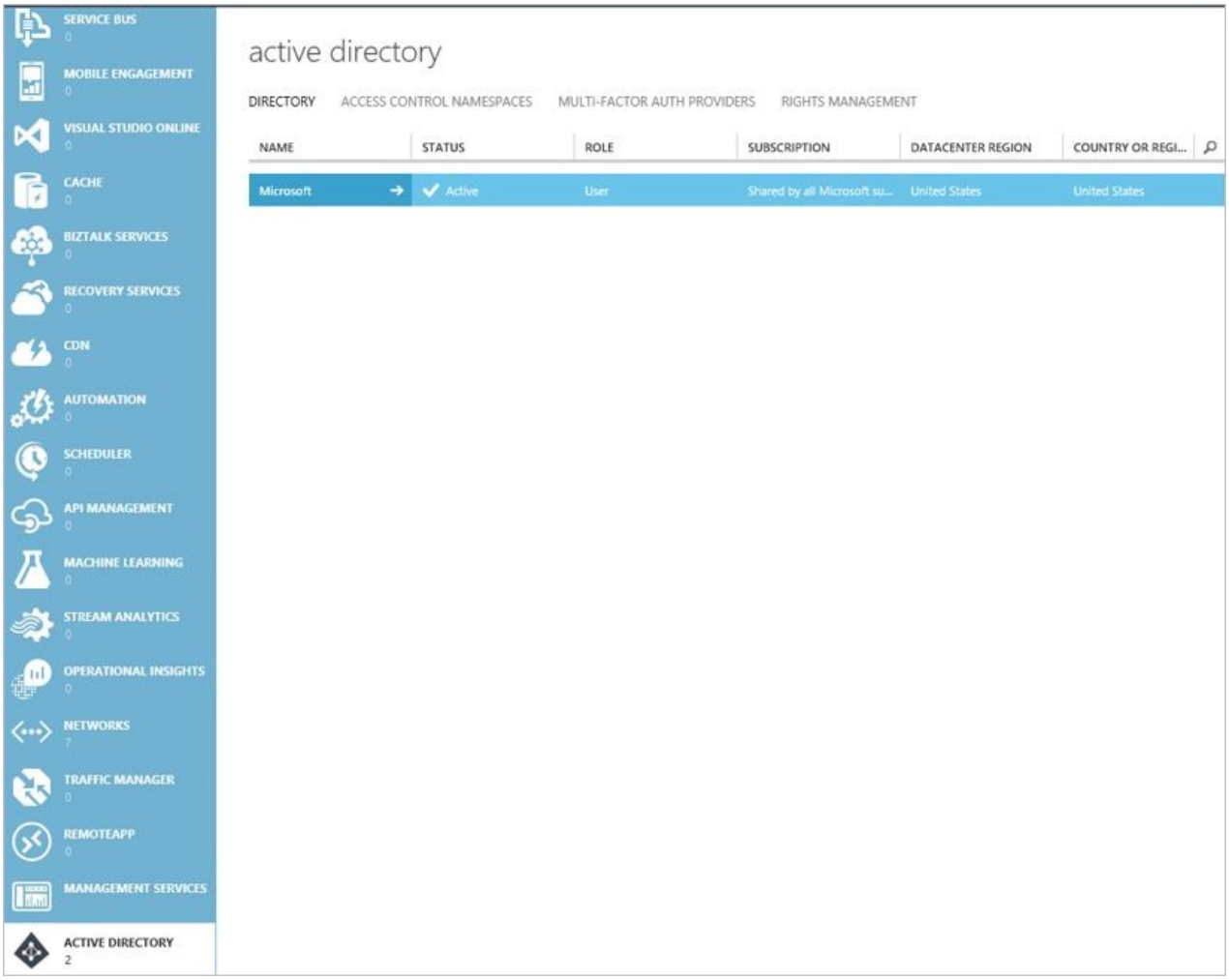
```
$aadClientSecret = "yourSecret"
$azureAdApplication = New-AzureRmADApplication -DisplayName "<Your Application Display Name>" -HomePage "
<https://YourApplicationHomePage>" -IdentifierUris "<https://YouApplicationUri>" -Password $aadClientSecret
$servicePrincipal = New-AzureRmADServicePrincipal –ApplicationId $azureAdApplication.ApplicationId
```

Note: $azureAdApplication.ApplicationId is the Azure AD ClientID and $aadClientSecret is the client Secret that you should use later to enable ADE.You should safeguard the Azure AD client secret appropriately.

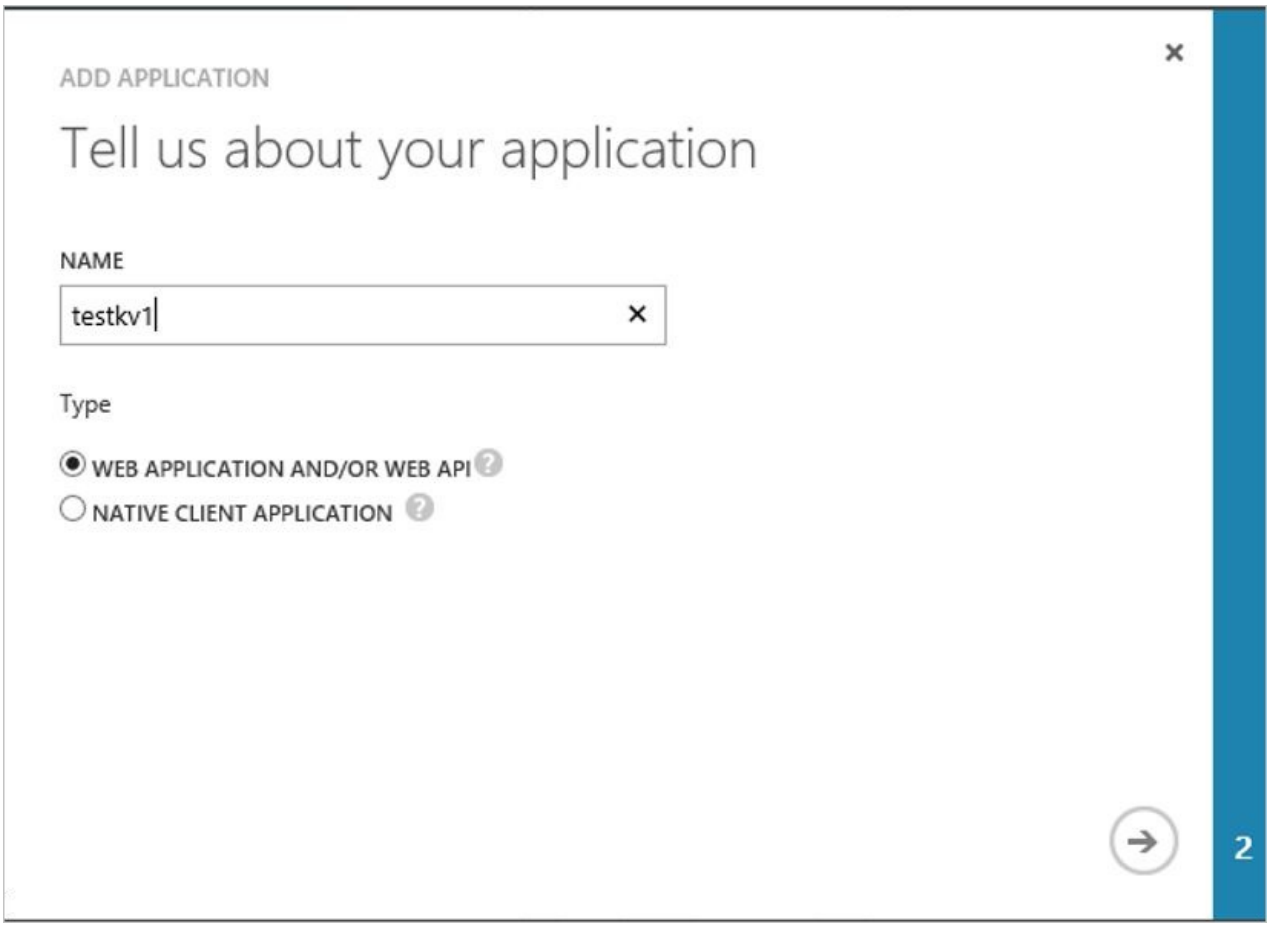
Provisioning the Azure AD client ID and secret from

Azure AD Client ID and secret can also be provisioned using the Azure Classic deployment model Portal at https://manage.windowsazure.com, follow the steps below to perform this task:

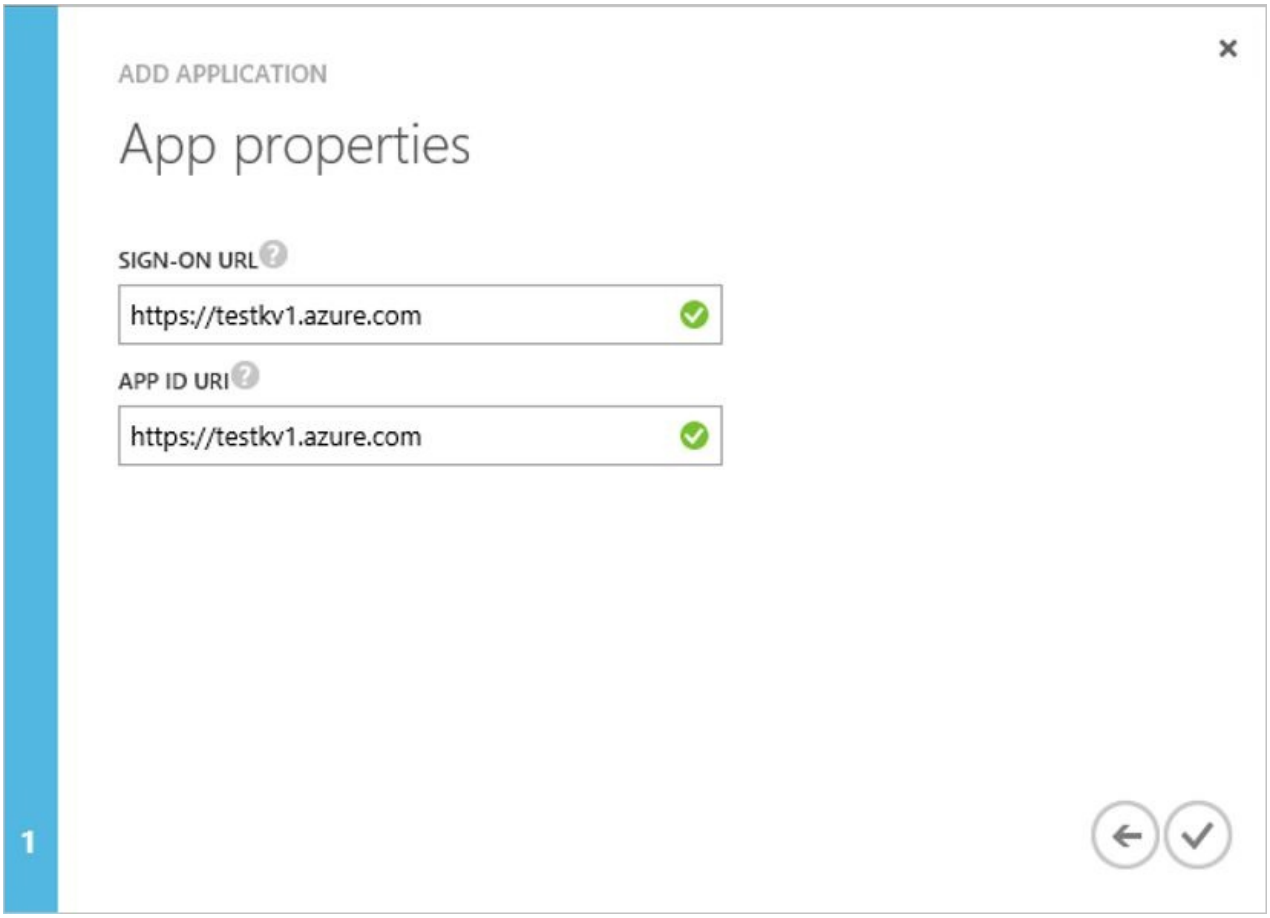
1.Click the Active Directory tab as shown in Figure below:



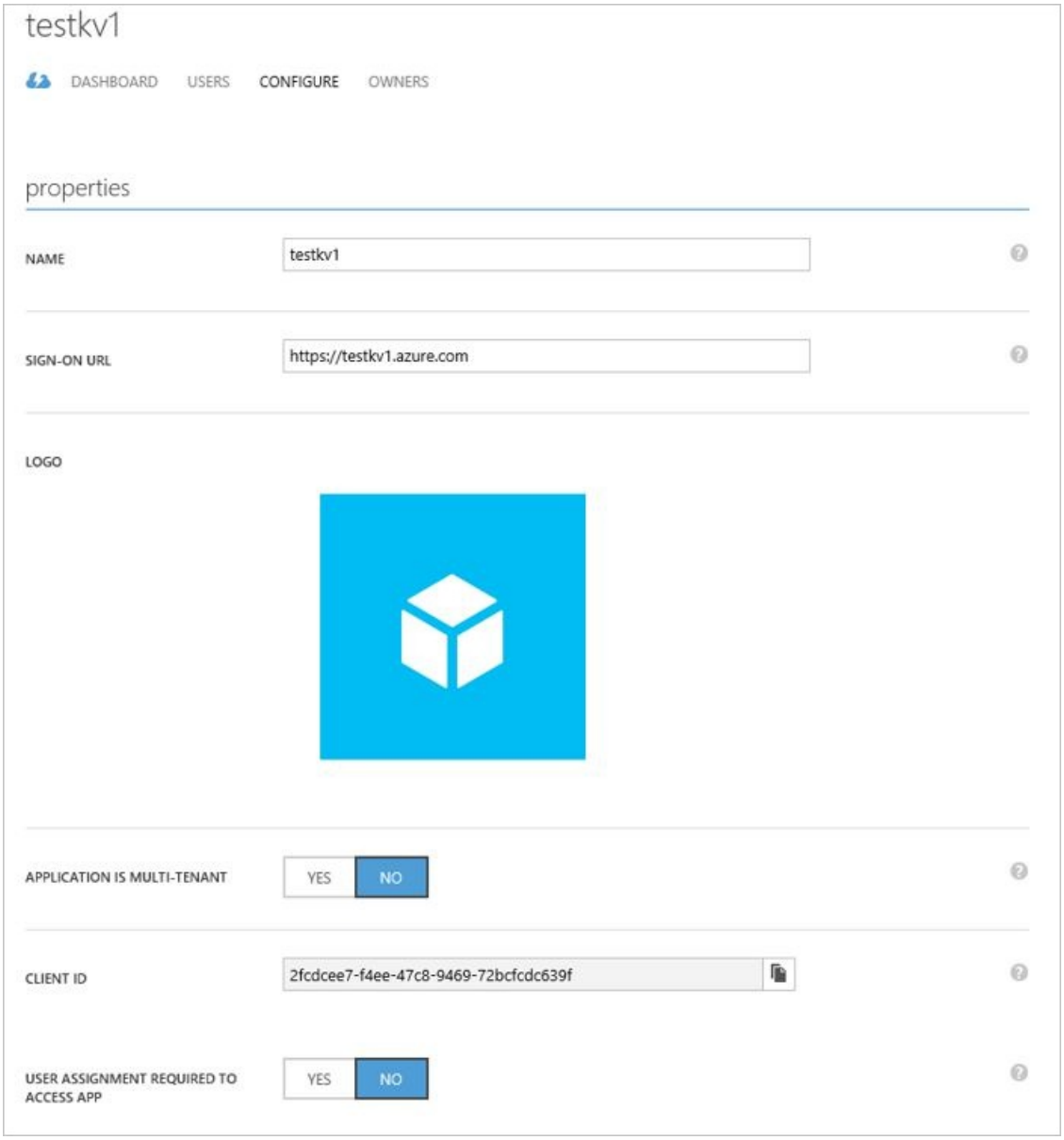
2.Click Add Application and type the application name as shown below:



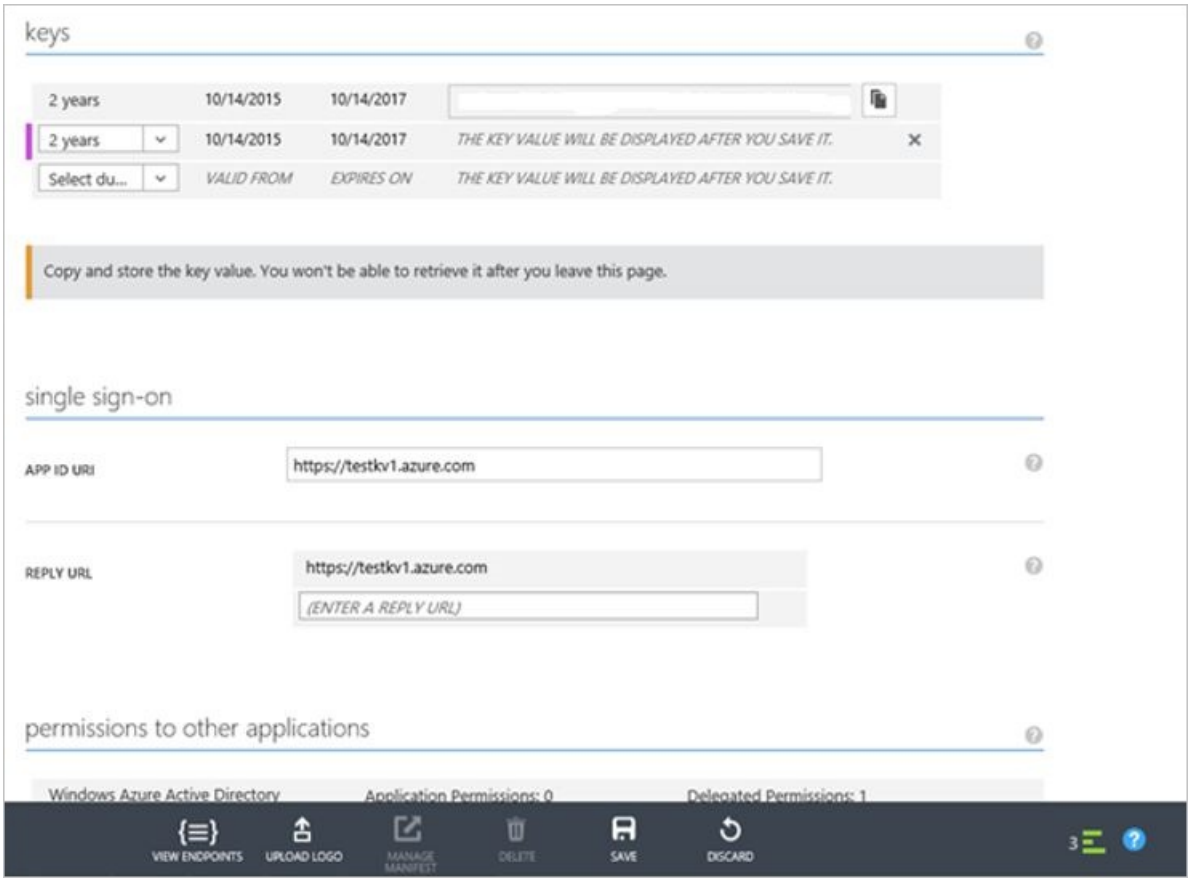
3.Click the arrow button and configure the app's properties as shown below:



4.Click the check mark in the lower left corner to finish. The app's configuration page appears. Notice the Azure AD Client ID is located in the bottom of the page as shown in figure below.



5.Save the Azure AD client secret by click in the Save button. Click the save button and note the secret from the keys text box, this is the Azure AD client secret. You should safeguard the Azure AD client secret appropriately.



Note: this flow above is not supported in the Portal.

Use an existing app

In order to execute the commands below you need the Azure AD PowerShell module, which can be obtained from here.

Note: the commands below must be executed from a new PowerShell window. Do NOT use Azure PowerShell or the Azure Resource Manager window to execute these commands. The reason for this recommendation is because these cmdlets are in the MSOnline module or Azure AD PowerShell.

```
$clientSecret = '<yourAadClientSecret>'
$aadClientID = '<Client ID of your AAD app>'
connect-msolservice
New-MsolServicePrincipalCredential -AppPrincipalId $aadClientID -Type password -Value $clientSecret
```

Certificate based authentication for Azure AD

> **NOTE**
> AAD certificate based authentication is currently not supported on Linux VMs.

The sections that follow have the necessary steps to configure a certificate based authentication for Azure AD.

Create a new Azure AD app

Execute the PowerShell cmdlets below to create a new Azure AD app:

Note: Replace `yourpassword` string below with your secure password and safeguard the password.

```
$cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate("C:\certificates\examplecert.pfx", "yourpassword")
$keyValue = [System.Convert]::ToBase64String($cert.GetRawCertData())
$azureAdApplication = New-AzureRmADApplication -DisplayName "<Your Application Display Name>" -HomePage "
<https://YourApplicationHomePage>" -IdentifierUris "<https://YouApplicationUri>" -KeyValue $keyValue -KeyType
AsymmetricX509Cert
$servicePrincipal = New-AzureRmADServicePrincipal –ApplicationId $azureAdApplication.ApplicationId
```

Once you finish this step, upload a .pfx file to Key Vault and enable the access policy needed to deploy that certificate to a VM.

Use an existing Azure AD app

If you are configuring certificate based authentication for an existing app, use the PowerShell cmdlets below. Make sure to execute them from a new PowerShell window.

```
$certLocalPath = 'C:\certs\myaadapp.cer'
$aadClientID = '<Client ID of your AAD app>'
connect-msolservice
$cer = New-Object System.Security.Cryptography.X509Certificates.X509Certificate
$cer.Import($certLocalPath)
$binCert = $cer.GetRawCertData()
$credValue = [System.Convert]::ToBase64String($binCert);
New-MsolServicePrincipalCredential -AppPrincipalId $aadClientID -Type asymmetric -Value $credValue -Usage verify
```

Once you finish this step, upload a .pfx file to Key Vault and enable the access policy needed to deploy that certificate to a VM.

Upload a PFX file to Key Vault

You can read this blog post for detail explanation on how this process works. However, the PowerShell cmdlets below are all you need for this task. Make sure to execute them from Azure PowerShell console:

Note: Replace `yourpassword` string below with your secure password and safeguard the password.

```
$certLocalPath = 'C:\certs\myaadapp.pfx'
$certPassword = "yourpassword"
$resourceGroupName = ‘yourResourceGroup’
$keyVaultName = ‘yourKeyVaultName’
$keyVaultSecretName = ‘yourAadCertSecretName’

$fileContentBytes = get-content $certLocalPath -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)

Switch-AzureMode -Name AzureResourceManager
$secret = ConvertTo-SecureString -String $jsonEncoded -AsPlainText -Force
Set-AzureKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName -SecretValue $secret
Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ResourceGroupName $resourceGroupName –
EnabledForDeployment
```

Deploy a certificate in Key Vault to an existing VM

After finishing uploading the PFX, use the steps below to deploy a certificate in Key Vault to an existing VM:

```
$resourceGroupName = ‘yourResourceGroup’
$keyVaultName = ‘yourKeyVaultName’
$keyVaultSecretName = ‘yourAadCertSecretName’
$vmName = ‘yourVMName’
$certUrl = (Get-AzureKeyVaultSecret -VaultName $keyVaultName -Name $keyVaultSecretName).Id
$sourceVaultId = (Get-AzureRmKeyVault -VaultName $keyVaultName -ResourceGroupName $resourceGroupName).ResourceId
$vm = Get-AzureRmVM -ResourceGroupName $resourceGroupName -Name $vmName
$vm = Add-AzureRmVMSecret -VM $vm -SourceVaultId $sourceVaultId -CertificateStore "My" -CertificateUrl $certUrl
Update-AzureRmVM -VM $vm  -ResourceGroupName $resourceGroupName
```

Setting Key Vault Access policy for the Azure AD Application

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. You can read this blog post for some examples on that. Below you also have an example of how to perform this task via PowerShell:

```
$keyVaultName = '<yourKeyVaultName>'
$aadClientID = '<yourAadAppClientID>'
$rgname = '<yourResourceGroup>'
Set-AzureRmKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName $aadClientID -PermissionsToKeys
'WrapKey' -PermissionsToSecrets 'Set' -ResourceGroupName $rgname
```

NOTE: Azure disk encryption requires you to configure the following access policies to your AAD Client Application - 'WrapKey' and 'Set' permissions

Terminology

Use the terminology table as reference to understand some of the common terms used by this technology:

TERMINOLOGY	DEFINITION
Azure AD	Azure AD is Azure Active Directory. Azure AD account is a pre-requisite for authenticating, storing, and retrieving secrets from the Key Vault.
Azure Key Vault [AKV]	Azure Key Vault is a cryptographic key management service based on FIPS-validated Hardware Security Modules to safeguard your cryptographic keys and sensitive secrets securely.,Refer to Key Vault documentation for more details.
ARM	Azure Resource Manager
BitLocker	BitLocker is an industry recognized Windows volume encryption technology used to enable disk encryption on Windows IaaS VMs
BEK	BitLocker Encryption Keys are used to encrypt the OS boot volume and data volumes. The BitLocker keys are safeguard in customer’s Azure key vault as secrets.
CLI	Azure Command-Line Interface
DM-Crypt	DM-Crypt is the Linux-based transparent disk encryption subsystem used to enable disk encryption on Linux IaaS VMs

TERMINOLOGY	DEFINITION
KEK	Key Encryption Key is the asymmetric key (RSA 2048) used to protect or wrap the secret if desired. You can provide an HSM-protected key or software-protected key. For more details, refer to Azure Key Vault documentation for more details
PS cmdlets	Azure PowerShell cmdlets

Setting and Configuring Azure Key Vault for Azure disk encryption usage

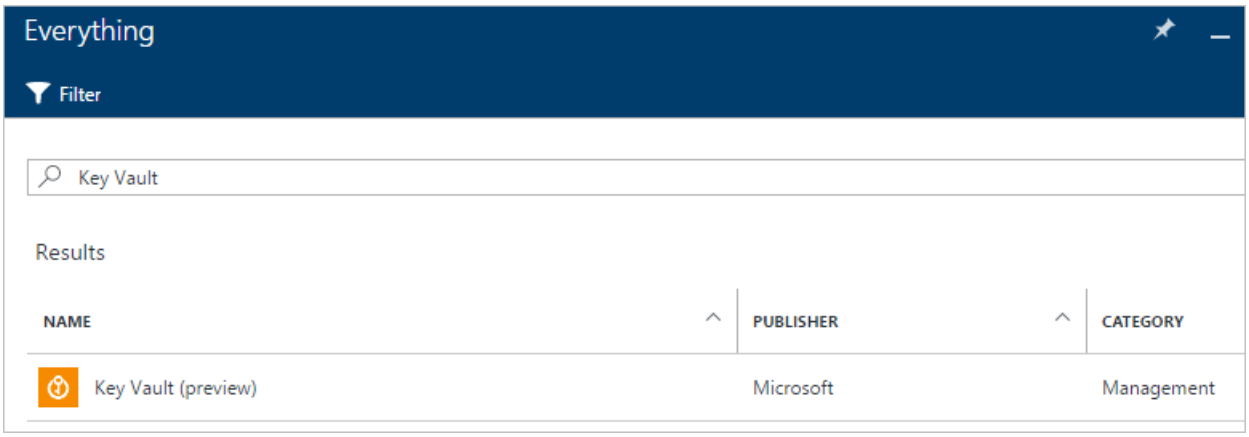
Azure disk encryption safeguards the disk encryption keys and secrets in your Azure Key Vault. Follow the steps on each one of the sections below to setup Key Vault for Azure disk encryption usage.

Create a New Key Vault

To create a new Key Vault, use one of the options listed below:

- Use the "101-Create-KeyVault" Resource Manager template located here
- Use the Azure PowerShell Key Vault cmdlets.
- Use the Azure resource manager portal.

Note: If you already have a Key Vault setup for your subscription, please proceed to next section.

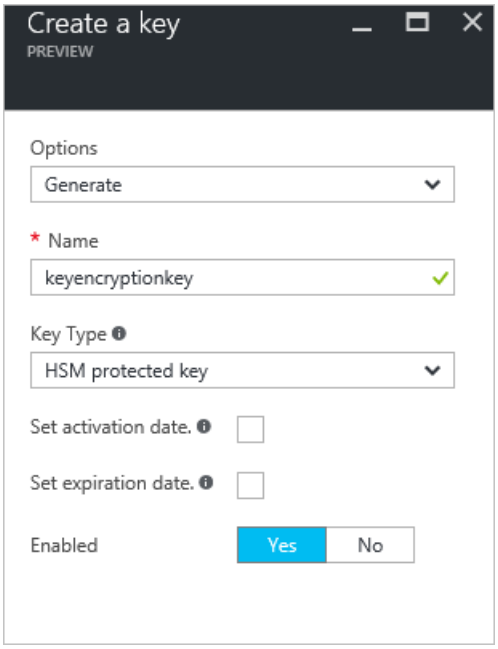


Provisioning a Key Encryption Key (optional)

If you wish to use a Key Encryption Key (KEK) for an additional layer of security to wrap the BitLocker encryption keys, you should add a KEK to your Key Vault for use in the provisioning process. Use the Add-AzureKeyVaultKey cmdlet to create a new Key Encryption Key in Key Vault. You can also import KEK from your on-premises key management HSM. For more details, see Key Vault documentation.

```
Add-AzureKeyVaultKey [-VaultName] <string> [-Name] <string> -Destination <string> {HSM | Software}
```

The KEK can be added from Azure Resource Manager portal as well using Azure Key Vault UX.



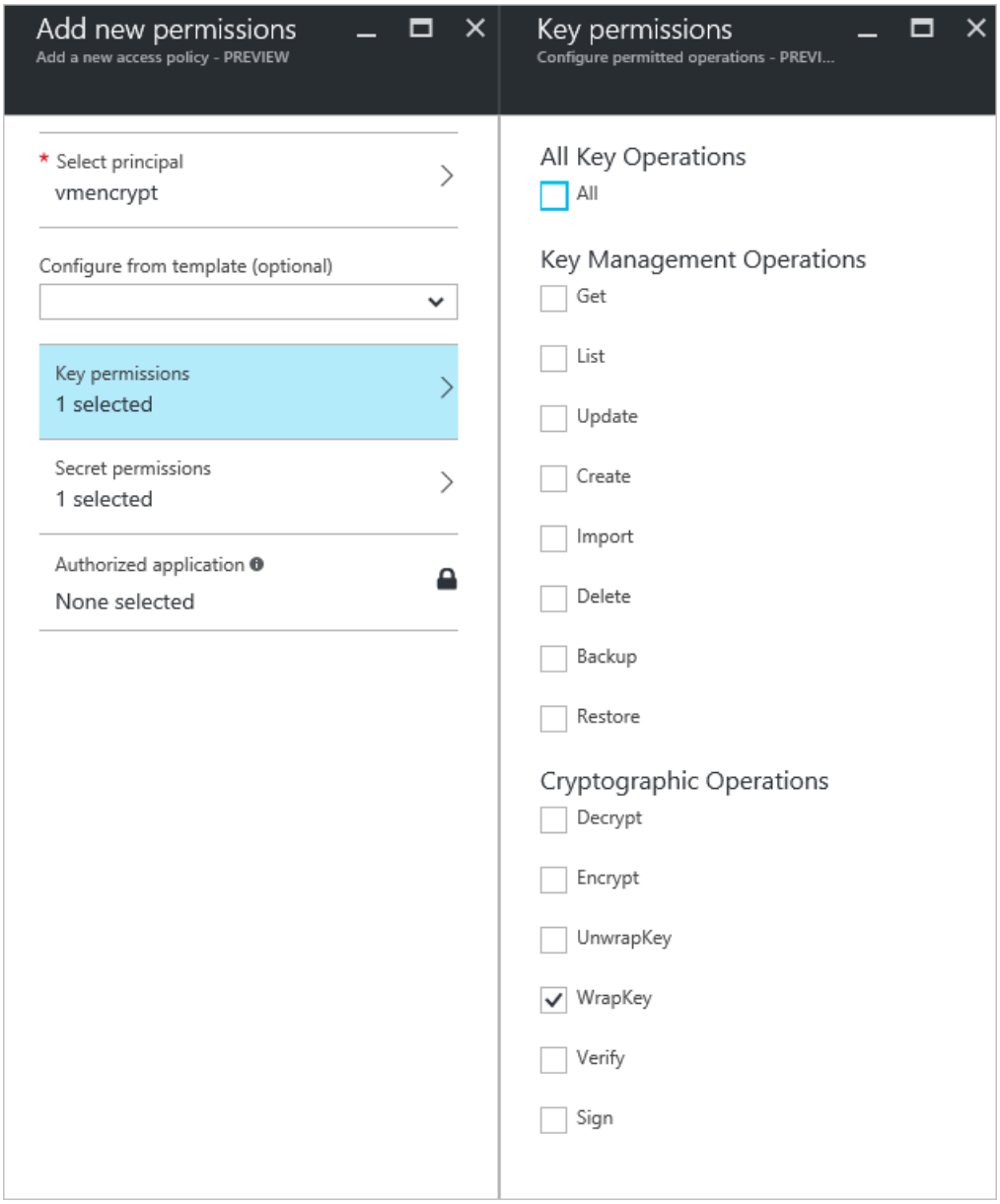
Set Key Vault permissions to allow the Azure platform access to the keys and secrets

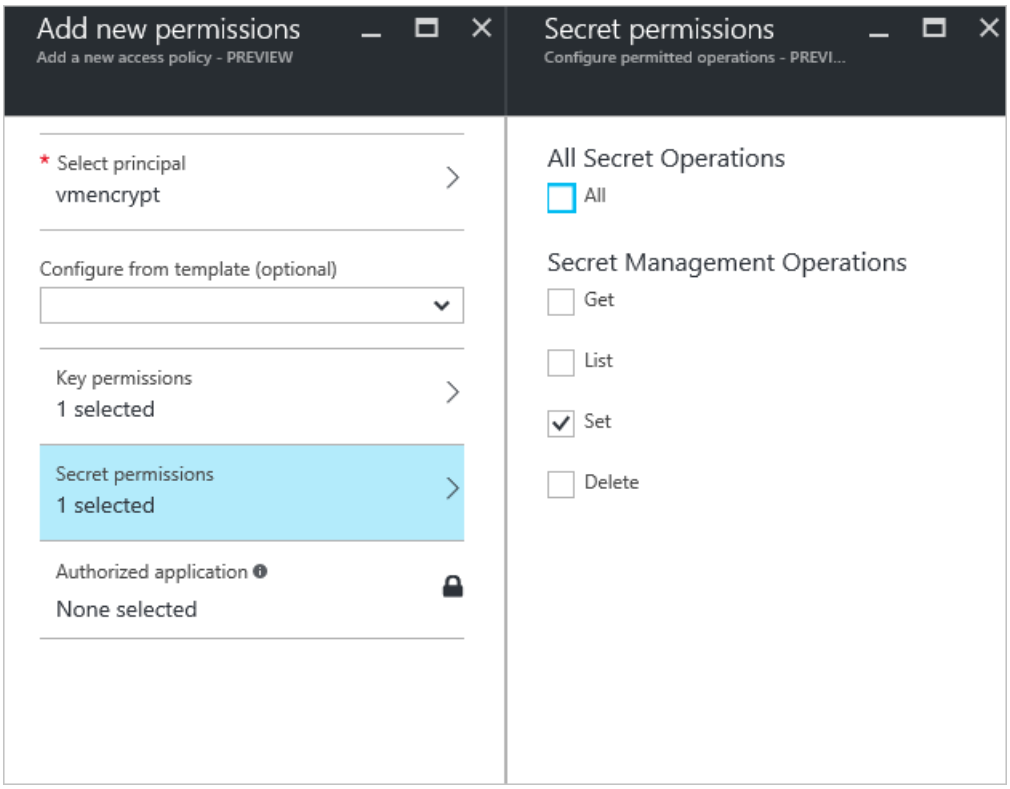
The Azure platform needs access to the encryption keys or secrets in your Azure Key Vault in order to make them available to the VM to boot and decrypt the volumes. To grant permissions to the Azure platform so that it can access the Key Vault, the *enabledForDiskEncryption* property must be set on the Key Vault. You can set the enabledForDiskEncryption property on your key vault using the key vault PS cmdlet:

```
Set-AzureRmKeyVaultAccessPolicy -VaultName <yourVaultName> -ResourceGroupName <yourResourceGroup> -
EnabledForDiskEncryption
```

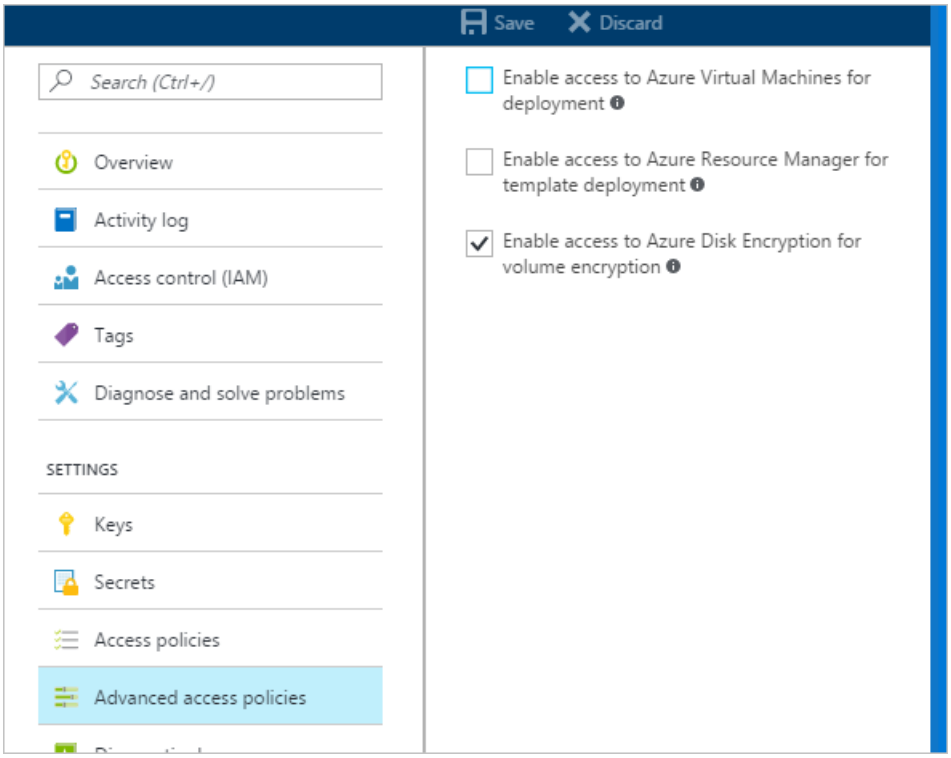
You can also set the *enabledForDiskEncryption* property by visiting https://resources.azure.com. You must set the *enabledForDiskEncryption* property on your Key Vault as mentioned before. Otherwise the deployment will fail.

You can setup access policies for your AAD application from the Key Vault UX:





Make sure that Key Vault is enabled for Disk Encryption in "Advanced Access Policies":



Disk Encryption deployment scenarios and user experiences

There are many scenarios that you can enable disk encryption and the steps may vary according to the scenario. The sections that follow will cover in more details these scenarios.

Enable encryption on new IaaS VM's created from the Azure Gallery

Disk encryption can be enabled on new IaaS Windows VM from Azure gallery in Azure using the Resource Manager template published here. Click on "Deploy to Azure" button on the Azure quickstart template, input encryption configuration in the parameters blade and click OK. Select the subscription, resource group, resource group location, legal terms and agreement and click Create button to enable encryption on a new IaaS VM.

Note: This template creates a new encrypted Windows VM using the Windows Server 2012 gallery image.

Disk encryption can be enabled on a new IaaS RedHat Linux 7.2 VM with a 200 GB RAID-0 array using this resource manager template. After the template is deployed, verify the VM encryption status using the `Get-AzureRmVmDiskEncryptionStatus` cmdlet as described in the section "Encrypting OS drive on a running Linux VM". When the machine returns status `VMRestartPending`, restart the VM.

You can see the Resource Manager template parameters details for new VM from Azure gallery scenario using Azure AD Client ID in the table below:

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 to which the VM NIC should belong to.

PARAMETER	DESCRIPTION
subnetName	Name of the subnet in the vNet to which the VM NIC should belong to
AADClientID	Client ID of the Azure AD app that has permissions to write secrets to Key Vault
AADClientSecret	Client Secret of the Azure AD app that has permissions to write secrets to Key Vault
keyVaultURL	URL of the Key Vault to which BitLocker key should be uploaded to. You can get it using the cmdlet: (Get-AzureRmKeyVault -VaultName,-ResourceGroupName).VaultURI
keyEncryptionKeyURL	URL of the Key Encryption Key that's used to encrypt the generated BitLocker key. This is optional.
keyVaultResourceGroup	Resource Group of the key vault
vmName	Name of the VM on which encryption operation is to be performed

Note: KeyEncryptionKeyURL is an optional parameter. You can bring your own KEK to further safeguard the data encryption key (Passphrase secret) in Key Vault.

Enable encryption on new IaaS VM's 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 sections below will explain in more details the Resource Manager template and CLI commands.

Follow the instructions from one of these sections for preparing pre-encrypted images that can be used in Azure. Once the image is created, the steps in the next section can be used for creating an encrypted Azure VM.

- Preparing a pre-encrypted Windows VHD
- Preparing a pre-encrypted Linux VHD

Using Resource Manager template

Disk encryption can be enabled on customer encrypted VHD using the Resource Manager template published here. Click on "Deploy to Azure" button on the Azure quickstart template, input encryption configuration in the parameters blade and click OK. Select the subscription, resource group, resource group location, legal terms and agreement and click Create button to enable encryption on new IaaS VM.

The Resource Manager template parameters details for customer encrypted VHD scenario are described in the table below:

PARAMETER	DESCRIPTION
newStorageAccountName	Name of the storage account to store encrypted OS vhd. This storage account should have already been created in the same resource group and same location as the VM
osVhdUri	URI of OS vhd from storage account
osType	OS product type (Windows/Linux)

PARAMETER	DESCRIPTION
virtualNetworkName	Name of the VNet to which the VM NIC should belong to. This should have been already created in the same resource group and same location as the VM
subnetName	Name of the subnet in the vNet to which the VM NIC should belong to
vmSize	Size of the VM. Currently, only Standard A, D and G series are supported
keyVaultResourceID	ResourceID identifying the key vault resource in ARM. You can get it using the PowerShell cmdlet: (Get-AzureRmKeyVault -VaultName <yourKeyVaultName> -ResourceGroupName <yourResourceGroupName>).ResourceId
keyVaultSecretUrl	URL of the disk encryption key provisioned in key vault
keyVaultKekUrl	URL of the Key Encryption Key that's to encrypt the generated disk encryption key
vmName	Name of the IaaS VM

Using PowerShell cmdlets

Disk encryption can be enabled on customer encrypted VHD using the PS cmdlets published here.

Using CLI Commands

Follow the steps below to enable disk encryption for this scenario using CLI commands:

1. Set access policies on Key Vault:
   - Set 'EnabledForDiskEncryption' flag:
     ```
     azure keyvault set-policy --vault-name <keyVaultName> --enabled-for-disk-encryption true
     ```
   - Set permissions to Azure AD app to write secrets to KeyVault:
     ```
     azure keyvault set-policy --vault-name <keyVaultName> --spn <aadClientID> --perms-to-keys [\"all\"] --perms-to-secrets [\"all\"]
     ```
2. To enable encryption on an existing/running VM, type: *azure vm enable-disk-encryption --resource-group --name --aad-client-id --aad-client-secret --disk-encryption-key-vault-url --disk-encryption-key-vault-id*
3. Get encryption status: *"azure vm show-disk-encryption-status --resource-group --name --json"*
4. To enable encryption on a new VM from customer encrypted VHD, use the below parameters with "azure vm create" command:
   - disk-encryption-key-vault-id
   - disk-encryption-key-url
   - key-encryption-key-vault-id
   - key-encryption-key-url

Enable encryption on existing or running IaaS Windows VM in Azure

In this scenario you can enable encrypting by using the Resource Manager template, PowerShell cmdlets or CLI commands. The sections below will explain in more details how to enable it using Resource Manager template and CLI commands.

Using Resource Manager template

Disk encryption can be enabled on existing/running IaaS Windows VM in Azure using the Resource Manager template published here. Click on "Deploy to Azure" button on the Azure quickstart template, input encryption

configuration in the parameters blade and click OK. Select the subscription, resource group, resource group location, legal terms and agreement and click Create button to enable encryption on existing/running IaaS VM.

The Resource Manager template parameters details for existing/running VM scenario using Azure AD Client ID are available in the table below:

PARAMETER	DESCRIPTION
AADClientID	Client ID of the Azure AD app that has permissions to write secrets to Key Vault
AADClientSecret	Client Secret of the Azure AD app that has permissions to write secrets to Key Vault
keyVaultName	Name of the Key Vault to which BitLocker key should be uploaded to. You can get it using the cmdlet: (Get-AzureRmKeyVault -ResourceGroupName). Vaultname
keyEncryptionKeyURL	URL of the Key Encryption Key that's used to encrypt the generated BitLocker key. This is optional if you select `nokek` in the UseExistingKek dropdown. If you select `kek` in the UseExistingKek dropdown, you must input the keyEncryptionKeyURL value
volumeType	Type of the volume on which encryption operation is performed. Valid values are "OS", "Data" , "All"
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 on which encryption operation is to be performed

Note: KeyEncryptionKeyURL is an optional parameter. You can bring your own KEK to further safeguard the data encryption key (BitLocker encryption secret) in Key Vault.

Using PowerShell cmdlets

Refer to the **Explore Azure disk encryption with Azure PowerShell** blog post part 1 and part 2 for details on how to enable encryption using Azure Disk Encryption using PS cmdlets.

Using CLI Commands

Follow the steps below to enable encryption on existing/running IaaS Windows VM in Azure using CLI commands:

1. Set access policies on Key Vault:
   - Set 'EnabledForDiskEncryption' flag: "azure keyvault set-policy --vault-name --enabled-for-disk-encryption true"
   - Set permissions to Azure AD app to write secrets to KeyVault: "azure keyvault set-policy --vault-name --spn --perms-to-keys [\"all\"] --perms-to-secrets [\"all\"]"
2. To enable encryption on an existing/running VM, type: *azure vm enable-disk-encryption --resource-group --name --aad-client-id --aad-client-secret --disk-encryption-key-vault-url --disk-encryption-key-vault-id*
3. Get encryption status: *"azure vm show-disk-encryption-status --resource-group --name --json"*
4. To enable encryption on a new VM from customer encrypted VHD, use the below parameters with "azure vm create" command:
   - disk-encryption-key-vault-id
   - disk-encryption-key-url

- key-encryption-key-vault-id
- key-encryption-key-url

Enable encryption on existing or running IaaS Linux VM in Azure

Disk encryption can be enabled on existing/running IaaS Linux VM in Azure using the Resource Manager template published here. Click on "Deploy to Azure" button on the Azure quickstart template, input encryption configuration in the parameters blade and click OK. Select the subscription, resource group, resource group location, legal terms and agreement and click Create button to enable encryption on existing/running IaaS VM.

The Resource Manager template parameters details for existing/running VM scenario using Azure AD Client ID are described in the table below:

PARAMETER	DESCRIPTION
AADClientID	Client ID of the Azure AD app that has permissions to write secrets to Key Vault
AADClientSecret	Client Secret of the Azure AD app that has permissions to write secrets to Key Vault
keyVaultName	Name of the Key Vault to which BitLocker key should be uploaded to. You can get it using the cmdlet: (Get-AzureRmKeyVault -ResourceGroupName). Vaultname
keyEncryptionKeyURL	URL of the Key Encryption Key that's used to encrypt the generated BitLocker key. This is optional if you select "nokek" in the UseExistingKek dropdown. If you select "kek" in the UseExistingKek dropdown, you must input the keyEncryptionKeyURL value
volumeType	Type of the volume on which encryption operation is performed. Valid supported values are "OS"/"All" (for RHEL 7.2, CentOS 7.2 & Ubuntu 16.04) and "Data" for all other distros.
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 on which encryption operation is to be performed
passPhrase	Type a strong passphrase as the data encryption key

Note: KeyEncryptionKeyURL is an optional parameter. You can bring your own KEK to further safeguard the data encryption key (Passphrase secret) in Key Vault.

CLI Commands

Disk encryption can be enabled on customer encrypted VHD using the CLI command installed from here. Follow the steps below to enable encryption on existing/running IaaS Linux VM in Azure using CLI commands:

1. Set access policies on Key Vault:
   - Set 'EnabledForDiskEncryption' flag: "azure keyvault set-policy --vault-name --enabled-for-disk-encryption true"
   - Set permissions to Azure AD app to write secrets to KeyVault: "azure keyvault set-policy --vault-name --spn --perms-to-keys [\"all\"] --perms-to-secrets [\"all\"]"

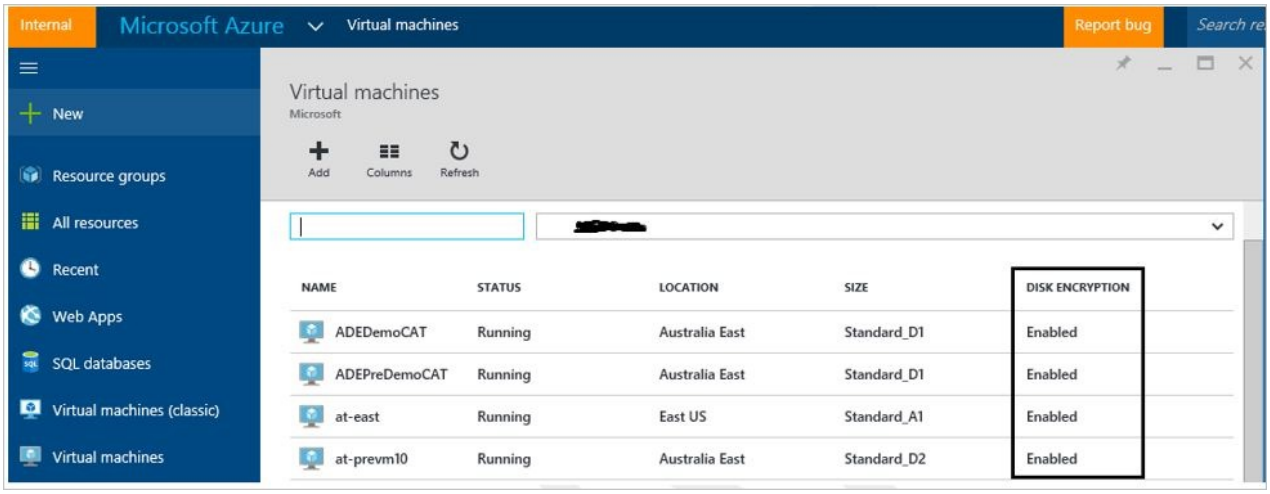
2. To enable encryption on an existing/running VM, type: *azure vm enable-disk-encryption --resource-group --name --aad-client-id --aad-client-secret --disk-encryption-key-vault-url --disk-encryption-key-vault-id*
3. Get encryption status: "azure vm show-disk-encryption-status --resource-group --name --json"
4. To enable encryption on a new VM from customer encrypted VHD, use the below parameters with "azure vm create" command.
   - *disk-encryption-key-vault-id*
   - *disk-encryption-key-url*
   - *key-encryption-key-vault-id*
   - *key-encryption-key-url*

Get encryption status of an encrypted IaaS VM

You can get encryption status using Azure Resource Manager portal, PowerShell cmdlets or CLI commands. The sections below will explain how to use the Azure portal and CLI commands to get the encryption status.

Get encryption status of an encrypted Windows VM using Azure Resource Manager portal

You can get the encryption status of the IaaS VM from Azure Resource Manager portal. Logon to Azure portal at https://portal.azure.com/, click on virtual machines link in the left menu to see summary view of the virtual machines in your subscription. You can filter the virtual machines view by selecting the subscription name from the subscription dropdown. Click on columns located at the top of the virtual machines page menu. Select Disk Encryption column from the choose column blade and click update. You should see the disk encryption column showing the encryption state "Enabled" or "Not Enabled" for each VM as shown in the figure below.



Get encryption status of an encrypted (Windows/Linux) IaaS VM using disk encryption PS Cmdlet

You can get the encryption status of the IaaS VM from disk encryption PS cmdlet "Get-AzureRmVMDiskEncryptionStatus". To get the encryption settings for your VM, type in your Azure PowerShell session:

```
C:\> Get-AzureRmVmDiskEncryptionStatus  -ResourceGroupName $ResourceGroupName -VMName $VMName
-ExtensionName $ExtensionName

OsVolumeEncrypted          : NotEncrypted
DataVolumesEncrypted       : Encrypted
OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : https://rheltest1keyvault.vault.azure.net/secrets/bdb6bfb1-5431-4c28-af46-
b18d0025ef2a/abebacb83d864a5fa729508315020f8a
```

The output of Get-AzureRmVMDiskEncryptionStatus can be inspected for encryption key URLs.

```
C:\> $status = Get-AzureRmVmDiskEncryptionStatus  -ResourceGroupName $ResourceGroupName -VMNam
e $VMName -ExtensionName $ExtensionName
C:\> $status.OsVolumeEncryptionSettings

DiskEncryptionKey                                               KeyEncryptionKey
Enabled
-----------------                                               ----------------
-------
Microsoft.Azure.Management.Compute.Models.KeyVaultSecretReference
Microsoft.Azure.Management.Compute.Models.KeyVaultKeyReference    True


C:\> $status.OsVolumeEncryptionSettings.DiskEncryptionKey.SecretUrl
https://rheltest1keyvault.vault.azure.net/secrets/bdb6bfb1-5431-4c28-af46-
b18d0025ef2a/abebacb83d864a5fa729508315020f8a
C:\> $status.OsVolumeEncryptionSettings.DiskEncryptionKey

SecretUrl
SourceVault
---------
-----------
https://rheltest1keyvault.vault.azure.net/secrets/bdb6bfb1-5431-4c28-af46-
b18d0025ef2a/abebacb83d864a5fa729508315020f8a Microsoft.Azure.Management....
```

The OSVolumeEncrypted and DataVolumesEncrypted settings value are set to "Encrypted" showing that both the volumes are encrypted using Azure disk encryption. Refer to the **Explore Azure disk encryption with Azure PowerShell** blog post part 1 and part 2 for details on how to enable encryption using Azure Disk Encryption using PS cmdlets.

NOTE: On Linux VMs, the `Get-AzureRmVMDiskEncryptionStatus` cmdlet takes 3-4 minutes to report the encryption status.

Get encryption status of the IaaS VM from disk encryption CLI command

You can get the encryption status of the IaaS VM from disk encryption CLI command *azure vm show-disk-encryption-status*. To get the encryption settings for your VM, type in your Azure CLI session:

```
azure vm show-disk-encryption-status --resource-group <yourResourceGroupName> --name <yourVMName> --json
```

Disable Encryption on running Windows IaaS VM

You can disable encryption on a running Windows or Linux IaaS VM via the Azure disk encryption Resource Manager template or PS cmdlets and specifies the decryption configuration.

Windows VM

The disable encryption step disables encryption of the OS or data volume or both on the running Windows IaaS VM. You cannot disable the OS volume and leave the data volume encrypted. When the disable encryption step is performed, Azure classic deployment model updates the VM service model and the Windows IaaS VM is marked decrypted. The contents of the VM are not encrypted at rest anymore. The disable encryption does not delete the customer key vault and the encryption key material, which is BitLocker Encryption Keys for Windows and Passphrase for Linux.

Linux VM

The disable encryption step disables encryption of the data volume on the running Linux IaaS VM

NOTE: Disabling encryption on OS disk is not allowed on Linux VMs.

Disable encryption on existing/running IaaS VM in A

Disk encryption can be disabled on running Windows IaaS VM using the Resource Manager template published here. Click on “Deploy to Azure” button on the Azure quickstart template, input decryption configuration in the parameters blade and click OK. Select the subscription, resource group, resource group location, legal terms and agreement and click Create button to enable encryption on a new IaaS VM.

For Linux VM, this template can be used to disable encryption.

Resource Manager template parameters details for disabling encryption on running IaaS VM:

VMNAME	NAME OF THE VM ON WHICH ENCRYPTION OPERATION IS TO BE PERFORMED
volumeType	Type of the volume on which decryption operation is performed. Valid values are "OS", "Data", "All". **Note:** You cannot disable encryption on running Windows IaaS VM OS/boot volume without disabling encryption on "Data" volume. **Note**: Disabling encryption on OS disk is not allowed on Linux VMs.
sequenceVersion	Sequence version of the BitLocker operation. Increment this version number every time a disk decryption operation is performed on the same VM

Disable encryption on existing/running IaaS VM in A

To disable using the PS cmdlet, Disable-AzureRmVMDiskEncryption cmdlet disables encryption on an infrastructure as a service (IaaS) virtual machine. This cmdlet supports both Windows and Linux VMs. This cmdlet installs an extension on the virtual machine to disable encryption. If the Name parameter is not specified, an extension with the default name "AzureDiskEncryption for Windows VMs" is created.

On Linux VMs, the "AzureDiskEncryptionForLinux" extension is used.

Note: This cmdlet reboots the virtual machine.

Appendix

Connect to your subscription

Make sure to review the *Prerequisites* section in this document before proceeding. After ensuring that all prerequisites were fulfilled, follow the steps below to connect to your subscription:

1.Start an Azure PowerShell session and sign in to your Azure account with the following command:

```
Login-AzureRmAccount
```

2.If you have multiple subscriptions and want to specify a specific 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 the subscription configured is correct, type:

```
Get-AzureRmSubscription
```

5.To confirm the Azure Disk Encryption cmdlets are installed, type:

```
Get-command *diskencryption*
```

6.You should see the below output confirming Azure Disk Encryption PowerShell installation:

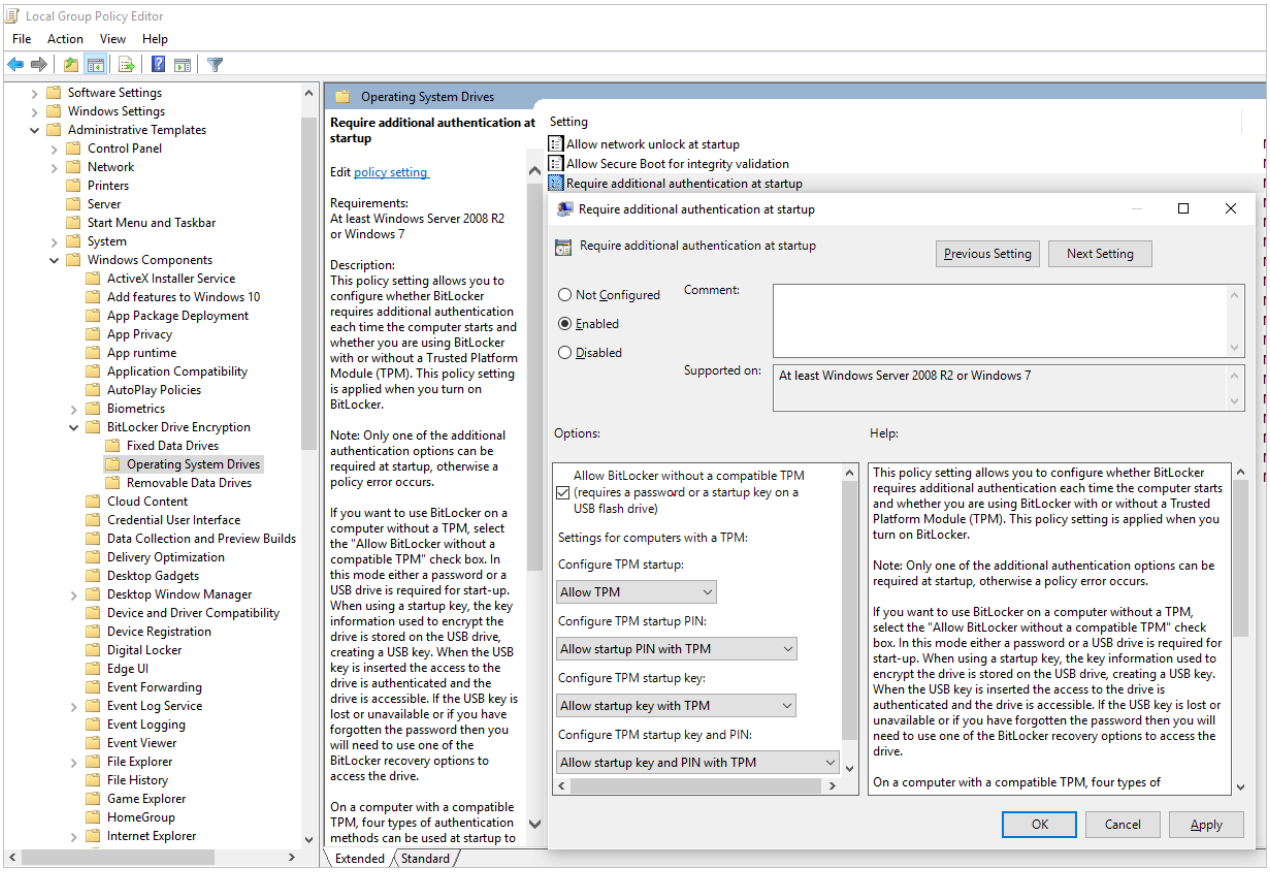
```
PS C:\Windows\System32\WindowsPowerShell\v1.0> get-command *diskencryption*
CommandType   Name                                        Source
Cmdlet        Get-AzureRmVMDiskEncryptionStatus           AzureRM.Compute
Cmdlet        Disable-AzureRmVMDiskEncryption             AzureRM.Compute
Cmdlet        Set-AzureRmVMDiskEncryptionExtension        AzureRM.Compute
```

Preparing a pre-encrypted Windows VHD

The sections that follow are necessary in order to prepare a pre-encrypted Windows VHD for deployment as an encrypted VHD in Azure IaaS. The steps are used to prepare and boot a fresh windows VM (vhd) on Hyper-V or Azure.

Update group policy to allow non-TPM for OS protection

You need to configure the BitLocker Group Policy setting called BitLocker Drive Encryption, located 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 figure below:



Install BitLocker feature components

For Windows Server 2012 and above use the below command:

```
dism /online /Enable-Feature /all /FeatureName:Bitlocker /quiet /norestart
```

For Windows Server 2008 R2 use the below command:

```
ServerManagerCmd -install BitLockers
```

Prepare OS volume for BitLocker using `bdehdcfg`

Execute the command below to compress the OS partition and prepare the machine for BitLocker.

```
bdehdcfg -target c: shrink -quiet
```

Using BitLocker to protect the OS volume

Use the `manage-bde` command to enable encryption on the boot volume using an external key protector and place the external key (.bek file) on the external drive or volume. Encryption will be enabled on the system/boot volume after the next reboot.

```
manage-bde -on %systemdrive% -sk [ExternalDriveOrVolume]
reboot
```

Note: The VM needs to be prepared with a separate data/resource vhd for getting the external key using BitLocker.

Encrypting OS drive on a running Linux VM

Encryption of OS drive on a running Linux VM is supported on the following distros:

- RHEL 7.2
- CentOS 7.2
- Ubuntu 16.04

Prerequisites for OS disk encryption:

- VM must be created from Azure Gallery image in Azure Resource Manager portal.
- Azure VM with at least 4 GB of RAM (recommended size is 7 GB).
- (For RHEL and CentOS) SELinux must be disabled on the VM. The VM must be rebooted at least once after disabling SELinux.

S t e p s

1.Create a VM using one of the distros specified above.

For CentOS 7.2, OS disk encryption is supported via a special image. To use this image, specify "7.2n" as the Sku when creating 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 are going to encrypt all the (OS + data) drives the data drives need to be specified and mountable from /etc/fstab.

> **NOTE**
> You must use UUID=... to specify data drives in /etc/fstab instead of specifying the block device name, e.g., /dev/sdb1. During encryption the order of drives will change on the VM. If your VM relies on a specific order of block devices it will fail to mount them after encryption.

3.Logout SSH sessions.

4.To encrypt the OS, specify volumeType as "All" or "OS" when enabling encryption.

> **NOTE**
> All user-space processes that are not running as `systemd` services shall be killed with a `SIGKILL`. The VM shall be rebooted. Please plan on downtime of the VM when enabling OS disk encryption on a running VM.

5.Periodically monitor the progress of encryption using instructions in the next section.

6.Once Get-AzureRmVmDiskEncryptionStatus shows "VMRestartPending", restart your VM by either logging on to it or via Portal/PowerShell/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, please reboot the VM
```

It is recommended to save boot diagnostics of the VM *before* rebooting.

Monitoring OS encryption progress

There are three ways to monitor OS encryption progress.

1.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
```

Once the VM reaches "OS disk encryption started" it will take roughly 40-50 minutes on a Premium-storage backed VM.

Due to issue #388 in WALinuxAgent, `OsVolumeEncrypted` and `DataVolumesEncrypted` show up as `Unknown` in some distros. With WALinuxAgent version 2.1.5 and above this will be fixed automatically. In case you see `Unknown` in the output, you can verify disk encryption status by using Azure Resource Viewer.

Go to Azure Resource Viewer, 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.

2.Look at boot diagnostics. Messages from ADE extension shall be prefixed with `[AzureDiskEncryption]`.

3.Logon on to the VM via SSH and getting the extension log from

```
/var/log/azure/Microsoft.Azure.Security.AzureDiskEncryptionForLinux
```

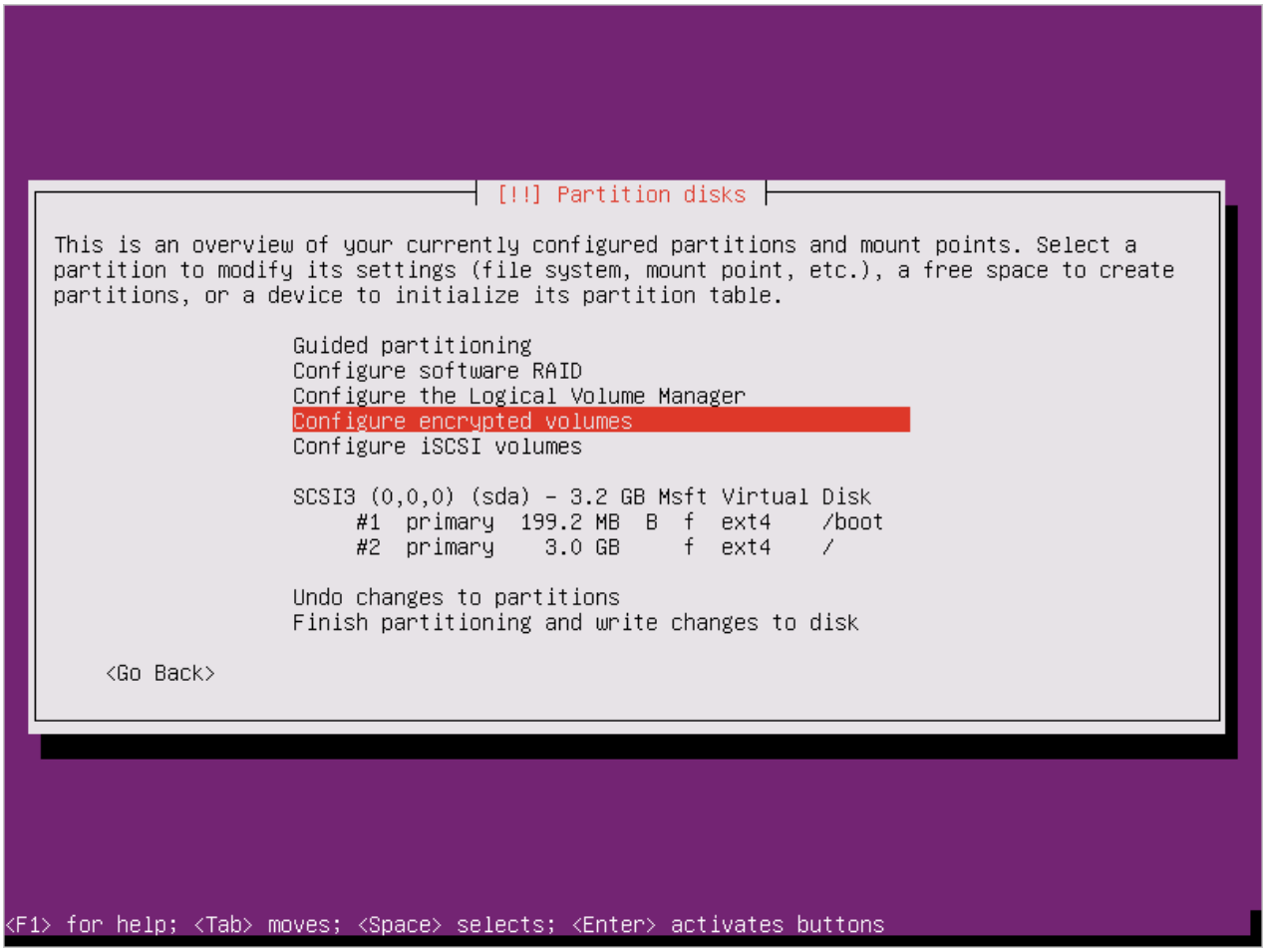
It is not recommended to log on to the VM while OS encryption is in progress. Therefore, the logs should be copied only when other two methods have failed.

Preparing a pre-encrypted Linux VHD

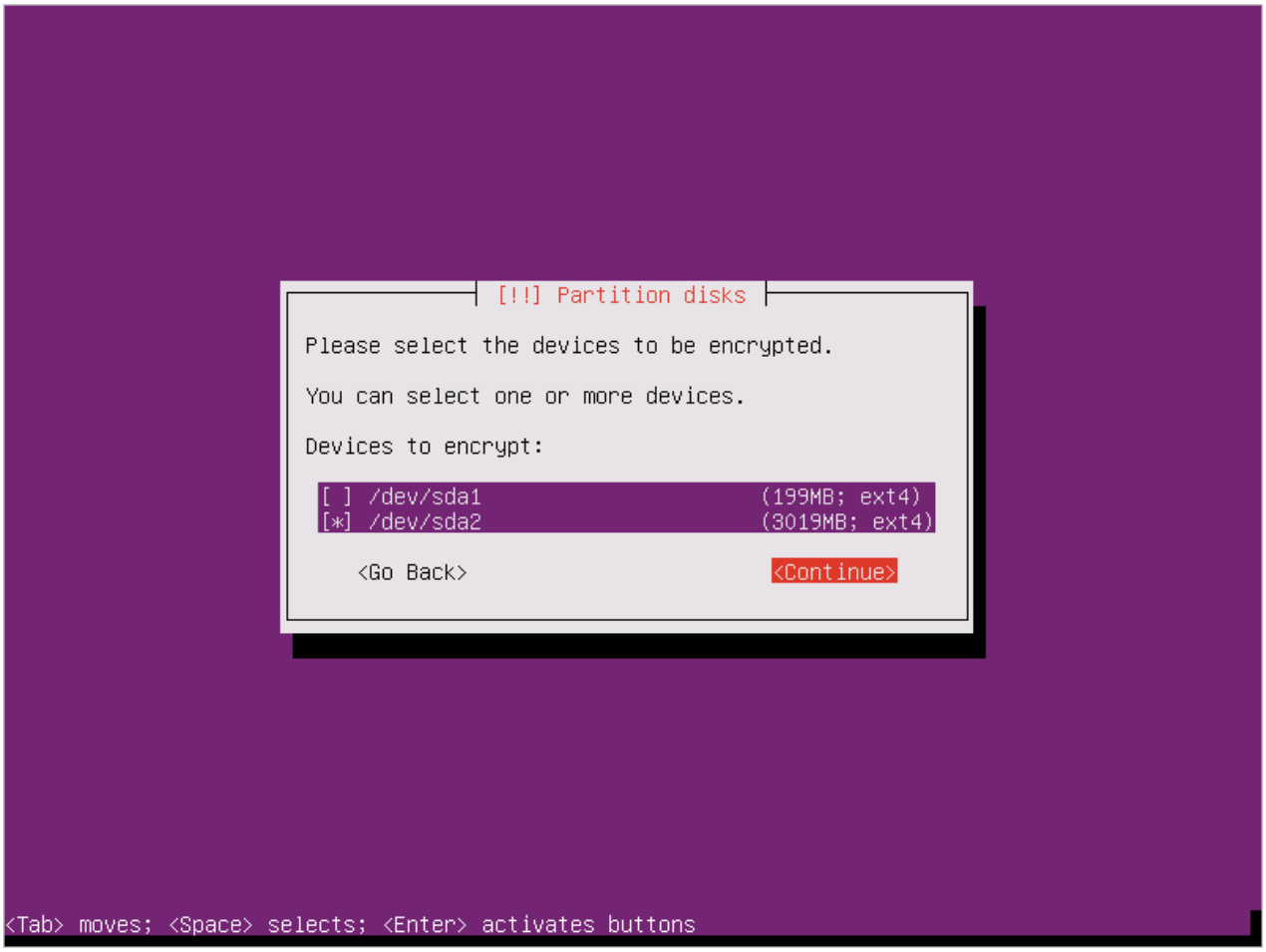
Ubuntu 16

Configure encryption during distro install

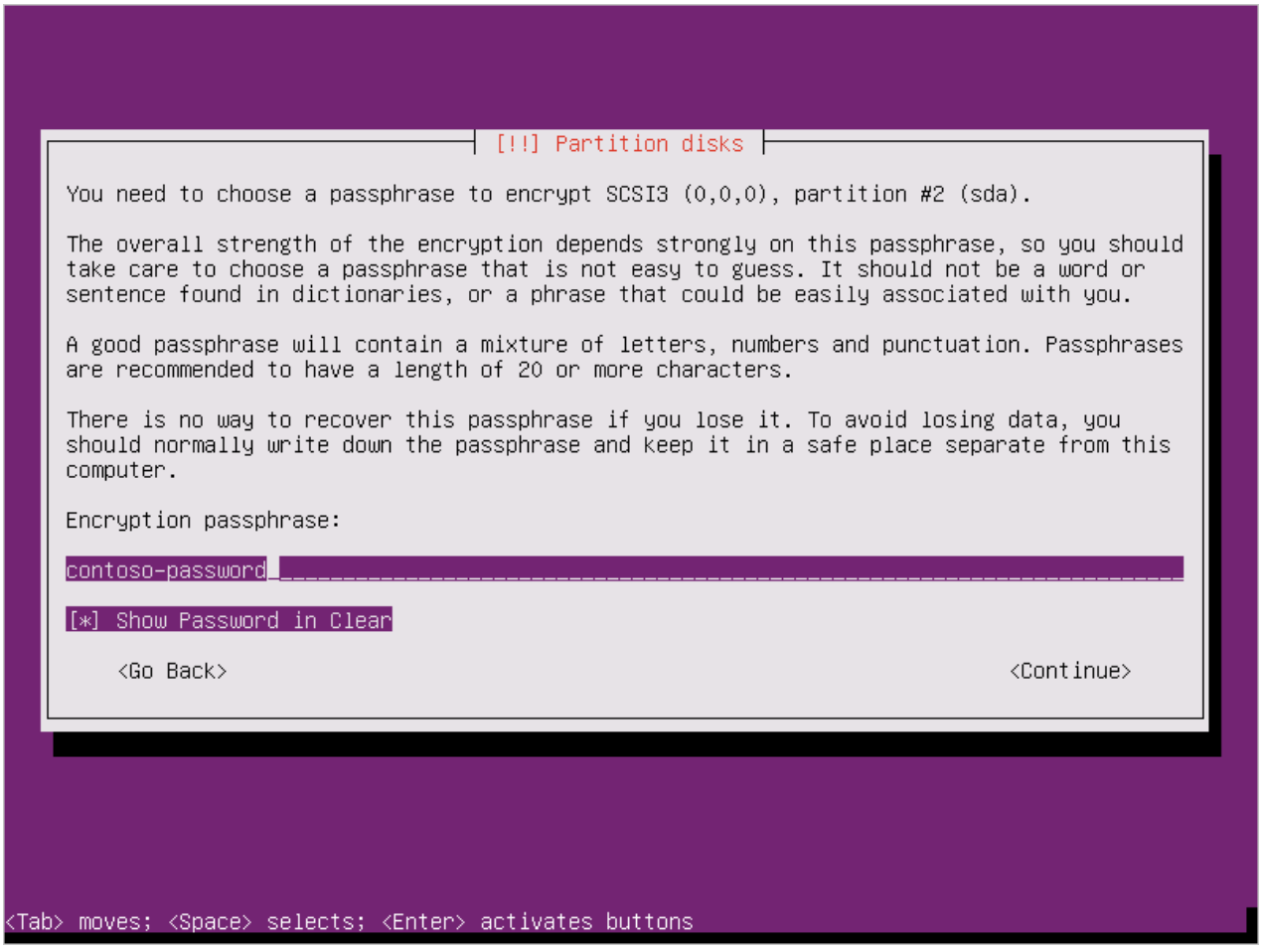
1.Select "Configure encrypted volumes" when partitioning disks.



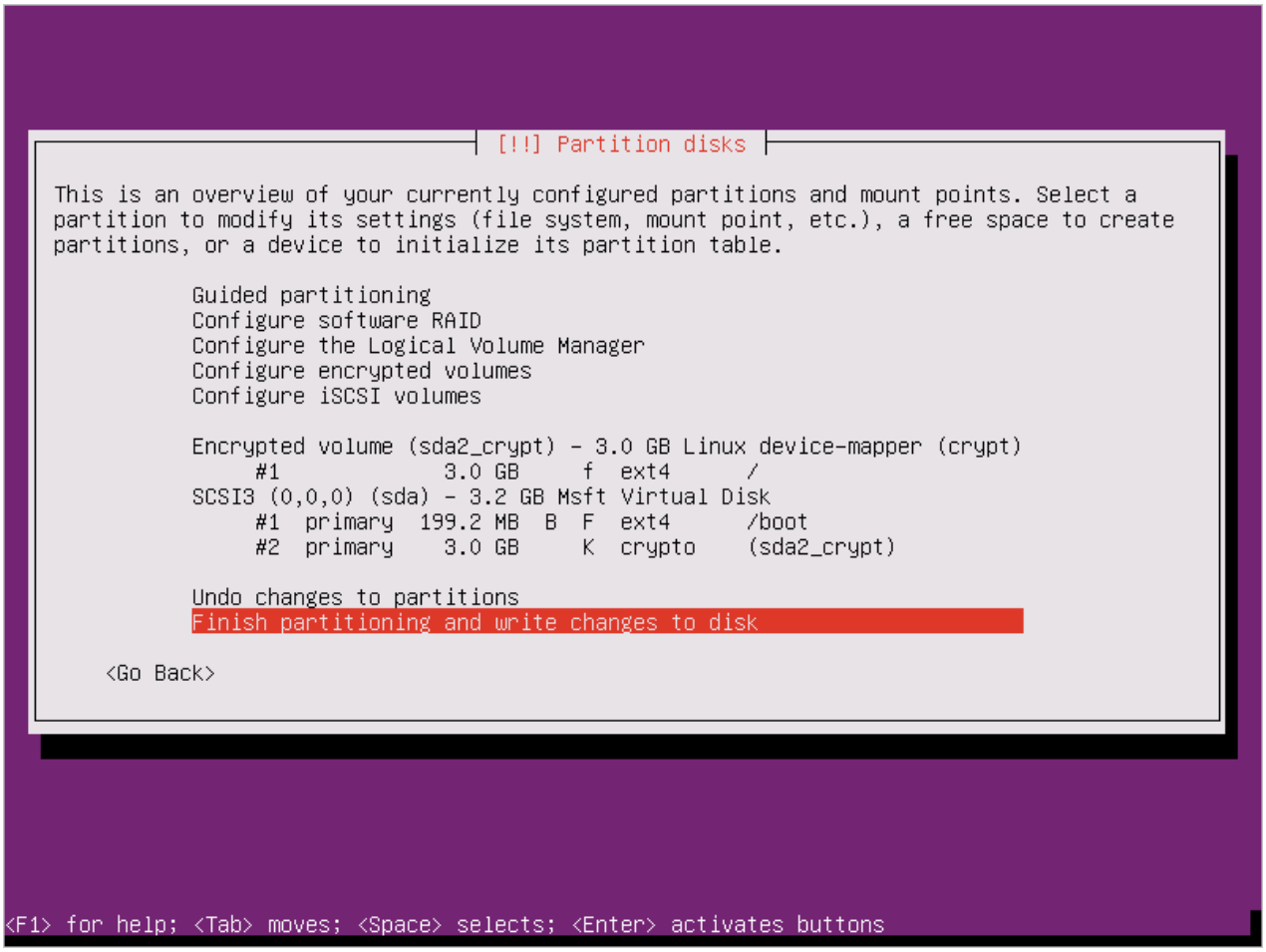
2.Create a separate boot drive which must not be encrypted. Encrypt your root drive.



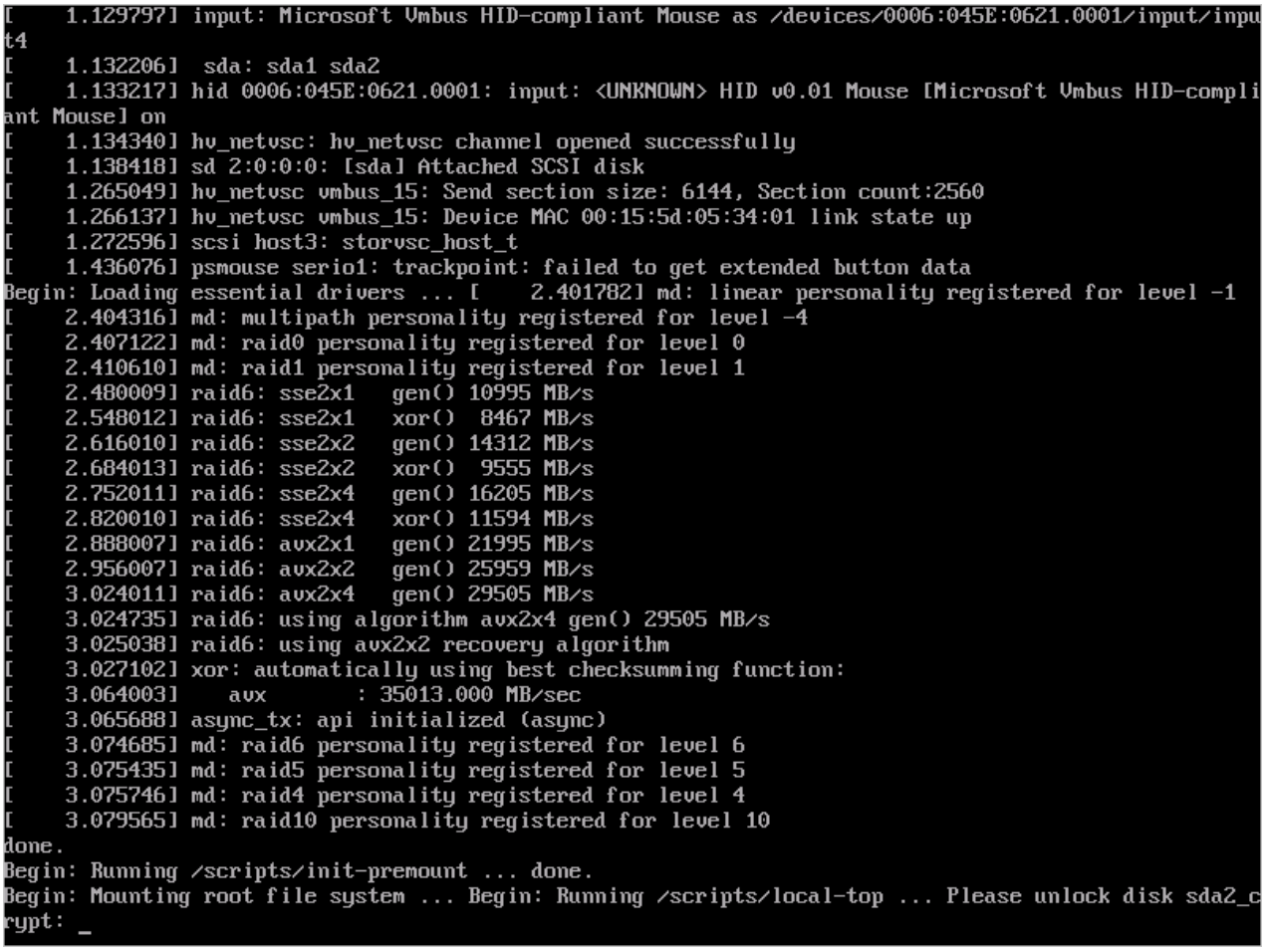
3.Provide a passphrase. This is the passphrase that you will upload into KeyVault.



4.Finish partitioning.



5.When booting the VM, you will be asked for a passphrase. Use the passphrase you provided in step 3.



6.Prepare VM for uploading into Azure using these instructions. Do not run the last step (deprovisioning the VM)

yet.

Configure encryption to work with Azure

1.Create a file under /usr/local/sbin/azure_crypt_key.sh, with the content in the script below. Pay attention to the KeyFileName, because it is the passphrase file name put 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=xxxxxxxxxxxxxxxxxxxx none luks,discard,keyscript=/usr/local/sbin/azure_crypt_key.sh
```

3.If you are editing the *azure_crypt_key.sh* in Windows and copied it to Linux, do not forget to 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
```

4.Edit */etc/initramfs-tools/modules* by appending lines:

```
vfat
ntfs
nls_cp437
nls_utf8
nls_iso8859-1
```

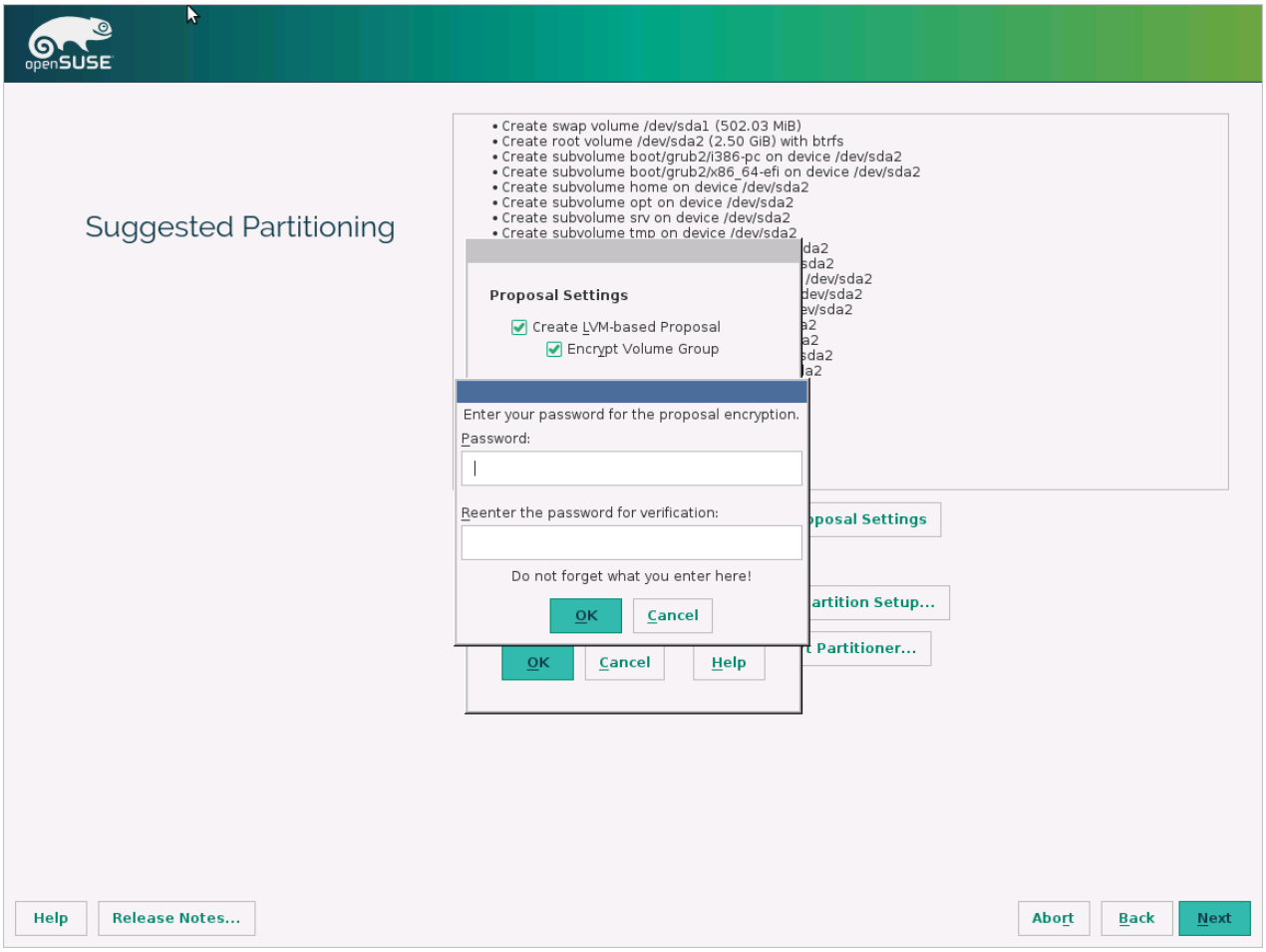
5.Run `update-initramfs -u -k all` to update the initramfs to make the `keyscript` take effect. 6.Now you can deprovision the VM.

```
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-4f02-9a13-289c6c99dbb8 none luks,discard,keyscript=/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_iso8859-1
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: 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.
[ 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
```

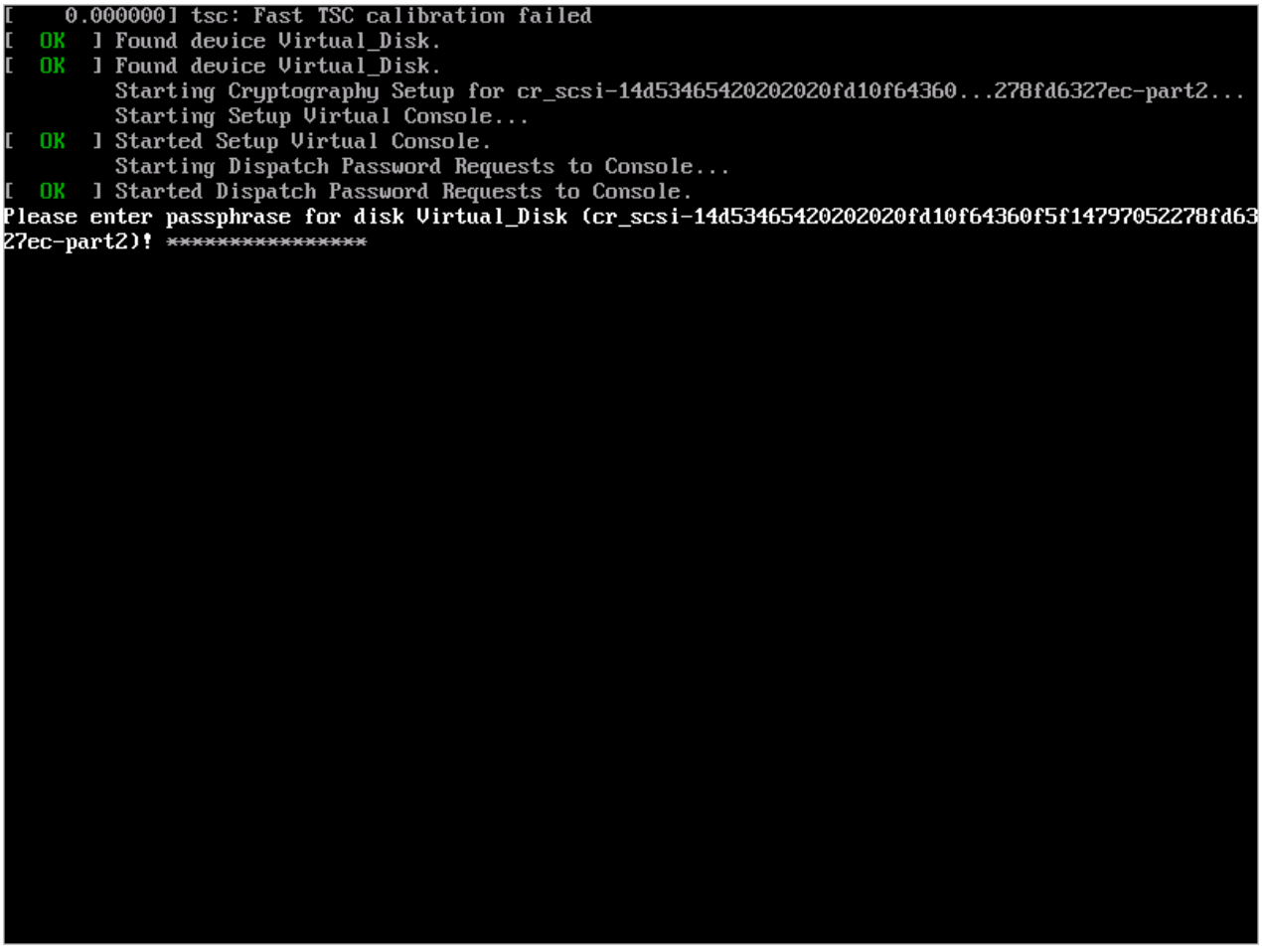
7.Continue to next step and upload your VHD into Azure.

openSUSE 13.2
Configure encryption during distro install

1.Select "Encrypt Volume Group" when partitioning disks. Provide a passphrase. This is the passphrase that you will upload into KeyVault.



2.Boot the VM using your passphrase.



3.Prepare VM for uploading into Azure using these instructions. Do not run the last step (deprovisioning the VM)

yet.

Configure encryption to work with Azure

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 \
#       $systemdutildir/system-generators/systemd-cryptsetup-generator \
#       $systemdutildir/systemd-cryptsetup \
#       $systemdsystemunitdir/systemd-ask-password-console.path \
#       $systemdsystemunitdir/systemd-ask-password-console.service \
#       $systemdsystemunitdir/cryptsetup.target \
#       $systemdsystemunitdir/sysinit.target.wants/cryptsetup.target \
#       systemd-ask-password systemd-tty-ask-password-agent
#       inst_script "$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 this 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
```

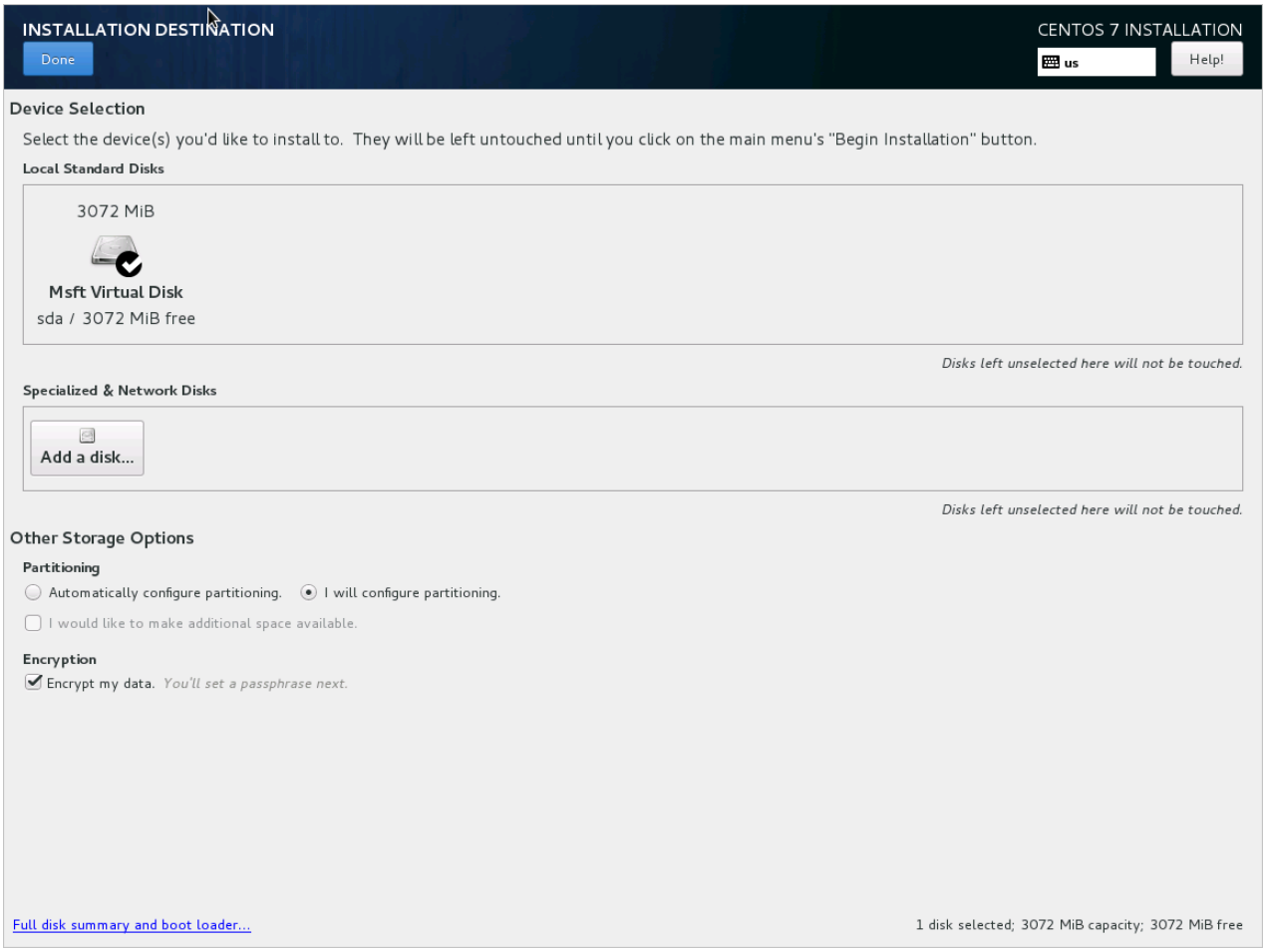
5.Run the "/usr/sbin/dracut -f -v" to update the initrd.

6.Now you can deprovision the VM and upload your VHD into Azure.

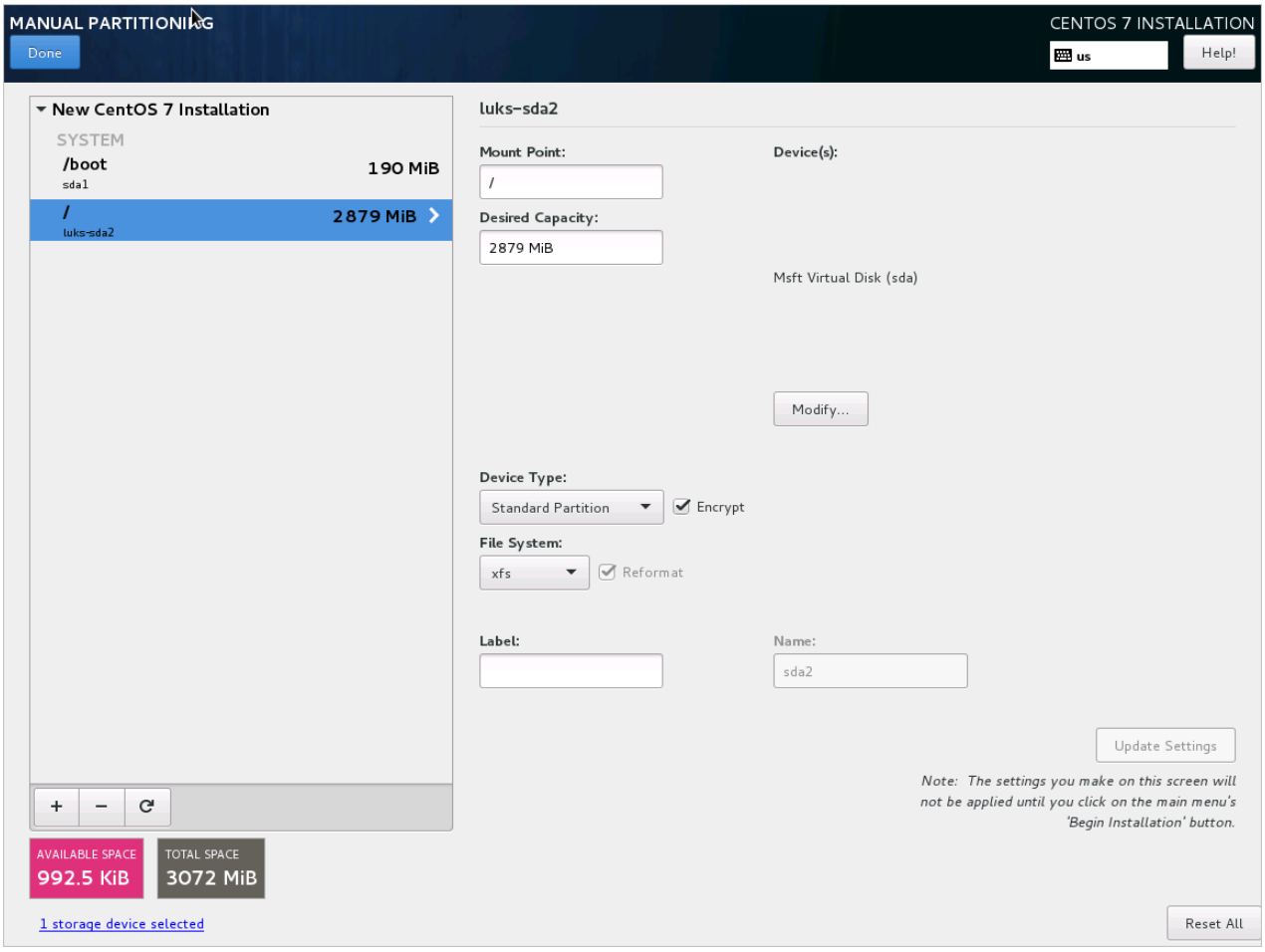
CentOS 7

Configure encryption during distro install

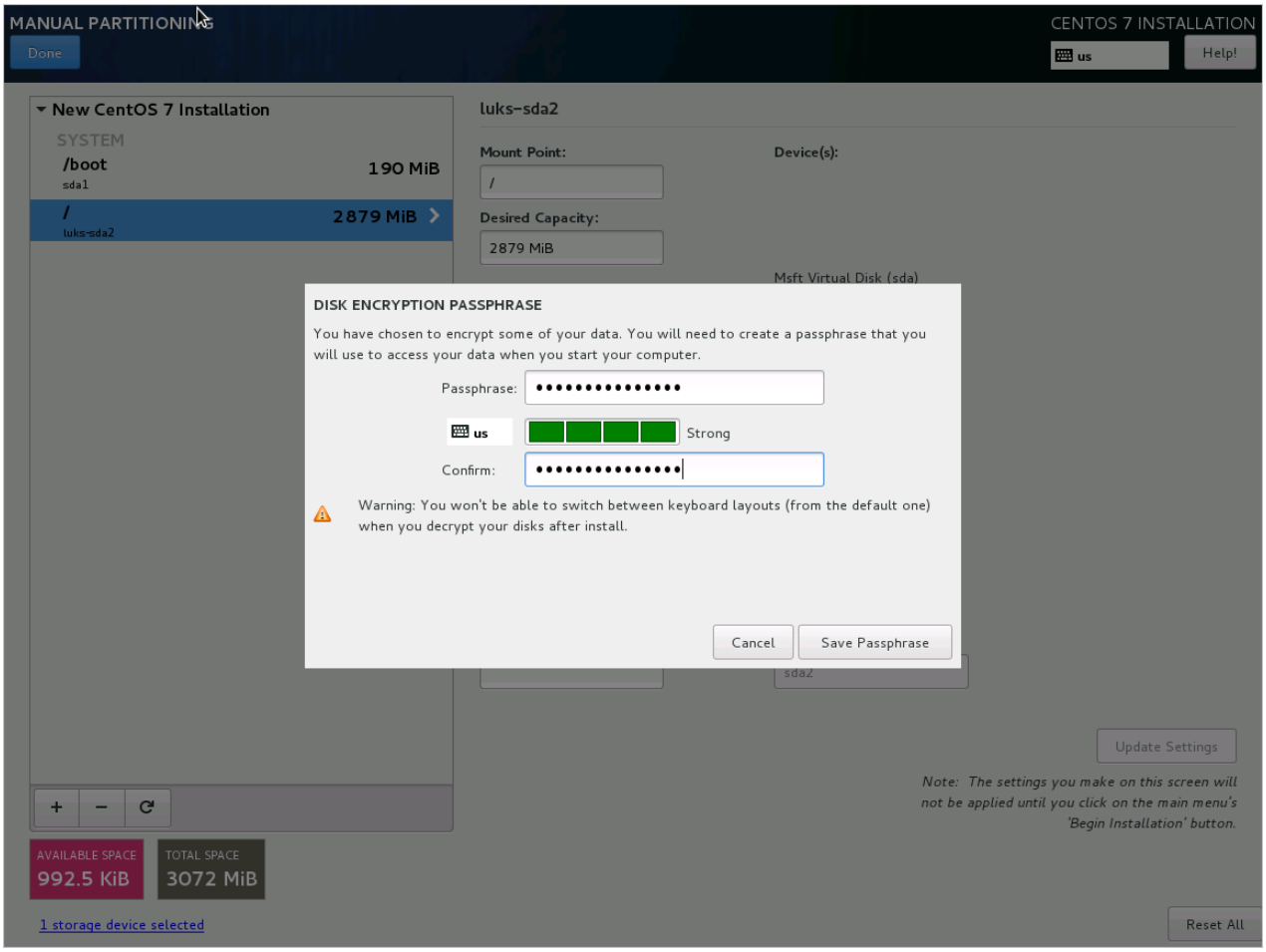
1.Select "Encrypt my data" when partitioning disks.



2.Make sure "Encrypt" is selected for root partition.



3.Provide a passphrase. This is the passphrase that you will upload into KeyVault.



4.When booting the VM, you will be asked for a passphrase. Use the passphrase you provided in step 3.

5.Prepare VM for uploading into Azure using these instructions. Do not run the last step (deprovisioning the VM)

yet.

6.Now you can deprovision the VM and upload your VHD into Azure.

Configure encryption to work with Azure

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 \
#        $systemdutildir/system-generators/systemd-cryptsetup-generator \
#        $systemdutildir/systemd-cryptsetup \
#        $systemdsystemunitdir/systemd-ask-password-console.path \
#        $systemdsystemunitdir/systemd-ask-password-console.service \
#        $systemdsystemunitdir/cryptsetup.target \
#        $systemdsystemunitdir/sysinit.target.wants/cryptsetup.target \
#        systemd-ask-password systemd-tty-ask-password-agent
#        inst_script "$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
```

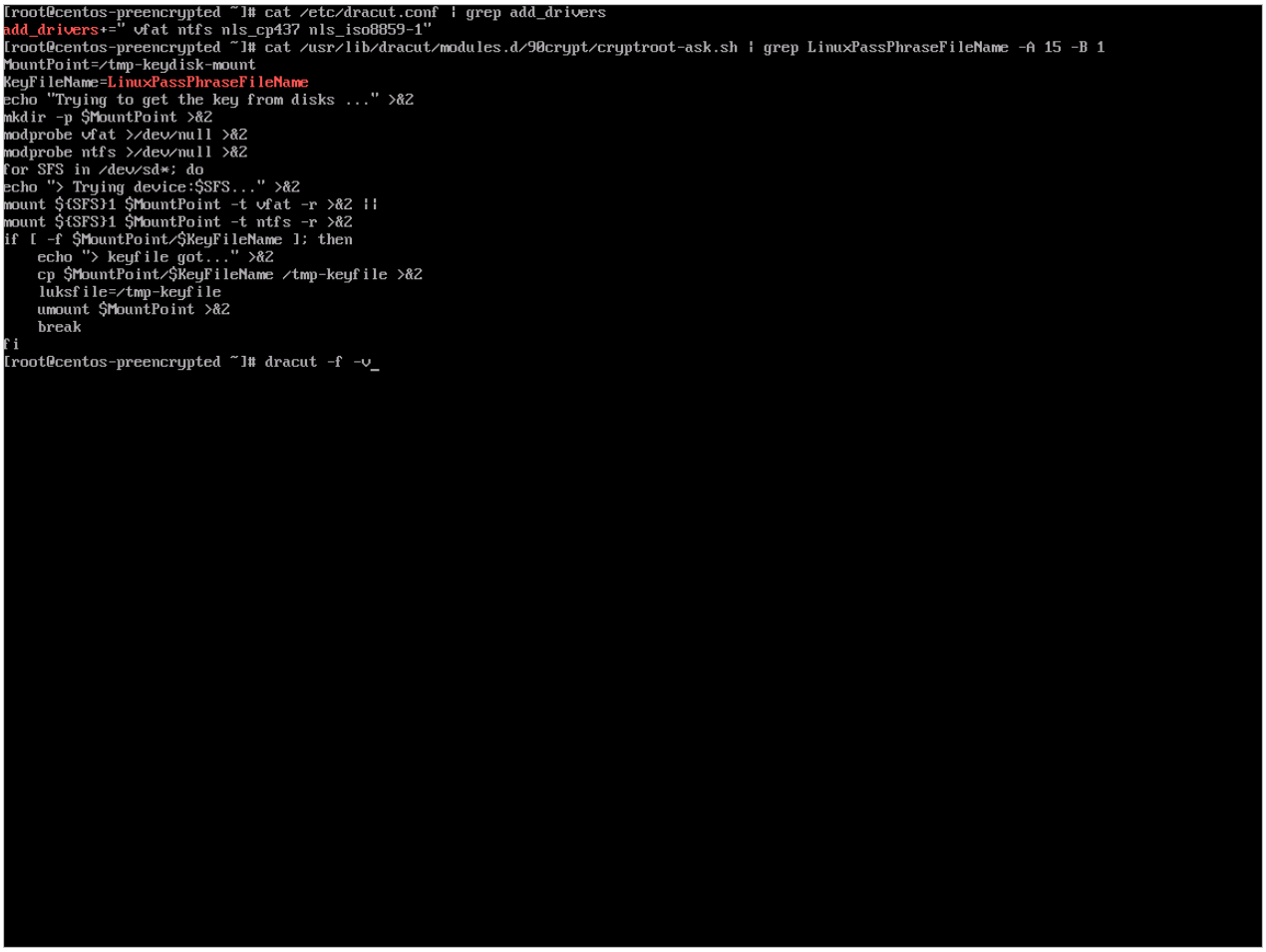
to

```
if [ 1 ]; then
```

4.Edit /usr/lib/dracut/modules.d/90crypt/cryptroot-ask.sh and append this 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.



Upload encrypted VHD to an Azure storage account

Once BitLocker encryption pr 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 disk encryption secret for the pre-encrypted VM to Key Vault

The disk encryption secret obtained previously needs to be uploaded as a secret in Key Vault. The Key Vault needs to have permissions enabled for your AAD client as well as disk encryption.

```
$AadClientId = "YourAADClientId"
$AadClientSecret = "YourAADClientSecret"

$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

Use Set-AzureKeyVaultSecret to provision the secret in key vault. In case of a Windows virtual machine, the bek file is encoded as a base64 string and then uploaded to key vault using the Set-AzureKeyVaultSecret cmdlet. For Linux, the passphrase is encoded as a base64 string and then uploaded to Key Vault. In addition, make sure that the following tags are set while creating the secret in key vault.

```
# This is the passphrase that was provided for encryption during distro install
$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
```

The `$secretUrl` shall be used in the next step for attaching the OS disk without using KEK.

Disk encryption secret encrypted with a KEK

The secret can optionally be encrypted with a Key Encryption Key before uploading to Key vault. 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 is then uploaded as a secret using the Set-AzureKeyVaultSecret cmdlet.

```
# This is the passphrase that was provided for encryption during distro install
$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 = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP"; "DiskEncryptionKeyFileName" =
"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
```

The `$KeyEncryptionKey` and `$secretUrl` shall be used in the next step for attaching the OS disk using KEK.

Specify secret URL when attaching OS Disk

Without using a KEK

While attaching the OS disk, `$secretUrl` needs to be passed. The URL was generated in the section "disk encryption secret not encrypted with a KEK".

```
Set-AzureRmVMOSDisk `
        -VM $VirtualMachine `
        -Name $OSDiskName `
        -SourceImageUri $VhdUri `
        -VhdUri $OSDiskUri `
        -Linux `
        -CreateOption FromImage `
        -DiskEncryptionKeyVaultId $KeyVault.ResourceId `
        -DiskEncryptionKeyUrl $SecretUrl
```

Using a KEK

While attaching the OS disk, `$KeyEncryptionKey` and `$secretUrl` need to be passed. The URL was generated in the section "disk encryption secret encrypted with a KEK".

```
Set-AzureRmVMOSDisk `
        -VM $VirtualMachine `
        -Name $OSDiskName `
        -SourceImageUri $CopiedTemplateBlobUri `
        -VhdUri $OSDiskUri `
        -Linux `
        -CreateOption FromImage `
        -DiskEncryptionKeyVaultId $KeyVault.ResourceId `
        -DiskEncryptionKeyUrl $SecretUrl `
        -KeyEncryptionKeyVaultId $KeyVault.ResourceId `
        -KeyEncryptionKeyURL $KeyEncryptionKey.Id
```

Download this Guide

You can download this guide from the TechNet Gallery.

For more information

Explore Azure Disk Encryption with Azure PowerShell

Explore Azure Disk Encryption with Azure PowerShell - Part 2

Encrypt an Azure Virtual Machine

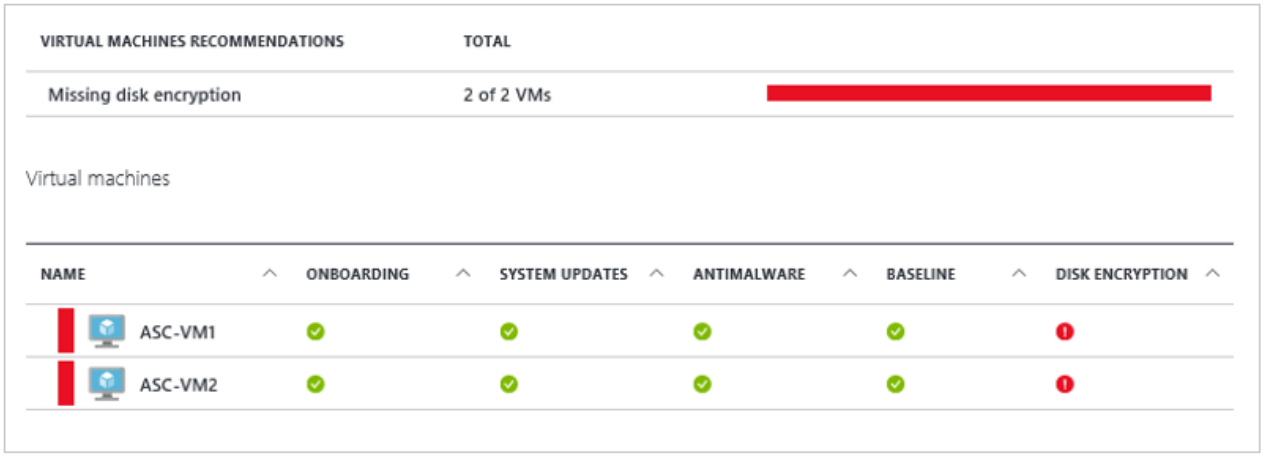
11/15/2016 • 10 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear • Yuri Diogenes • curtand • alexandair

---

Azure Security Center will alert you if you have virtual machines that are not encrypted. These alerts will show as High Severity and the recommendation is to encrypt these virtual machines.



> **NOTE**
> The information in this document applies to the preview release of Azure Security Center.

To encrypt Azure Virtual Machines that have been identified by Azure Security Center as needing encryption, we recommend the following steps:

- Install and configure Azure PowerShell. This will enable you to run the PowerShell commands required to set up the prerequisites required to encrypt Azure Virtual Machines.
- Obtain and run the Azure Disk Encryption Prerequisites Azure PowerShell script
- Encrypt your virtual machines

The goal of this document is to enable you to encrypt your virtual machines, even if you have little or no background in Azure PowerShell. This document assumes you are using Windows 10 as the client machine from which you will configure Azure Disk Encryption.

There are many approaches that can be used to setup the prerequisites and to configure encryption for Azure Virtual Machines. If you are already well-versed in Azure PowerShell or Azure CLI, then you may prefer to use alternate approaches.

> **NOTE**
> To learn more about alternate approaches to configuring encryption for Azure virtual machines, please see Azure Disk Encryption for Windows and Linux Azure Virtual Machines.

Install and configure Azure PowerShell

You need Azure PowerShell version 1.2.1 or above installed on your computer. The article How to install and configure Azure PowerShell contains all the steps you need to provision your computer to work with Azure PowerShell. The most straightforward approach is to use the Web PI installation approach mentioned in that article. Even if you already have Azure PowerShell installed, install again using the Web PI approach so that you have the latest version of Azure PowerShell.

Obtain and run the Azure disk encryption prerequisites configuration script

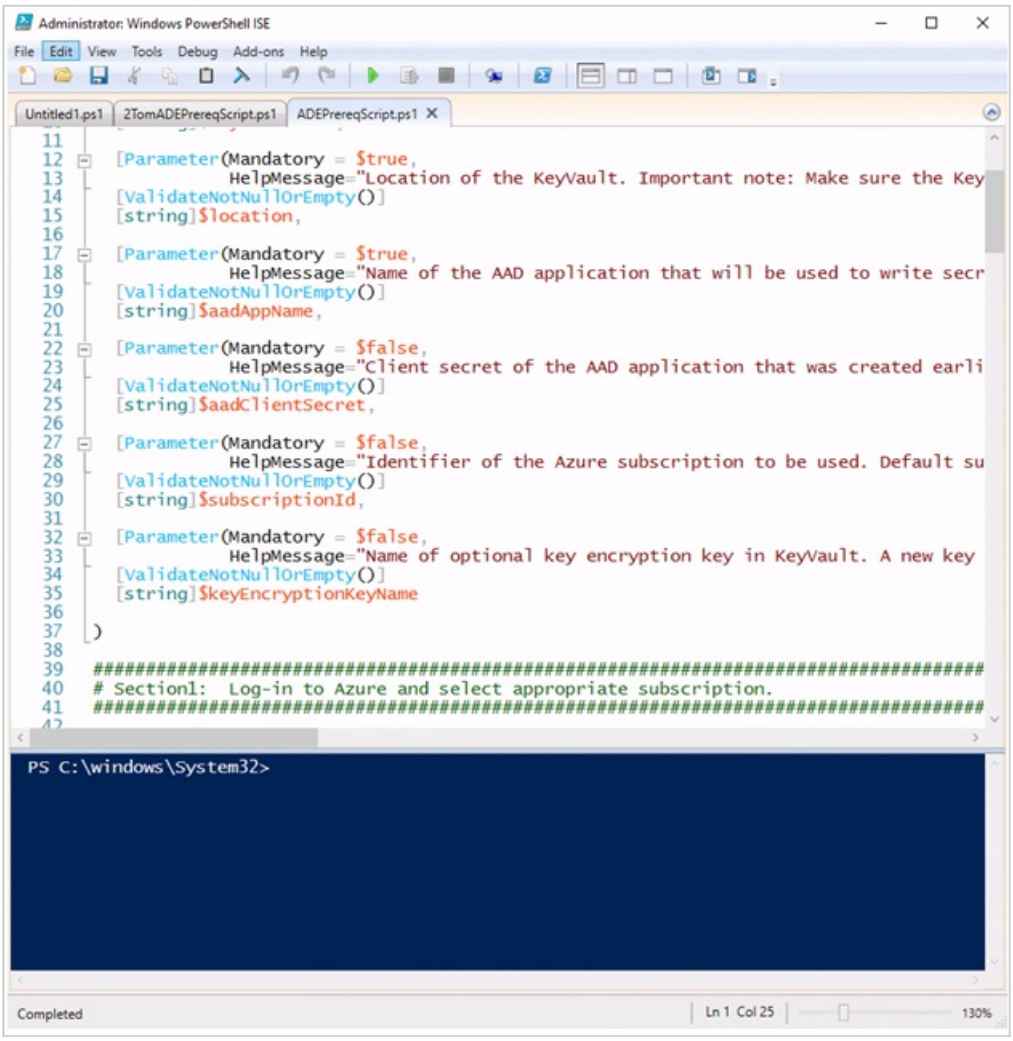
The Azure Disk Encryption Prerequisites Configuration Script will set up all the prerequisites required for encrypting your Azure Virtual Machines.

1. Go to the GitHub page that has the Azure Disk Encryption Prerequisite Setup Script.
2. On the GibHub page, click the **Raw** button.
3. Use **CTRL-A** to select all the text on the page and then use **CTRL-C** to copy all the text on the page to the clipboard.
4. Open **Notepad** and paste the copied text into Notepad.
5. Create a new folder on your C: drive named **AzureADEScript**.
6. Save the Notepad file – click **File**, then click **Save As**. In the File name textbox, enter **"ADEPrereqScript.ps1"** and click **Save**. (make sure you put the quotation marks around the name, otherwise it will save the file with a .txt file extension).

Now that the script content is saved, open the script in the PowerShell ISE:

1. In the Start Menu, click **Cortana**. Ask **Cortana** "PowerShell" by typing **PowerShell** in the Cortana search text box.
2. Right click **Windows PowerShell ISE** and click **Run as administrator**.
3. In the **Administrator: Windows PowerShell ISE** window, click **View** and then click **Show Script Pane**.
4. If you see the **Commands** pane on the right side of the window, click the **"x"** in the top right corner of the pane to close it. If the text is too small for you to see, use **CTRL+Add** ("Add" is the "+" sign). If the text is too large, use **CTRL+Subtract** (Subtract is the "-" sign).
5. Click **File** and then click **Open**. Navigate to the **C:\AzureADEScript** folder and the double-click on the **ADEPrereqScript**.
6. The **ADEPrereqScript** contents should now appear in the PowerShell ISE and is color-coded to help you see various components, such as commands, parameters and variables more easily.

You should now see something like the figure below.



The top pane is referred to as the “script pane” and the bottom pane is referred to as the “console”. We will use these terms later in this article.

Run the Azure disk encryption prerequisites PowerShell command

The Azure Disk Encryption Prerequisites script will ask you for the following information after you start the script:

- **Resource Group Name** - Name of the Resource Group that you want to put the Key Vault into. A new Resource Group with the name you enter will be created if there isn’t already one with that name created. If you already have a Resource Group that you want to use in this subscription, then enter the name of that Resource Group.
- **Key Vault Name** - Name of the Key Vault in which encryption keys are to be placed. A new Key Vault with this name will be created if you don’t already have a Key Vault with this name. If you already have a Key Vault that you want to use, enter the name of the existing Key Vault.
- **Location** - Location of the Key Vault. Make sure the Key Vault and VMs to be encrypted are in the same location. If you don’t know the location, there are steps later in this article that will show you how to find out.
- **Azure Active Directory Application Name** - Name of the Azure Active Directory application that will be used to write secrets to the Key Vault. A new application with this name will be created if one doesn't exist. If you already have an Azure Active Directory application that you want to use, enter the name of that Azure Active Directory application.

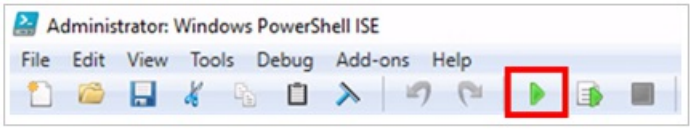
> **NOTE**
>
> If you’re curious as to why you need to create an Azure Active Directory application, please see *Register an application with Azure Active Directory* section in the article Getting Started with Azure Key Vault.

Perform the following steps to encrypt an Azure Virtual Machine:

1. If you closed the PowerShell ISE, open an elevated instance of the PowerShell ISE. Follow the instructions earlier in this article if the PowerShell ISE is not already open. If you closed the script, then open the **ADEPrereqScript.ps1** clicking **File**, then **Open** and selecting the script from the **c:\AzureADEScript** folder. If you have followed this article from the start, then just move on to the next step.
2. In the console of the PowerShell ISE (the bottom pane of the PowerShell ISE), change the focus to the local of the script by typing **cd c:\AzureADEScript** and press **ENTER**.
3. Set the execution policy on your machine so that you can run the script. Type **Set-ExecutionPolicy Unrestricted** in the console and then press ENTER. If you see a dialog box telling about the effects of the change to execution policy, click either **Yes to all** or **Yes** (if you see **Yes to all**, select that option – if you do not see **Yes to all**, then click **Yes**).
4. Log into your Azure account. In the console, type **Login-AzureRmAccount** and press **ENTER**. A dialog box will appear in which you enter your credentials (make sure you have rights to change the virtual machines – if you do not have rights, you will not be able to encrypt them. If you are not sure, ask your subscription owner or administrator). You should see information about your **Environment**, **Account**, **TenantId**, **SubscriptionId** and **CurrentStorageAccount**. Copy the **SubscriptionId** to Notepad. You will need to use this in step #6.
5. Find what subscription your virtual machine belongs to and its location. Go to https://portal.azure.com and log in. On the left side of the page, click **Virtual Machines**. You will see a list of your virtual machines and the subscriptions they belong to.

NAME	STATUS	RESOURCE GROUP	LOCATION	SUBSCRIPTION
ASC-VM1	Running	ASC-ResourceGroup	Central US	Microsoft Azure Internal Consumption
ASC-VM2	Running	ASC-ResourceGroup	Central US	Microsoft Azure Internal Consumption
TomVM7	Running	TomRG7	Central US	Microsoft Azure Internal Consumption

6. Return to the PowerShell ISE. Set the subscription context in which the script will be run. In the console, type **Select-AzureRmSubscription –SubscriptionId** (replace **< your_subscription_Id >** with your actual Subscription ID) and press **ENTER**. You will see information about the Environment, **Account**, **TenantId**, **SubscriptionId** and **CurrentStorageAccount**.
7. You are now ready to run the script. Click the **Run Script** button or press **F5** on the keyboard.

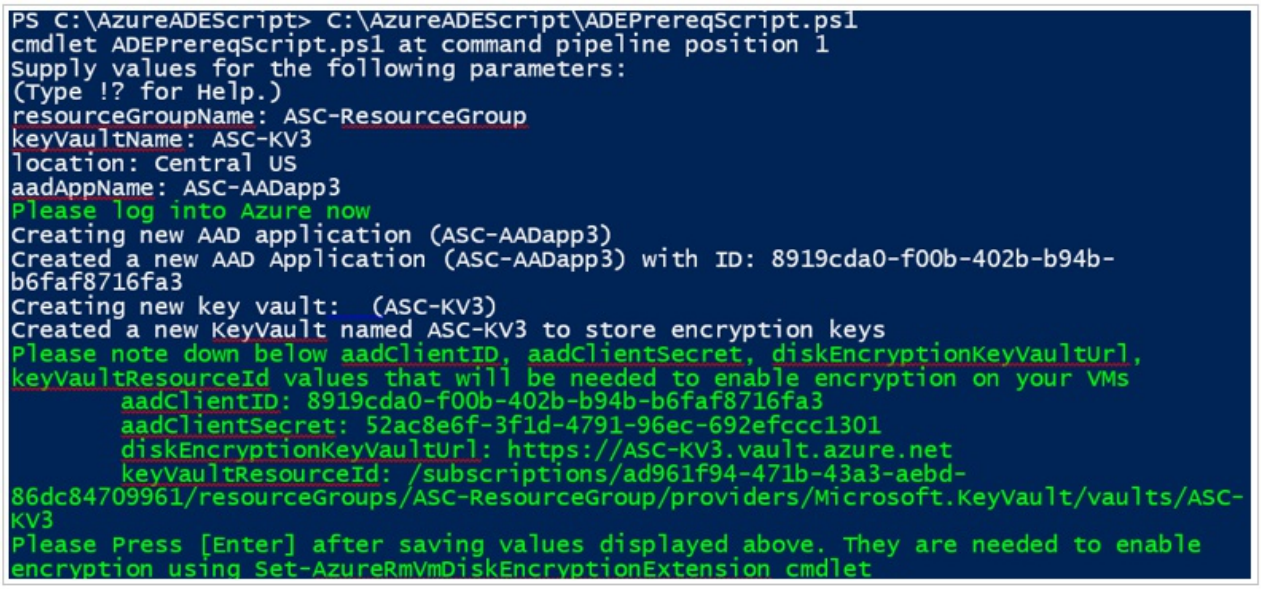


8. The script asks for **resourceGroupName:** - enter the name of *Resource Group* you want to use, then press **ENTER**. If you don't have one, enter a name you want to use for a new one. If you already have a *Resource Group* that you want to use (such as the one that your virtual machine is in), enter the name of the existing Resource Group.
9. The script asks for **keyVaultName:** - enter the name of the *Key Vault* you want to use, then press ENTER. If you don't have one, enter a name you want to use for a new one. If you already have a Key Vault that you want to use, enter the name of the existing *Key Vault*.
10. The script asks for **location:** - enter the name of the location in which the VM you want to encrypt is located, then press **ENTER**. If you don't remember the location, go back to step #5.
11. The script asks for **aadAppName:** - enter the name of the *Azure Active Directory* application you want to use, then press **ENTER**. If you don't have one, enter a name you want to use for a new one. If you already have an *Azure Active Directory application* that you want to use, enter the name of the existing *Azure Active Directory application*.
12. A log in dialog box will appear. Provide your credentials (yes, you have logged in once, but now you need to do

it again).

13. The script runs and when complete it will ask you to copy the values of the **aadClientID**, **aadClientSecret**, **diskEncryptionKeyVaultUrl**, and **keyVaultResourceId**. Copy each of these values to the clipboard and paste them into Notepad.
14. Return to the PowerShell ISE and place the cursor at the end of the last line, and press **ENTER**.

The output of the script should look something like the screen below:



Encrypt the Azure virtual machine

You are now ready to encrypt your virtual machine. If your virtual machine is located in the same Resource Group as your Key Vault, you can move on to the encryption steps section. However, if your virtual machine is not in the same Resource Group as your Key Vault, you will need to enter the following in the console in the PowerShell ISE:

$resourceGroupName = <'Virtual_Machine_RG'>

Replace **< Virtual_Machine_RG >** with the name of the Resource Group in which your virtual machines are contained, surrounded by a single quote. Then press **ENTER**. To confirm that the correct Resource Group name was entered, type the following in the PowerShell ISE console:

$resourceGroupName

Press **ENTER**. You should see the name of Resource Group that your virtual machines are located in. For example:

Encryption steps

First, you need to tell PowerShell the name of the virtual machine you want to encrypt. In the console, type:

$vmName = <'your_vm_name'>

Replace **<'your_vm_name'>** with the name of your VM (make sure the name is surrounded by a single quote) and then press **ENTER**.

To confirm that the correct VM name was entered, type:

$vmName

Press **ENTER**. You should see the name of the virtual machine you want to encrypt. For example:

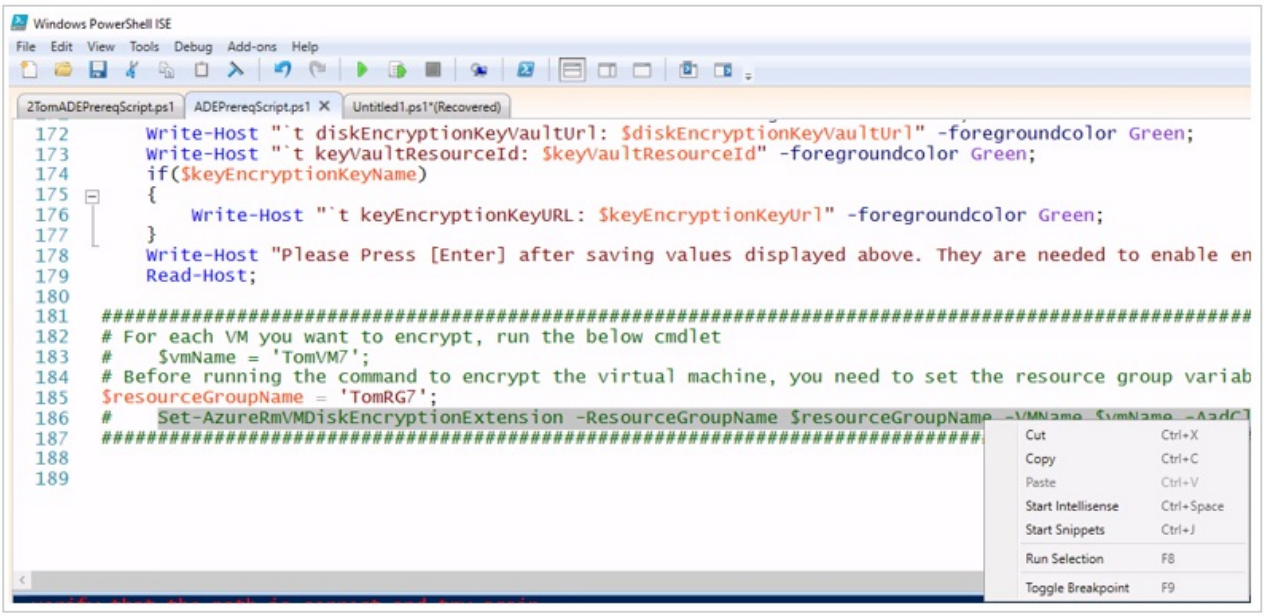
```
PS C:\AzureADEScript> $vmName = 'ASC-VM3'

PS C:\AzureADEScript> $vmName
ASC-VM3
```

There are two ways you can run the encryption command to encrypt the virtual machine. The first method is to type the following command in the PowerShell ISE console:

```
Set-AzureRmVMDiskEncryptionExtension -ResourceGroupName $resourceGroupName -VMName $vmName -AadClientID
$aadClientID -AadClientSecret $aadClientSecret -DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $keyVaultResourceId
```

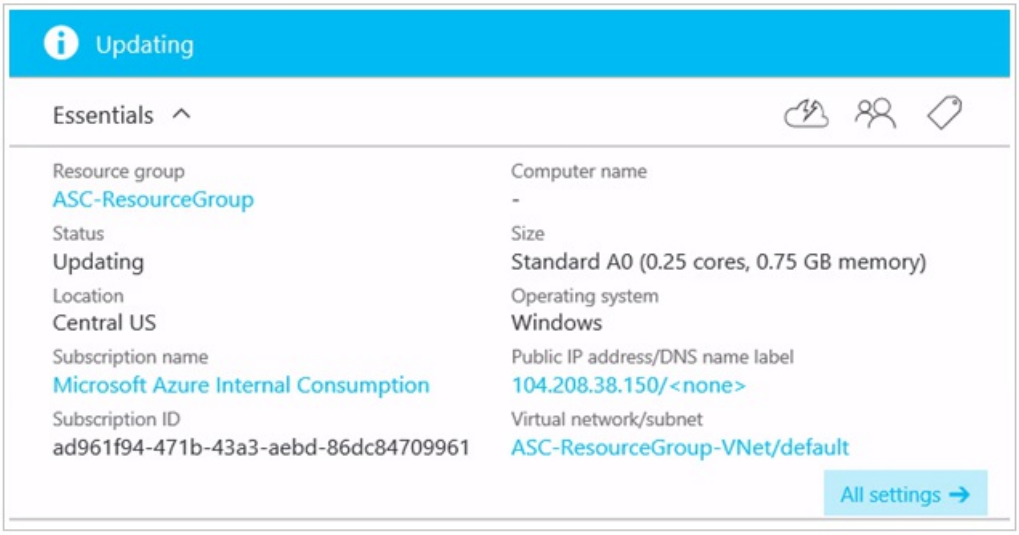
After typing this command press **ENTER**.

The second method is to click in the script pane (the top pane of the PowerShell ISE) and scroll down to the bottom of the script. Highlight the command listed above, and then right click it and then click **Run Selection** or press **F8** on the keyboard.



Regardless of the method you use, a dialog box will appear informing you that it will take 10-15 minutes for the operation to complete. Click **Yes**.

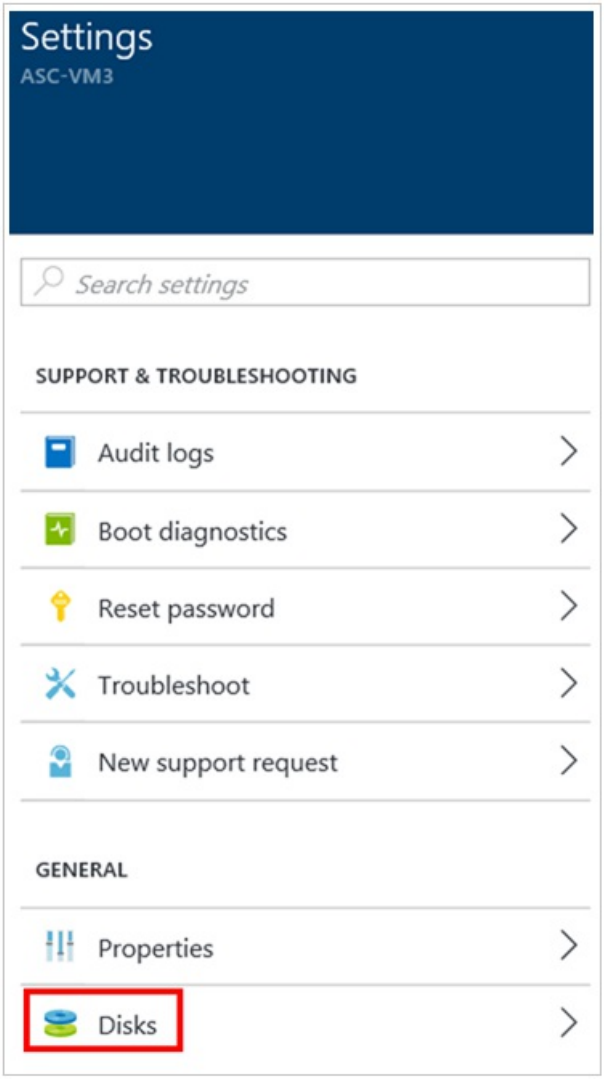
While the encryption process is taking place, you can return to the Azure Portal and see the status of the virtual machine. On the left side of the page, click **Virtual Machines**, then in the **Virtual Machines** blade, click the name of the virtual machine you're encrypting. In the blade that appears, you'll notice that the **Status** says that it's **Updating**. This demonstrates that encryption is in process.



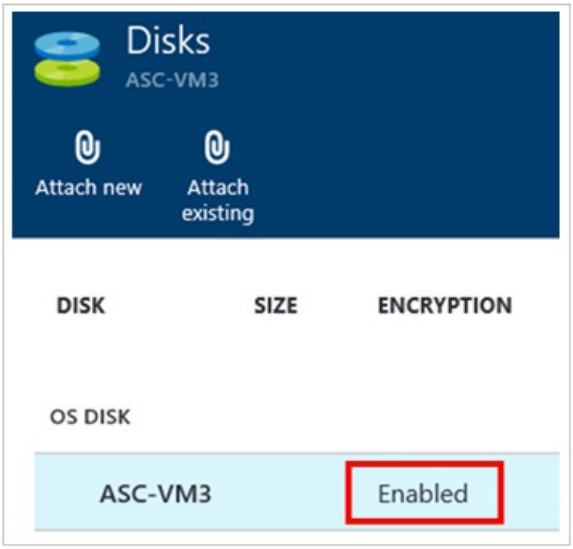
Return to the PowerShell ISE. When the script completes, you'll see what appears in the figure below.

```
RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
```

To demonstrate that the virtual machine is now encrypted, return to the Azure Portal and click **Virtual Machines** on the left side of the page. Click the name of the virtual machine you encrypted. In the **Settings** blade, click **Disks**.



On the **Disks** blade, you will see that **Encryption** is **Enabled**.



Next steps

In this document, you learned how to encrypt an Azure Virtual Machine. To learn more about Azure Security Center, see the following:

- Security health monitoring in Azure Security Center – Learn how to monitor the health of your Azure resources
- Managing and responding to security alerts in Azure Security Center - Learn how to manage and respond to security alerts
- Azure Security Center FAQ – Find frequently asked questions about using the service
- Azure Security Blog – Find blog posts about Azure security and compliance

Azure Security Management and Monitoring Overview

11/15/2016 • 6 min to read • Edit on GitHub

Contributors

TerryLanfear • Ralph Squillace • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes

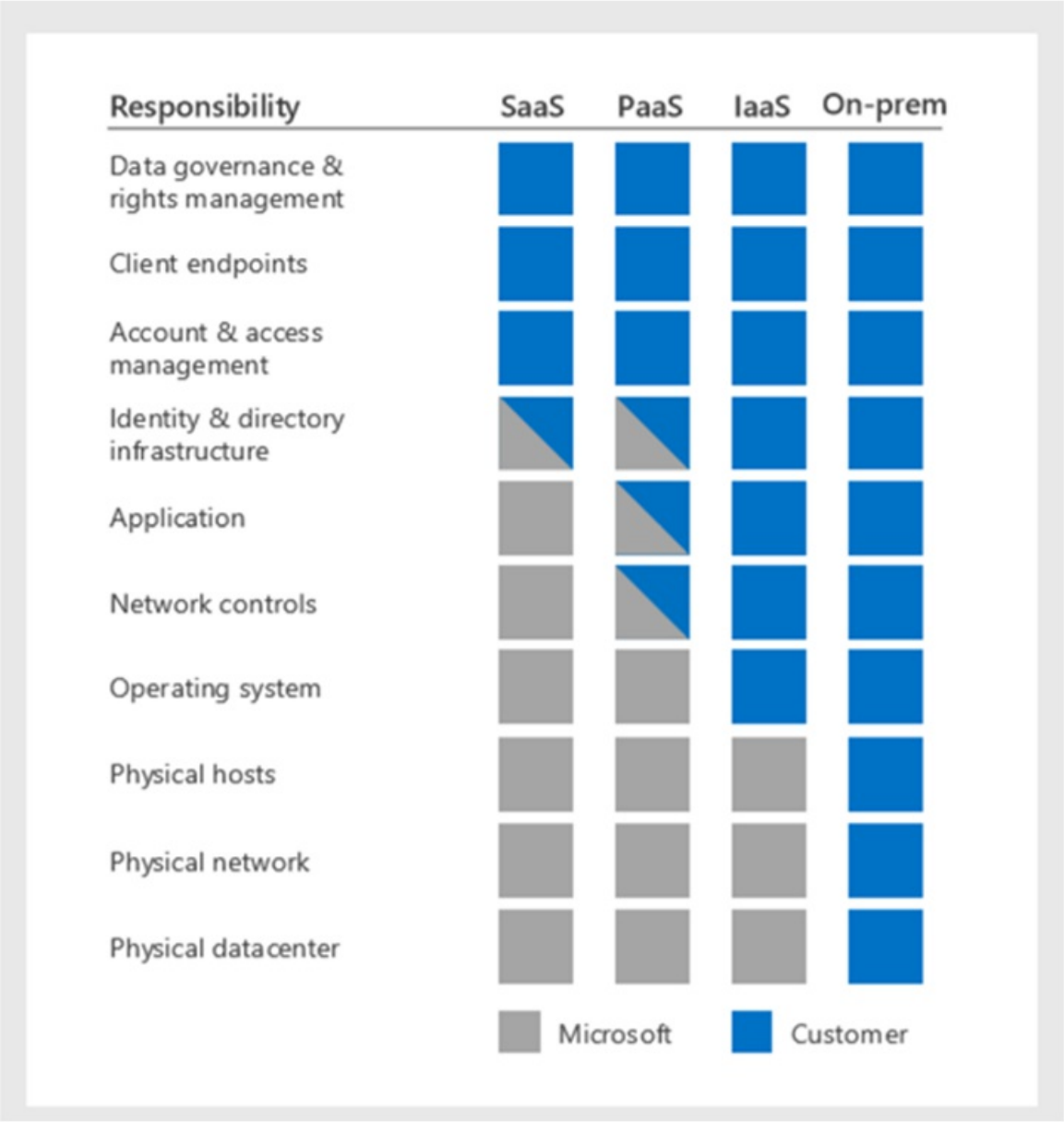
---

Azure provides security mechanisms to aid in the management and monitoring of Azure cloud services and virtual machines. This article provides an overview of these core security features and services. Links are provided to articles that will give details of each so you can learn more.

The security of your Microsoft cloud services is a partnership and shared responsibility between you and Microsoft. Shared responsibility means Microsoft is responsible for the Microsoft Azure and physical security of its data centers (through the use of security protections such as locked badge entry doors, fences, and guards). In addition, Azure provides strong levels of cloud security at the software layer that meets the security, privacy, and compliance needs of its demanding 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 provides you with security controls and capabilities to help you protect your data and applications. Your responsibility for security is based on the type of cloud service.

The following chart summarizes the balance of responsibility for both Microsoft and the customer.



For a deeper dive into security management, see Security management in Azure.

Here are the core features to be covered in this article:

- Role-Based Access Control
- Antimalware
- Multi-Factor Authentication
- ExpressRoute
- Virtual network gateways
- Privileged identity management
- Identity protection
- Security Center

Role-Based Access Control

Role-Based Access Control (RBAC) provides fine-grained access management for Azure resources. 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 to help protect your virtual machines from malicious files, adware, and other threats.

Microsoft Antimalware 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. DeepSecurity is an Antivirus solution and SecureCloud is an encryption solution. DeepSecurity will be deployed inside of VMs using an extension model. Using the portal UI and PowerShell, you can choose to use DeepSecurity inside of new VMs that are being spun up, or existing VMs that are already deployed.

Symantec End Point Protection (SEP) is also supported on Azure. Through portal integration, customers have the ability to specify that they intend to use SEP within a VM. SEP can be installed on a brand new VM via the Azure Portal or can be installed on an existing VM using 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 – McAfee Endpoint Protection

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 3rd party OATH tokens.

Learn more:

- Multi-factor authentication
- What is Azure Multi-Factor Authentication?
- How Azure Multi-Factor Authentication works

ExpressRoute

Microsoft Azure ExpressRoute 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. This allows ExpressRoute connections to 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 (VNet-to-VNet). 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 have to 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 could impact your 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, which 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 such as 8 hours.

Learn more:

- Azure AD Privileged Identity Management
- Get started with Azure AD Privileged Identity Management

Identity Protection

Azure Active Directory (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, and 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, and 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, 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.

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 and 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, along with the information you need to quickly investigate and recommendations on how to remediate an attack.

Learn more:

- Introduction to Azure Security Center

Security management in Azure

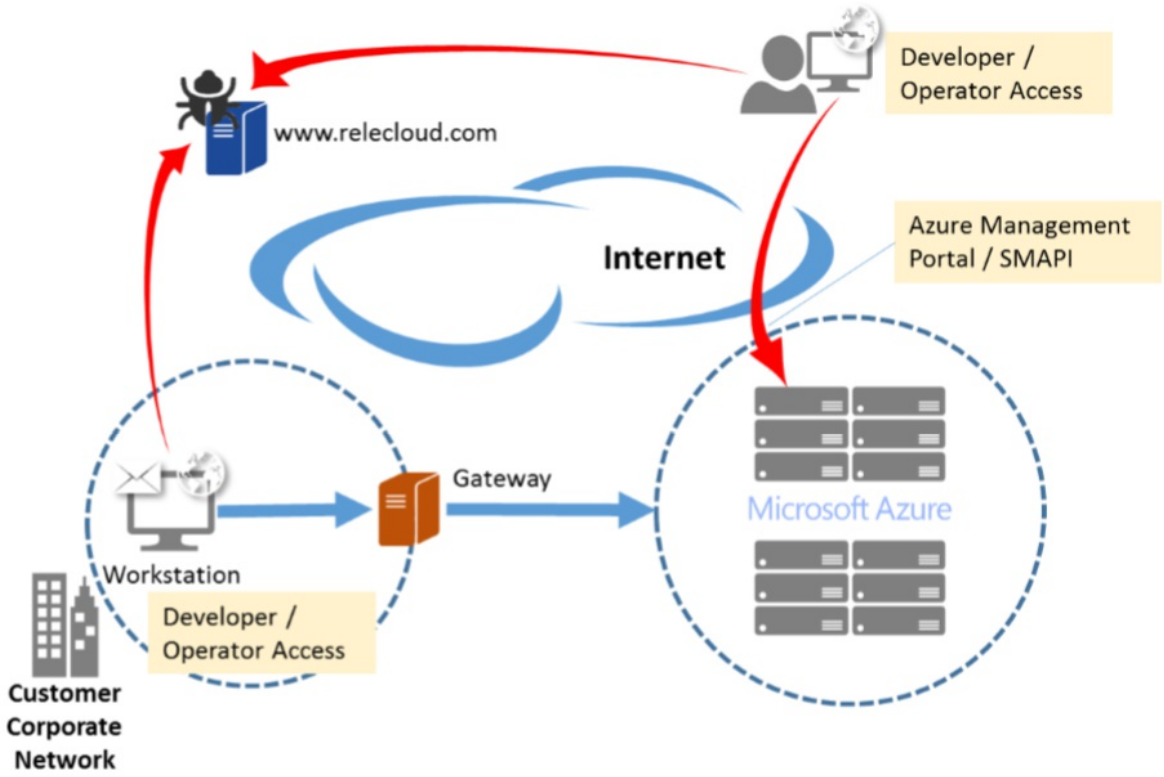
11/15/2016 • 20 min to read • Edit on GitHub

Contributors

TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes

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 will lead 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.

In combination 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—as well as 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 of 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 SMAPI, 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 SMAPI 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 management 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 management 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 very 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 SMAPI (via customer-developed tools such as Windows PowerShell cmdlets) and the Azure management 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 management 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 will provide 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		

CONFIGURATION	BENEFITS	CONS
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.

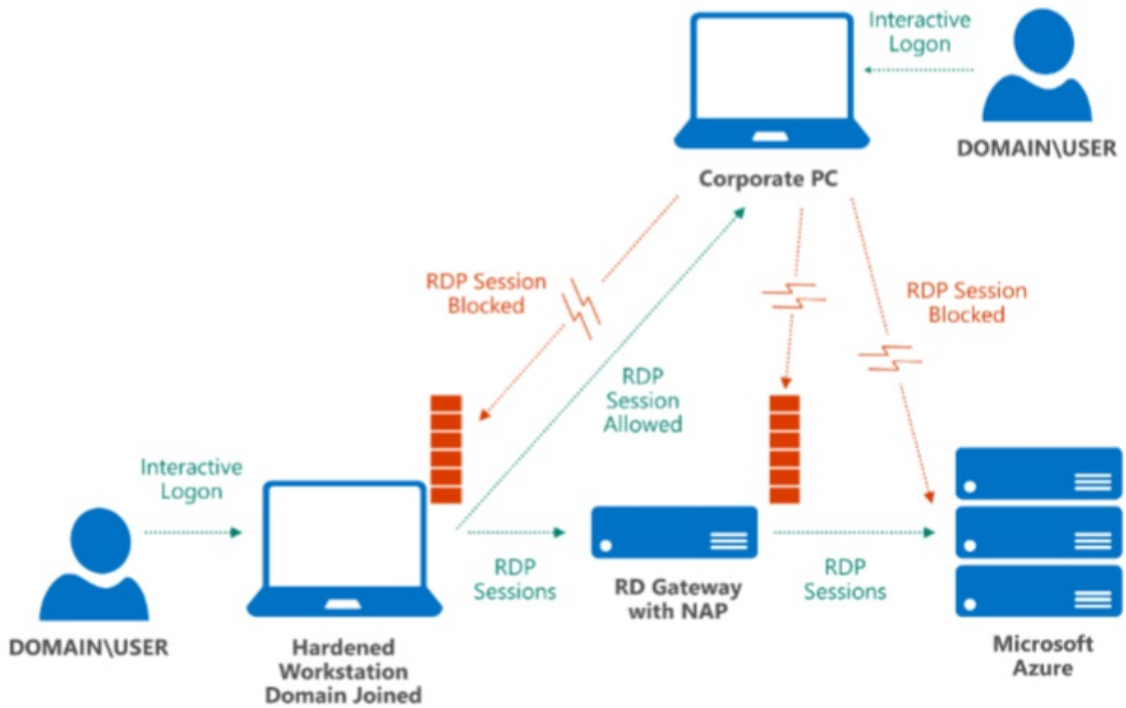
You can further segregate administrative functions through dedicated system images for each hardened workstation that have only the tools and permissions needed for managing select Azure and cloud applications, with specific local AD DS GPOs for the necessary tasks.

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.

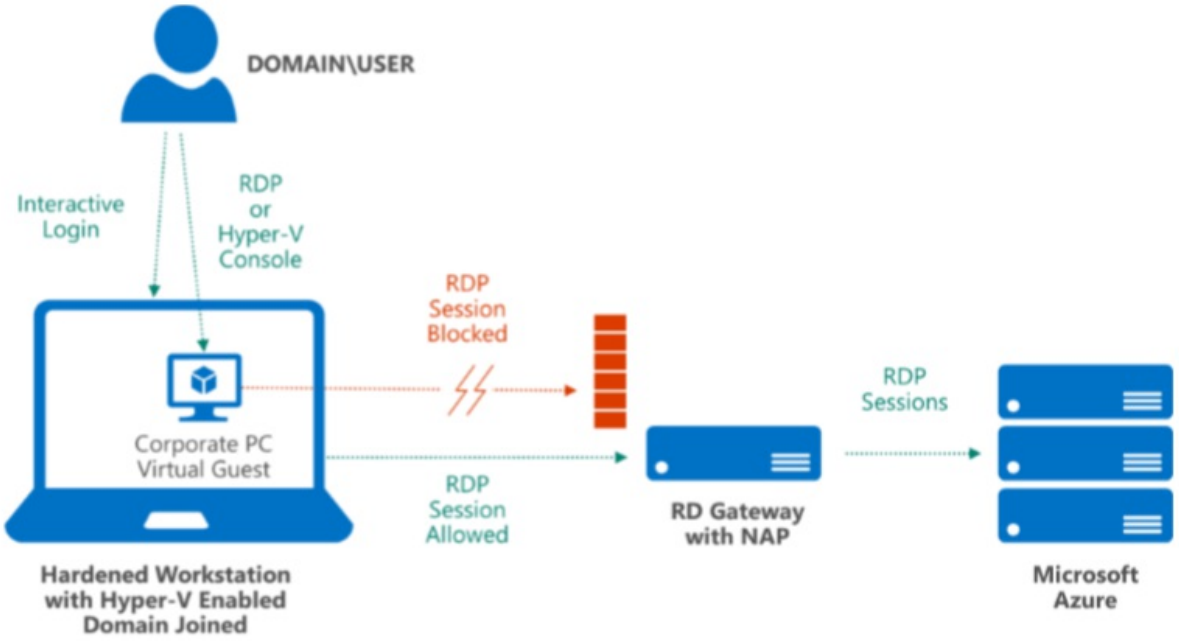
In the stand-alone hardened workstation scenario (shown below), the local instance of Windows Firewall (or a non-Microsoft client firewall) is configured to block inbound connections, such as RDP. The administrator can log on to the hardened workstation and start an RDP session that connects to Azure 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.



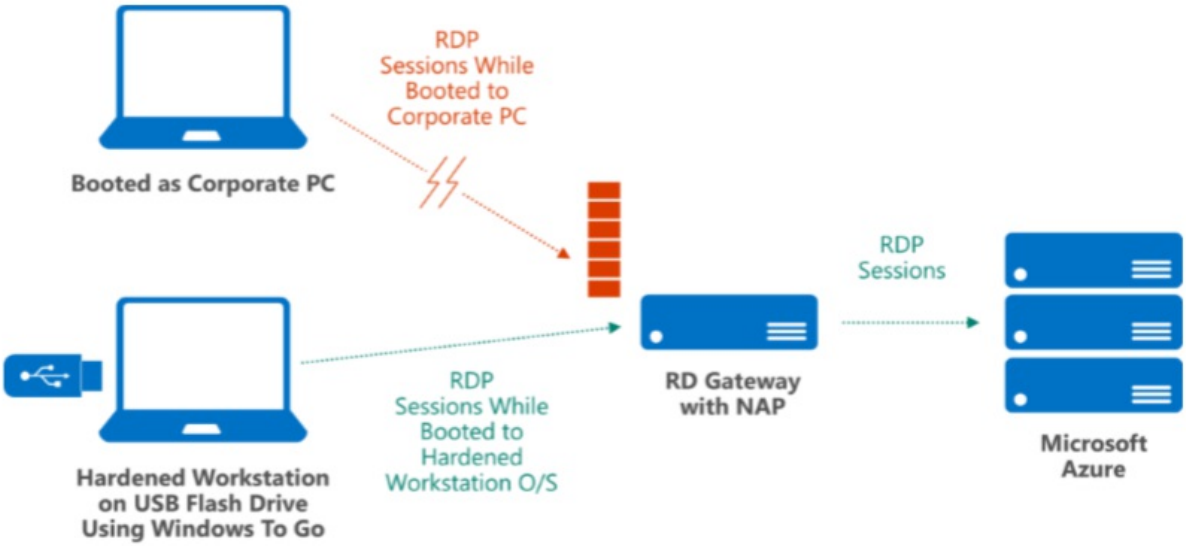
To avoid several security risks that can arise from using one workstation for systems management and other daily work tasks, you can deploy a Windows Hyper-V virtual machine to the hardened workstation. This virtual machine can be used as the corporate PC. The corporate PC environment can remain isolated 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 will be 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, will make 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 (changeon-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 will help 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), as well as 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 of 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

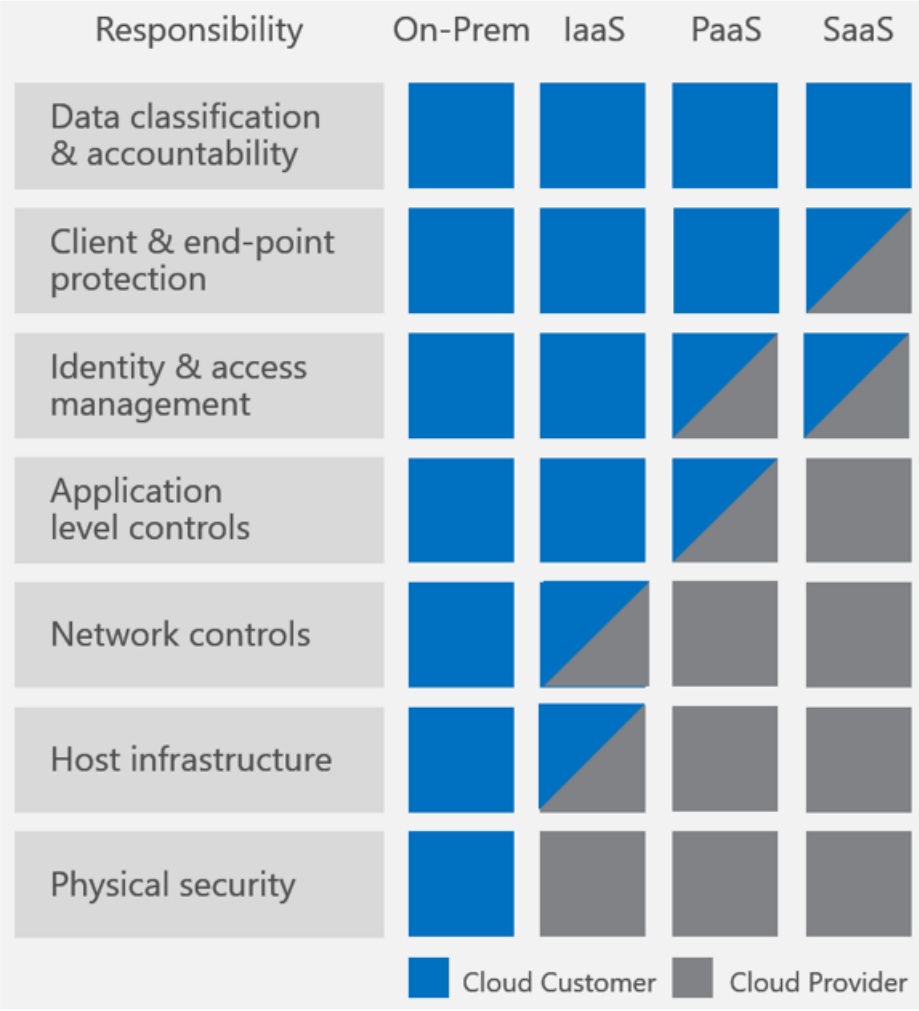
Best practices for software updates on Microsoft Azure IaaS

11/15/2016 • 6 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • 4c74356b41 • Carolyn Gronlund

Before diving into any kind of discussion on best practices for an Azure IaaS environment, it is important to understand what the scenarios are that will have you managing software updates and the responsibilities. The diagram below should help you understand these boundaries:



The left-most column shows seven responsibilities (defined in the sections that follow) that organizations should consider, all of which contribute to the security and privacy of a computing environment.

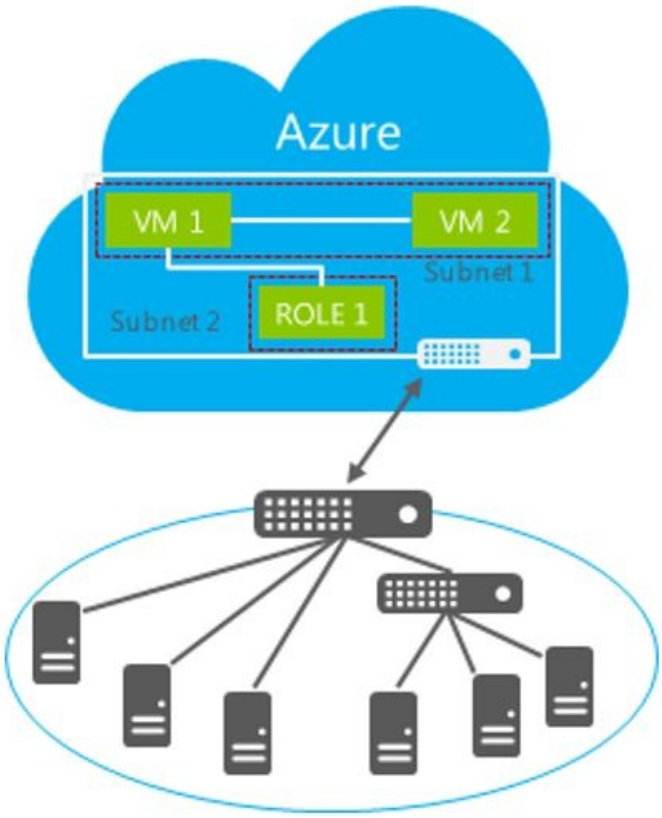
Data classification & accountability and Client & end-point protection are the responsibilities that are solely in the domain of customers, and Physical, Host, and Network responsibilities are in the domain of cloud service providers in the PaaS and SaaS models.

The remaining responsibilities are shared between customers and cloud service providers. Some responsibilities require the CSP and customer to manage and administer the responsibility together, including auditing of their domains. For example, consider Identity & access management when using Azure Active Directory Services; the configuration of services such as multi-factor authentication is up to the customer, but ensuring effective

functionality is the responsibility of Microsoft Azure.

> **NOTE**
> For more information about shared responsibilities in the cloud, read Shared Responsibilities for Cloud Computing

These same principles apply in a hybrid scenario where your company is using Azure IaaS VMs that communicate with on-premises resources as shown in the diagram below.



Initial assessment

Even if your company is already using an update management system and you already have software update policies in place, it is important to frequently revisit previous policy assessments and update them based on your current requirements. This means that you need to be familiar with the current state of the resources in your company. To reach this state, you have to know:

- The physical and virtual computers in your enterprise.
- Operating systems and versions running on each of these physical and virtual computers.
- Software updates currently installed on each computer (service pack versions, software updates, and other modifications).
- The function each computer performs in your enterprise.
- The applications and programs running on each computer.
- Ownership and contact information for each computer.
- The assets present in your environment and their relative value to determine which areas need the most attention and protection.
- Known security problems and the processes your enterprise has in place for identifying new security issues or changes in security level.
- Countermeasures that have been deployed to secure your environment.

You should update this information regularly, and it should be readily available to those involved in your software

update management process.

Establish a Baseline

An important part of the software update management process is creating initial standard installations of operating system versions, applications, and hardware for computers in your enterprise; these are called baselines. A baseline is the configuration of a product or system established at a specific point in time. An application or operating system baseline, for example, provides the ability to rebuild a computer or service to a specific state.

Baselines 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.

After performing the initial audit of your enterprise, you should use the information that is obtained from the audit to define an operational baseline for the IT components within your production environment. A number of baselines might be required, depending on the different types of hardware and software deployed into production.

For example, some servers require a software update to prevent them from hanging when they enter the shutdown process when running Windows Server 2012. A baseline for these servers should include this software update.

In large organizations, it is often helpful to divide the computers in your enterprise into asset categories and keep each category at a standard baseline by using the same versions of software and software updates. You can then use these asset categories in prioritizing a software update distribution.

Subscribe to the appropriate software update notification services

After you perform an initial audit of the software in use in your enterprise, you should determine the best method for receiving notifications of new software updates for each software product and version. Depending on the software product, the best notification method might be e-mail notifications, Web sites, or computer publications.

For example, the Microsoft Security Response Center (MSRC) responds to all security-related concerns about Microsoft products and provides the Microsoft Security Bulletin Service, a free e-mail notification of newly identified vulnerabilities and software updates that are released to address these vulnerabilities. You can subscribe to this service at http://www.microsoft.com/technet/security/bulletin/notify.mspx.

Software update considerations

After you perform an initial audit of the software in use in your enterprise, you should determine the requirements to setup your software update management system, which depends on the software update management system that you are using. For WSUS read Best Practices with Windows Server Update Services, for System Center read Planning for Software Updates in Configuration Manager.

However, there are some general considerations and best practices that you can apply regardless of the solution that you are using as shown in the sections that follows.

Setting up the environment

Consider the following practices when planning to setup the software update management environment:

- **Create production software update collections based on stable criteria**: In general, using stable criteria to create collections for your software update inventory and distribution will help to simplify all stages of the software update management process. The stable criteria can include the installed client operating system version and service pack level, system role, or target organization.
- **Create pre-production collections that include reference computers**: The pre-production collection should include representative configurations of the operating system versions, line of business software, and other software running in your enterprise.

You should also consider where the software update server will be located, if it will be in the Azure IaaS infrastructure in the cloud or if it will be on-premises. This is an important decision because you need to evaluate the amount of traffic between on-premises resources and Azure infrastructure. Read Connect an on-premises network to a Microsoft Azure virtual network for more information on how to connect your on-premises infrastructure to Azure.

The design options that will determine where the update server will be located will also vary according to your current infrastructure and the software update system that you are currently using. For WSUS read Deploy Windows Server Update Services in Your Organization and for System Center Configuration Manager read Planning for Sites and Hierarchies in Configuration Manager.

Backup

Regular backups are important not only for the software update management platform itself but also for the servers that will be updated. Organizations that have a change management process in place will require IT to justify the reasons for why the server needs to be updated, the estimated downtime and possible impact. To ensure that you have a rollback configuration in place in case an update fails, make sure to back up the system regularly.

Some backup options for Azure IaaS include:

- Azure IaaS workload protection using Data Protection Manager
- Back up Azure virtual machines

Monitoring

You should run regular reports to monitor the number of missing or installed updates, or updates with incomplete status, for each software update that is authorized. Similarly, reporting for software updates that are not yet authorized can facilitate easier deployment decisions.

You should also consider the following tasks:

- Conduct an audit of applicable and installed security updates for all the computers in your company.
- Authorize and deploy the updates to the appropriate computers.
- Track the inventory and update installation status and progress for all the computers in your company.

In addition to general considerations that were explained in this article, you should also consider each product's best practice, for example: if you have a VM in Azure with SQL Server, make sure that you are following the software updates recommendation for that product.

Next steps

Use the guidelines described in this article to assist you in determining the best options for software updates for virtual machines within Azure IaaS. There are many similarities between software update best practices in a traditional datacenter versus Azure IaaS, therefore it is recommended that you evaluate your current software update policies to include Azure VMs and include the relevant best practices from this article in your overall software update process.

Introduction to Azure Security Center

11/22/2016 • 5 min to read • Edit on GitHub

Contributors

TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • 4c74356b41 • Rebecca Agiewich

Learn about Azure Security Center, its key capabilities, and how it works.

> **NOTE**
> This document introduces the service by using an example deployment.

What is 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.

Key capabilities

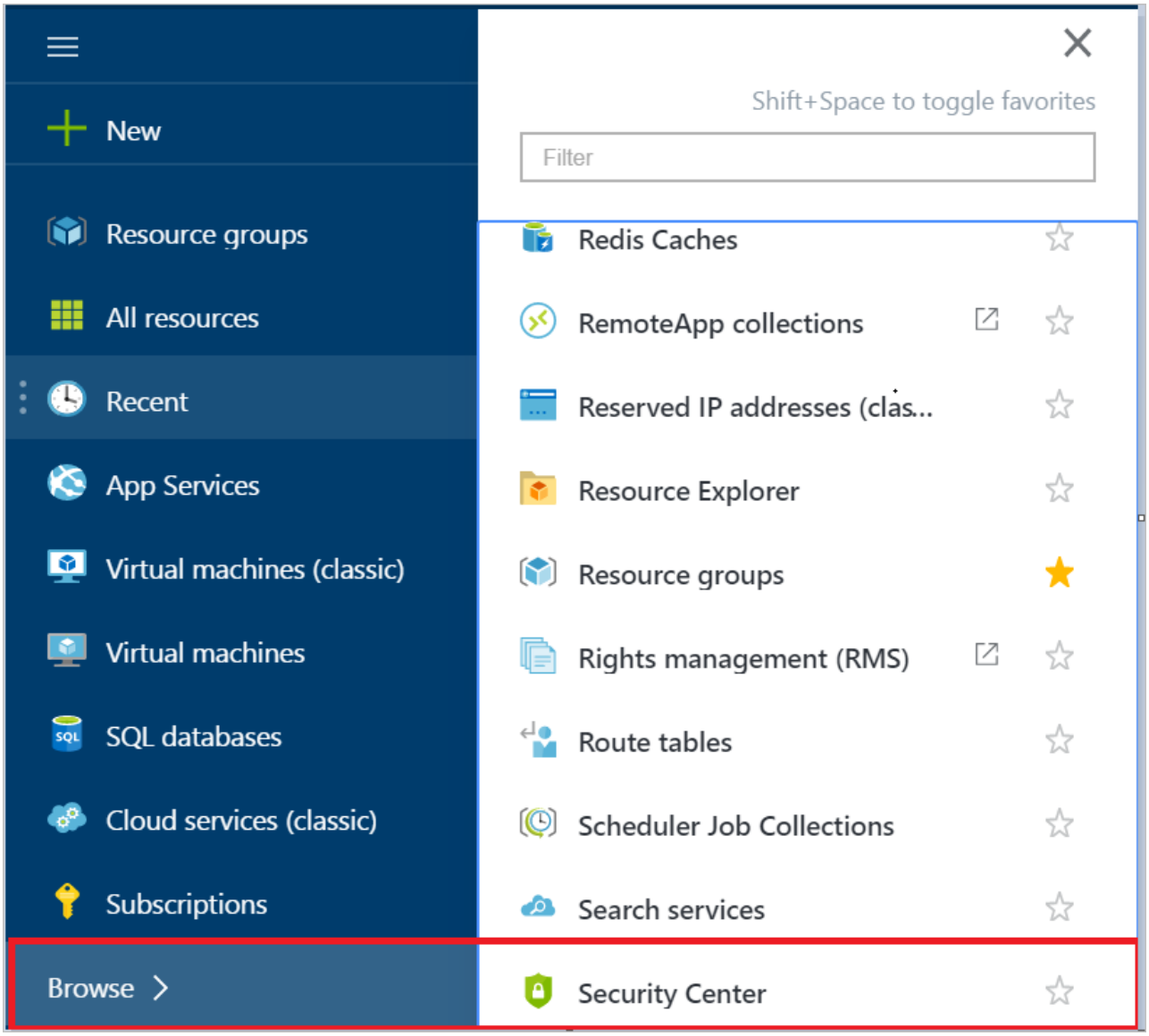
Security Center delivers easy-to-use and effective threat prevention, detection, and response capabilities that are built in to Azure. Key capabilities are:

STAGE	CAPABILITY
Prevent	Monitors the security state of your Azure resources
Prevent	Defines policies for your Azure subscriptions and resource groups based on your company's security requirements, the types of applications that you use, and the sensitivity of your data
Prevent	Uses policy-driven security recommendations to guide service owners through the process of implementing needed controls
Prevent	Rapidly deploys security services and appliances from Microsoft and partners
Detect	Automatically collects and analyzes security data from your Azure resources, the network, and partner solutions like antimalware programs and firewalls
Detect	Leverages global threat intelligence from Microsoft products and services, the Microsoft Digital Crimes Unit (DCU), the Microsoft Security Response Center (MSRC), and external feeds
Detect	Applies advanced analytics, including machine learning and behavioral analysis

STAGE	CAPABILITY
Respond	Provides prioritized security incidents/alerts
Respond	Offers insights into the source of the attack and impacted resources
Respond	Suggests ways to stop the current attack and help prevent future attacks

Introductory walkthrough

You access Security Center from the Azure portal. Sign in to the portal, select **Browse**, and then scroll to the **Security Center** option or select the **Security Center** tile that you previously pinned to the portal dashboard.



From Security Center, you can set security policies, monitor security configurations, and view security alerts.

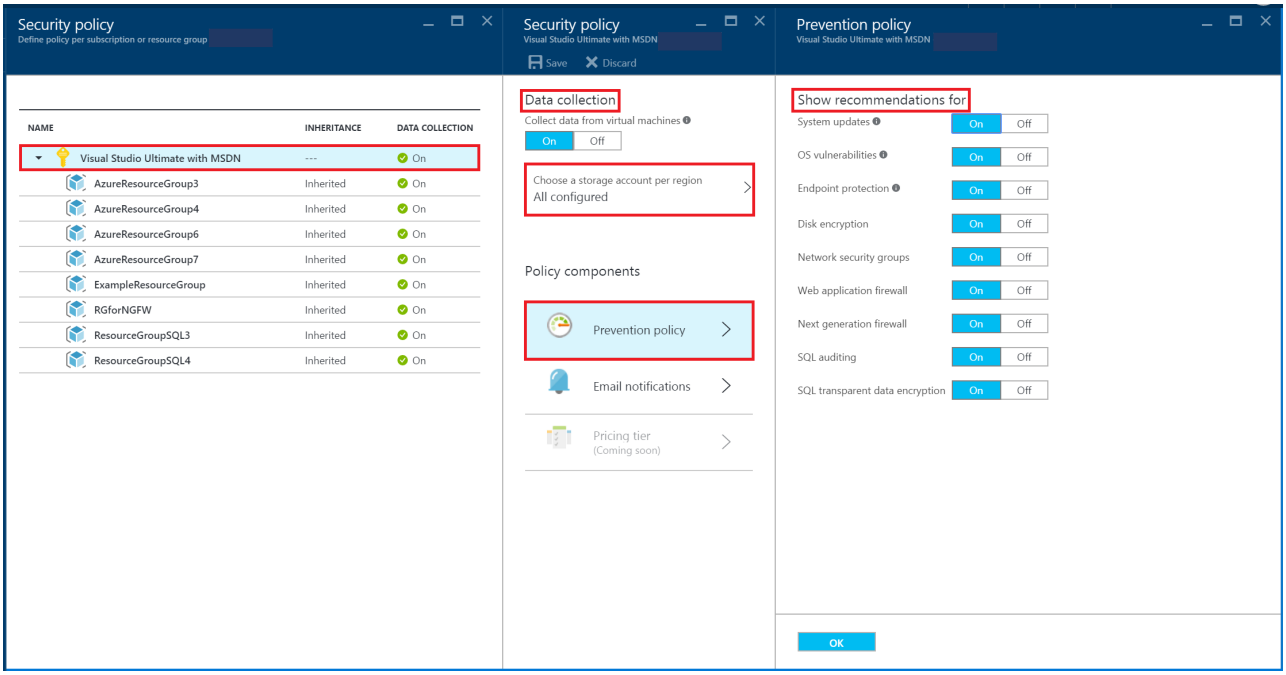
Security policies

You can define policies for your Azure subscriptions and resource groups according to your company's security requirements. You can also tailor them to the types of applications you're using or to the sensitivity of the data in each subscription. For example, resources used for development or testing may have different security requirements than those used for production applications. Likewise, applications with regulated data like PII may require a higher level of security.

> **NOTE**
>
> To modify a security policy at the subscription level or the resource group level, you must be the Owner of the subscription or a Contributor to it.

On the **Security Center** blade, select the **Policy** tile for a list of your subscriptions and resource groups.

On the **Security policy** blade, select a subscription to view the policy details.



Data collection (see above) enables data collection for a security policy. Enabling provides:

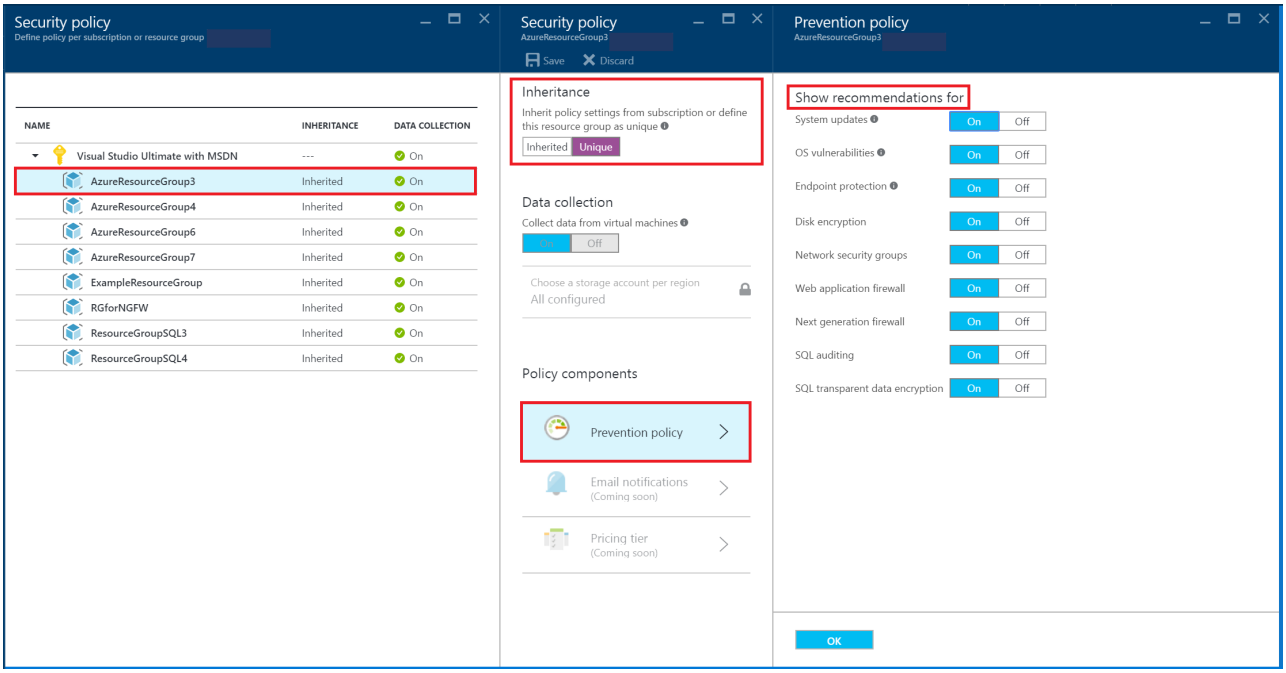
- Daily scanning of all supported virtual machines (VMs) for security monitoring and recommendations.
- Collection of security events for analysis and threat detection.

Choose a storage account per region (see above) lets you choose, for each region in which you have VMs running, the storage account where data collected from those VMs is stored. If you do not choose a storage account for each region, it is created for you. The data that's collected is logically isolated from other customers' data for security reasons.

> **NOTE**
> Data collection and choosing a storage account per region is configured at the subscription level.

Select **Prevention policy** (see above) to open the **Prevention policy** blade. **Show recommendations for** lets you choose the security controls that you want to monitor and recommend based on the security needs of the resources within the subscription.

Next, select a resource group to view policy details.



Inheritance (see above) lets you define the resource group as:

- Inherited (default) which means all security policies for this resource group are inherited from the subscription level.
- Unique which means the resource group has a custom security policy. You need to make changes under **Show recommendations for**.

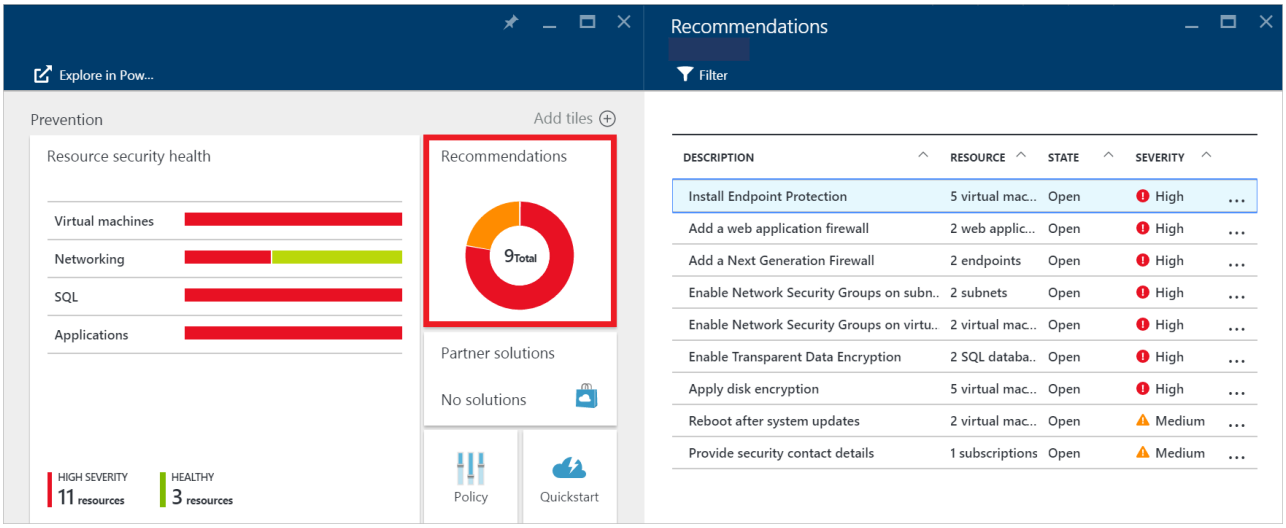
> **NOTE**
>
> If there is a conflict between subscription level policy and resource group level policy, the resource group level policy takes precedence.

Security recommendations

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

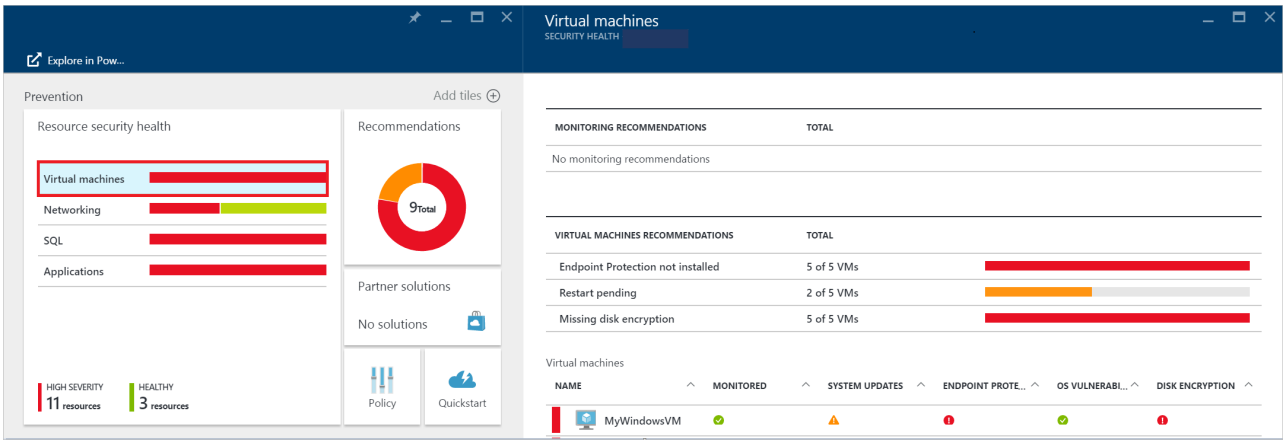
Click the **Recommendations** tile for a list of recommendations. Click each recommendation to view additional information or to take action to resolve the issue.



Resource health

The **Resource security health** tile shows the overall security posture of the environment by resource type, including VMs, web applications, and other resources.

Select a resource type on the **Resource security health** tile to view more information, including a list of any potential security vulnerabilities that have been identified. (**Virtual machines** is selected in the example below.)

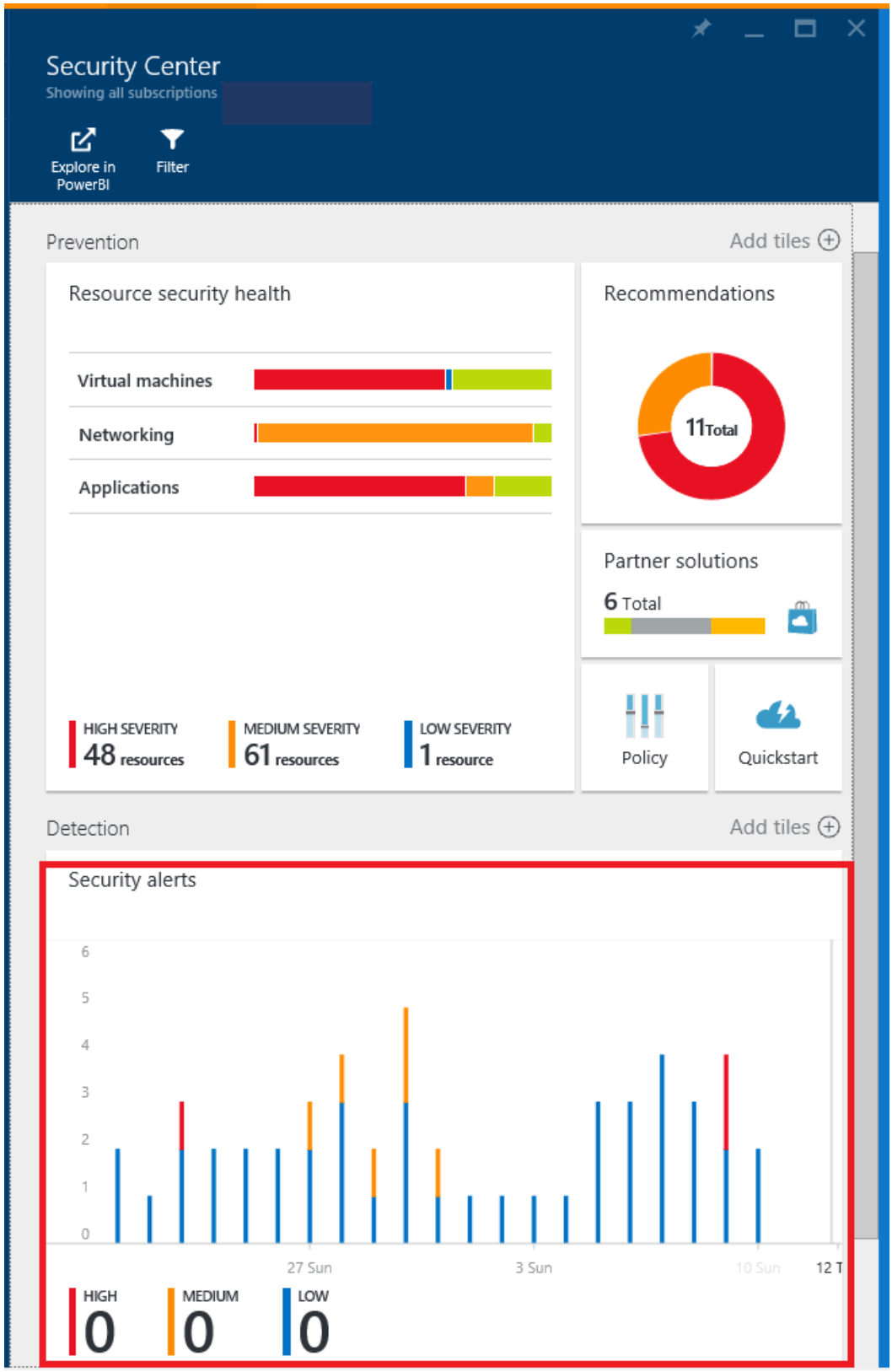


Security alerts

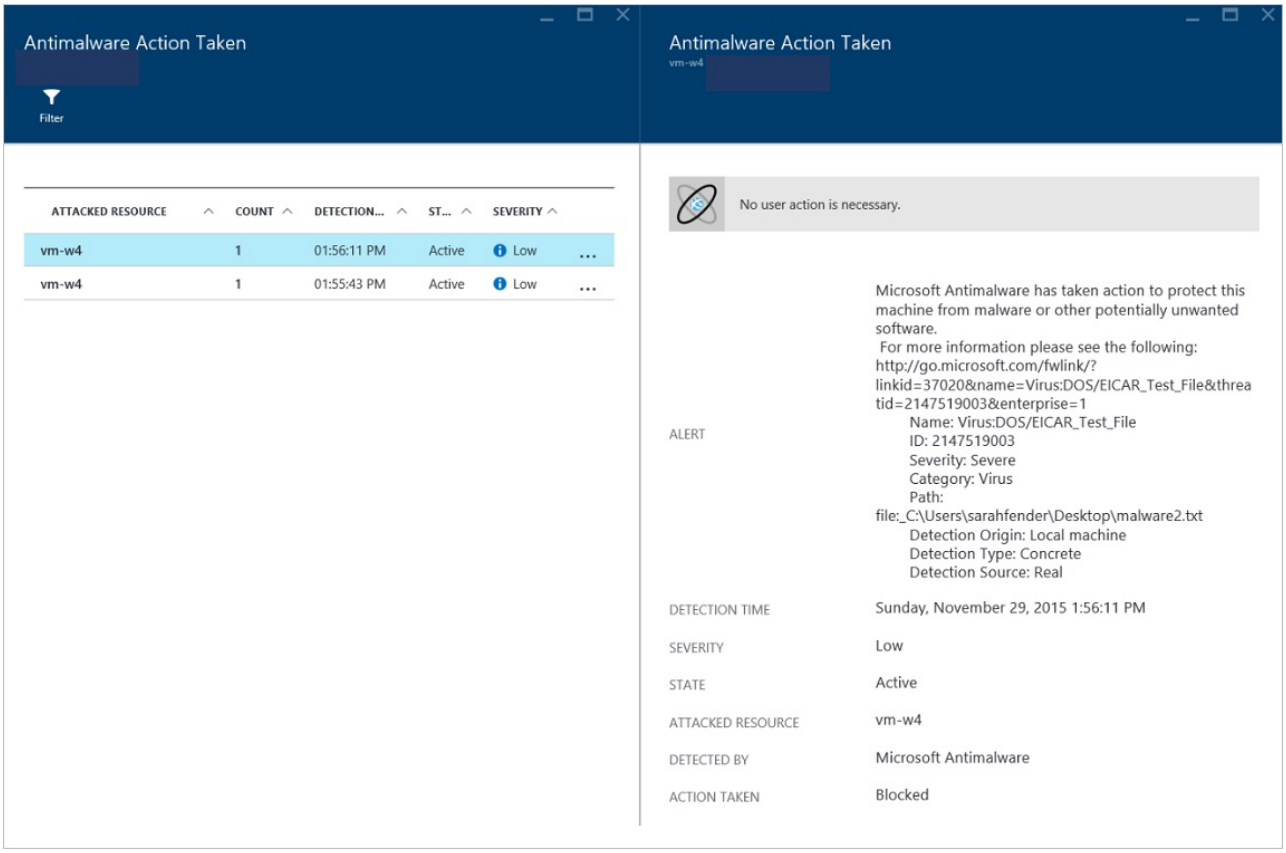
Security Center automatically collects, analyzes, and integrates log data from your Azure resources, the network, and partner solutions like antimalware programs and firewalls. When threats are detected, a security alert is created. Examples include detection of:

- 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

Clicking the **Security alerts** tile displays a list of prioritized alerts.



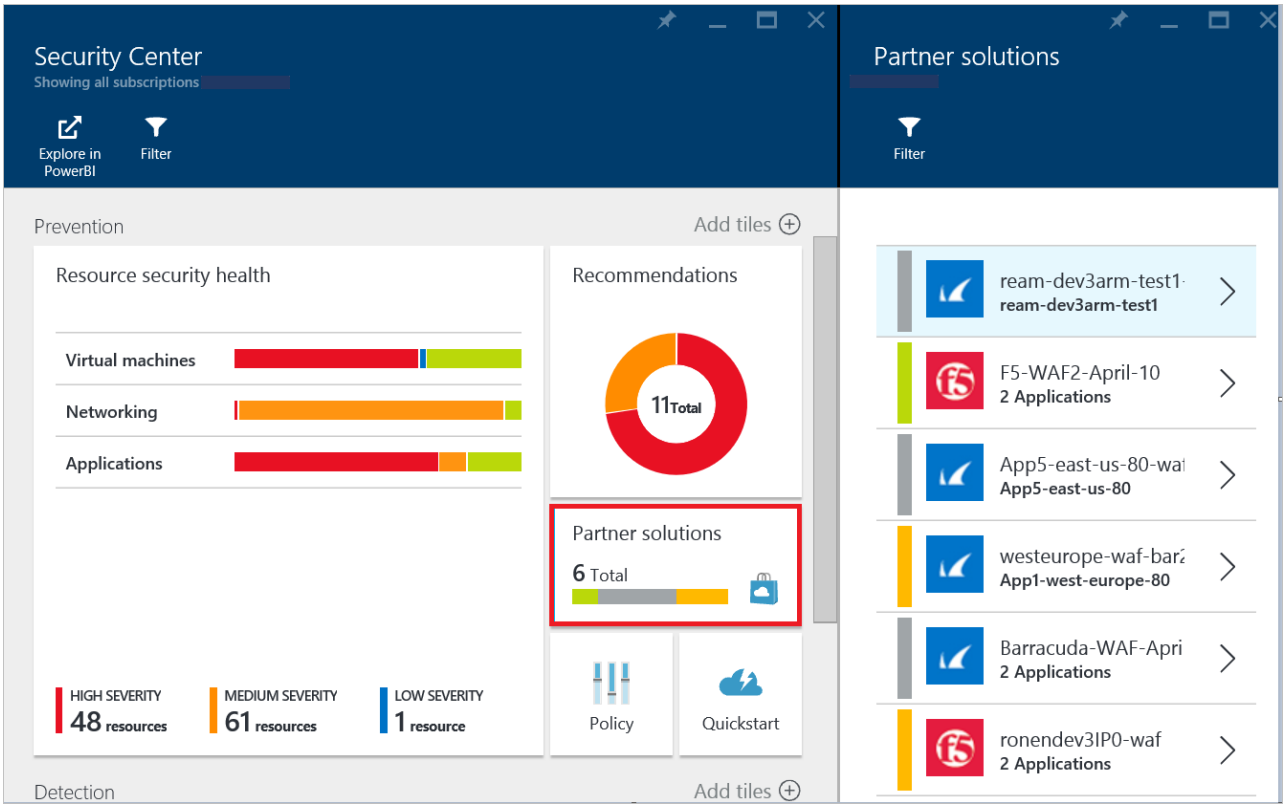
Selecting an alert shows more information about the attack and suggestions for how to remediate it.



Partner solutions

The **Partner solutions** tile lets you monitor at a glance the health status of your partner solutions integrated with your Azure subscription. Security Center displays alerts coming from the solutions.

Select the **Partner solutions** tile. A blade opens displaying a list of all connected partner solutions.



Get started

To get started with Security Center, you need a subscription to Microsoft Azure. Security Center is enabled with your Azure subscription. If you do not have a subscription, you can sign up for a free trial.

You access Security Center from the Azure portal. See the portal documentation to learn more.

Getting started with Azure Security Center quickly guides you through the security-monitoring and policy-management components of Security Center.

See also

In this document, you were introduced to Security Center, its key capabilities, and how to get started. To learn more, see the following:

- Setting security policies in Azure Security Center — Learn how to configure security policies for your Azure subscriptions and resource groups.
- Managing security recommendations in Azure Security Center — Learn how recommendations help you protect your Azure resources.
- Security health monitoring in Azure Security Center — Learn how to monitor the health of your Azure resources.
- Managing and responding to security alerts in Azure Security Center — Learn how to manage and respond to security alerts.
- Monitoring partner solutions with Azure Security Center — Learn how to monitor the health status of your partner solutions.
- Azure Security Center FAQ — Find frequently asked questions about using the service.
- Azure Security blog — Get the latest Azure security news and information.

Introduction to Microsoft Azure log integration (Preview)

11/15/2016 • 2 min to read • Edit on GitHub

Contributors

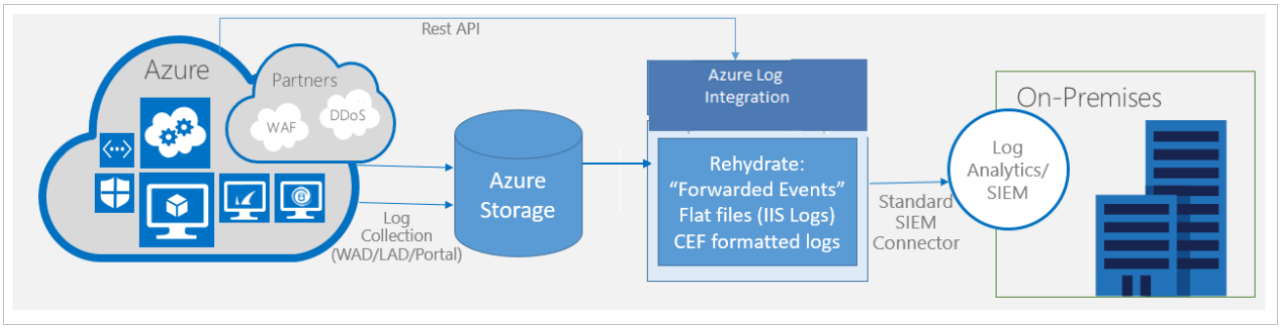
Thomas W. Shinder, M.D • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear

Learn about Azure log integration, its key capabilities, and how it works.

Overview

Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) hosted in Azure generate a large amount of data in security logs. These logs contain vital information that can provide intelligence and powerful insights into policy violations, internal and external threats, regulatory compliance issues, and anomalies in network, host, and user activity.

Azure log integration enables you to integrate raw logs from your Azure resources into your on-premises Security Information and Event Management (SIEM) systems. Azure log integration collects Azure Diagnostics from your Windows *(WAD)* virtual machines, as well as diagnostics from partner solutions such as a Web Application Firewall (WAF). This integration provides a unified dashboard for all your assets, on-premises or in the cloud, so that you can aggregate, correlate, analyze, and alert for security events.



What logs can I integrate?

Azure produces extensive logging for every Azure service. These logs are categorized by two main types:

- **Control/management logs**, which give visibility into the Azure Resource Manager CREATE, UPDATE, and DELETE operations. Azure Audit Logs is an example of this type of log.
- **Data plane logs**, which give visibility into the events raised as part of the usage of an Azure resource. Examples of this type of log are the Windows event System, Security, and Application logs in a virtual machine.

Azure log integration currently supports integration of Azure Audit Logs, virtual machine logs, and Azure Security Center alerts.

If you have questions about Azure Log Integration, please send an email to AzSIEMteam@microsoft.com

Next steps

In this document, you were introduced to Azure log integration. To learn more about Azure log integration and the types of logs supported, see the following:

- Microsoft Azure Log Integration for Azure logs (Preview) – Download Center for details, system requirements, and install instructions on Azure log integration.
- Get started with Azure log integration - This tutorial walks you through installation of Azure log integration and integrating logs from Azure storage, Azure Audit Logs, and Security Center alerts.
- 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 frequently asked questions (FAQ) - This FAQ answers questions about Azure log integration.
- Integrating Security Center alerts with Azure log Integration – This document shows you how to sync Security Center alerts, along with virtual machine security events collected by Azure Diagnostics and Azure Audit Logs, with your 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 help you gain insights into the operations of your Azure resources.

Get started with Azure log integration (Preview)

11/15/2016 • 6 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Ralph Squillace • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear

---

Azure log integration enables you 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, so that you can aggregate, correlate, analyze, and alert for security events associated with your applications.

This tutorial walks you through how to install Azure log integration and integrate logs from Azure storage, Azure Audit Logs, and Azure Security Center alerts. Estimated time to complete this tutorial is one hour.

Prerequisites

To complete this tutorial, you must have the following:

- A machine (on-premises or in the cloud) to install the Azure log integration service. This machine must be running a 64-bit Windows OS with .Net 4.5.1 installed. This machine is called the **Azlog Integrator**.
- Azure subscription. If you do not have one, you can sign up for a free account.
- Azure Diagnostics enabled for your Azure virtual machines (VMs). To enable diagnostics for Cloud Services, see Enabling Azure Diagnostics in Azure Cloud Services. To enable diagnostics for an Azure VM running Windows, see Use PowerShell to enable Azure Diagnostics in a Virtual Machine Running Windows.
- Connectivity from the Azlog Integrator to Azure storage and to authenticate and authorize to Azure subscription.
- For Azure VM logs, the SIEM agent (for example, Splunk Universal Forwarder, HP ArcSight Windows Event Collector agent, or IBM QRadar WinCollect) must be installed on the Azlog Integrator.

Deployment considerations

You can run multiple instances of the Azlog Integrator if event volume is high. Load balancing of Azure Diagnostics storage accounts for Windows *(WAD)* and the number of subscriptions to provide to the instances should be based on your capacity.

On an 8-processor (core) machine, a single instance of Azlog Integrator can process about 24 million events per day (~1M/hour).

On a 4-processor (core) machine, a single instance of Azlog Integrator can process about 1.5 million events per day (~62.5K/hour).

Install Azure log integration

Download Azure log integration.

The Azure log integration service collects telemetry data from the machine on which it is installed. Telemetry data collected is:

- 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 being used

> **NOTE**
> You can turn off collection of telemetry data by unchecking this option.

Integrate Azure VM logs from your Azure Diagnostics storage accounts

1. Check the prerequisites listed above to ensure that your WAD storage account is collecting logs before continuing your Azure log integration. Do not perform the following steps if your WAD storage account is not collecting logs.
2. Open the command prompt and **cd** into **c:\Program Files\Microsoft Azure Log Integration**.
3. Run the command

   ```
   azlog source add <FriendlyNameForTheSource> WAD <StorageAccountName> <StorageKey>
   ```

   Replace StorageAccountName with the name of the Azure storage account configured to receive diagnostics events from your VM.

   ```
   azlog source add azlogtest WAD azlog9414
   fxxxFxxxxxxxxywoEJK2xxxxxxxxxixxxJ+xVJx6m/X5SQDYc4Wpjpli9S9Mm+vXS2RVYtp1mes0t9H5cuqXEw==
   ```

   If you would like 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>
   ```

4. Wait 30 - 60 minutes (it could take as long as an hour), then view the events that are pulled from the storage account. To view, open **Event Viewer > Windows Logs > Forwarded Events** on the Azlog Integrator.
5. Make sure that your standard SIEM connector installed on the machine is configured to pick events from the **Forwarded Events** folder and pipe them to your SIEM instance. Review the SIEM specific configuration to configure and see the logs integrating.

What if data is not showing up in the Forwarded Events folder?

If after an hour data is not showing up in the **Forwarded Events** folder, then:

1. Check the machine and confirm that it can access Azure. To test connectivity, try to open the Azure portal from the browser.
2. Make sure the user account **azlog** has write permission on the folder **users\azlog**.
3. Make sure the storage account added in the command **azlog source add** is listed when you run the command **azlog source list**.
4. Go to **Event Viewer > Windows Logs > Application** to see if there are any errors reported from the Azure log integration.

If you still don't see the events, then:

1. Download Microsoft Azure Storage Explorer.
2. Connect to the storage account added in the command **azlog source add**.
3. In Microsoft Azure Storage Explorer, browse to table **WADWindowsEventLogsTable** to see if there is any data. If not, then diagnostics in the VM is not configured correctly.

Integrate Azure audit logs and Security Center alerts

1. Open the command prompt and **cd** into **c:\Program Files\Microsoft Azure Log Integration**.
2. Run the 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 Active Directory (AD). Creating a service principal for Azlog Integration creates the Azure AD identity that will be given access to read from Azure subscriptions.

3. Run the command

   ```
   azlog authorize <SubscriptionID>
   ```

   This assigns reader access on the subscription to the service principal created in step 2. If you don't specify a SubscriptionID, then it attempts to assign the service principal reader role to all subscriptions to which you have any access.

   ```
   azlog authorize 0ee9d577-9bc4-4a32-a4e8-c29981025328
   ```

   > **NOTE**
   > You may see warnings if you run the **authorize** command immediately after the **createazureid** command. There is some latency between when the Azure AD account is created and when the account is available for use. If you wait about 10 seconds after running the **createazureid** command to run the **authorize** command, then you should not see these warnings.

4. Check the following folders to confirm that the Audit log JSON files are there:

   - **c:\Users\azlog\AzureResourceManagerJson**
   - **c:\Users\azlog\AzureResourceManagerJsonLD**

5. Check the following folders to confirm that Security Center alerts exist in them:

   - **c:\Users\azlog\ AzureSecurityCenterJson**
   - **c:\Users\azlog\AzureSecurityCenterJsonLD**

6. Point the standard SIEM file forwarder connector to the appropriate folder to pipe the data to the SIEM instance. You may need some field mappings based on the SIEM product you are using.

If you have questions about Azure Log Integration, please send an email to AzSIEMteam@microsoft.com

Next steps

In this tutorial, you learned how to install Azure log integration and integrate logs from Azure storage. To learn more, see the following:

- Microsoft Azure Log Integration for Azure logs (Preview) – Download Center for details, system requirements, and install instructions on Azure log integration.
- Introduction to Azure log integration – This document 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.

- Azure log Integration frequently asked questions (FAQ) - This FAQ answers questions about Azure log integration.
- Integrating Security Center alerts with Azure log Integration – This document shows you how to sync Security Center alerts, along with virtual machine security events collected by Azure Diagnostics and Azure Audit Logs, with your 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 help you gain insights into the operations of your Azure resources.

Azure log integration frequently asked questions (FAQ)

11/15/2016 • 3 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Ralph Squillace • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Rob Boucher • TerryLanfear

This FAQ answers questions about Azure log integration, a service that enables you 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, so that you can aggregate, correlate, analyze, and alert for security events associated with your applications.

How can I see the storage accounts from which Azure log integration is pulling Azure VM logs from?

Run the command **azlog source list**.

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 update is done, stop and start the service using commands **net stop azlog** and **net start azlog**.

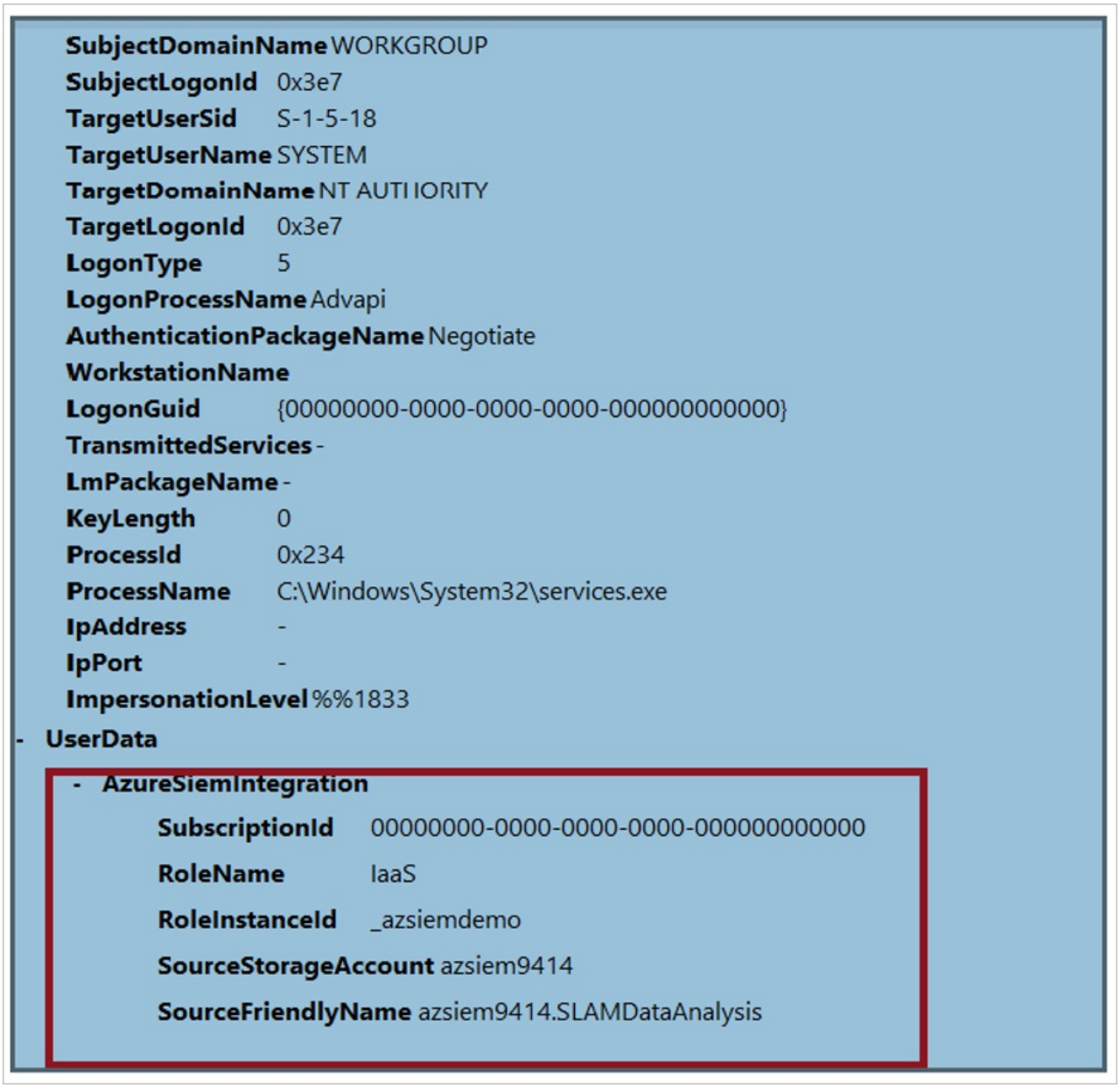
```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.net>
    <connectionManagement>
      <add address="*" maxconnection="400" />
    </connectionManagement>
    <defaultProxy>
      <proxy usesystemdefault="true"
      proxyaddress=http://127.0.0.1:8888
      bypassonlocal="true" />
    </defaultProxy>
  </system.net>
  <system.diagnostics>
    <performanceCounters filemappingsize="20971520" />
  </system.diagnostics>
```

How can I see the subscription information in Windows events?

Append the **subscriptionid** to the friendly name while adding the source.

```
Azlog source add <sourcefriendlyname>.<subscription id> <StorageName> <StorageKey>
```

The event XML has the metadata as shown below, including the subscription id.



Error messages

When running 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.'

Azlog createazureid attempts to create a service principal in all the Azure AD tenants for the subscriptions on which the Azure login has access to. If your Azure login is only a Guest user in that Azure AD tenant, then the command fails with 'Insufficient privileges to complete the operation.' Request Tenant admin to add your account as a user in the tenant.

When running 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'.

Azlog authorize command assigns the role of Reader to the Azure AD service principal (created with **Azlog**

createazureid) to the subscriptions provided. If the Azure login is not a Co-Administrator or an Owner of the subscription, it fails with '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 audit log?

See:

- Audit operations with Resource Manager
- List the management events in a subscription in 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?

See Use PowerShell to enable Azure Diagnostics in a virtual machine running Windows for details on how to Get, Modify, and Set the Azure Diagnostics in Windows *(WAD)* configuration. Following is a sample:

Get the WAD config

```
-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"
```

Modify the WAD Config

The following example is a configuration where only EventID 4624 and EventId 4625 are collected from the security event log. Microsoft Antimalware events are collected from the System event log. See [Consuming Events] (https://msdn.microsoft.com/library/windows/desktop/dd996910(v=vs.85) for details on the use of XPath expressions.

```
<WindowsEventLog scheduledTransferPeriod="PT1M">
    <DataSource name="Security!*[System[(EventID=4624 or EventID=4625)]]" />
    <DataSource name="System!*[System[Provider[@Name='Microsoft Antimalware']]]"/>
</WindowsEventLog>
```

Set the WAD configuration

```
$diagnosticsconfig_path = "d:\WADConfig.xml"
Set-AzureRmVMDiagnosticsExtension -ResourceGroupName AzLog-Integration -VMName AzlogClient -
DiagnosticsConfigurationPath $diagnosticsconfig_path -StorageAccountName log3121 -StorageAccountKey <storage
key>
```

After making changes, check the storage account to ensure that the correct events are collected.

If you have questions about Azure Log Integration, please send an email to AzSIEMteam@microsoft.com

Azure identity management security overview

11/22/2016 • 7 min to read • Edit on GitHub

Contributors

TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • curtand

---

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 (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 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
- Security monitoring, alerts, and machine learning-based reports
- Consumer identity and access management
- Device registration
- Privileged identity management
- Identity protection
- Hybrid identity management

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 of 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.

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 single sign-on (SSO) for many types of on-premises web applications with the thousands of SaaS applications that Azure AD supports. Employees can log 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 (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.

Learn more:

- Multi-factor authentication
- What is Azure Multi-Factor Authentication?
- How Azure Multi-Factor Authentication works

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 classic 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 make a determination 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.

Learn more:

- View your access and usage reports
- Getting started with Azure Active Directory Reporting
- Azure Active Directory Reporting Guide

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).

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 Active Directory 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.

Learn more:

- Get started with Azure Active Directory Device Registration
- Setting up on-premises conditional access using Azure Active Directory Device Registration
- Automatic device registration with Azure Active Directory for Windows domain-joined devices

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

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 affecting your organization's identities. Identity Protection leverages 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.

Learn more:

- Azure Active Directory Identity Protection
- Channel 9: Azure AD and Identity Show: Identity Protection Preview

Hybrid identity management

Microsoft's approach to identity spans on-premises and the cloud, creating a single user identity for authentication and authorization to all resources, regardless of location.

Learn more:

- Hybrid identity white paper
- Azure Active Directory
- Active Directory Team Blog

Azure Identity Management and access control security best practices

11/22/2016 • 10 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • 4c74356b41 • Ralph Squillace • TerryLanfear

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 will 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'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 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:

- Centralize your identity management
- Enable Single Sign-On (SSO)
- Deploy password management
- Enforce multi-factor authentication (MFA) for users
- Use role based access control (RBAC)
- Control locations where resources are created using resource manager
- Guide developers to leverage identity capabilities for SaaS apps
- Actively monitor for suspicious activities

Centralize your identity management

One important step towards securing your identity is to ensure that IT can manage accounts from one single location regarding where this account was created. While the majority of the enterprises IT organizations will have their primary account directory on-premises, hybrid cloud deployments are on the rise and it is important that you understand how to integrate on-premises and cloud directories and provide a seamless experience to the end user.

To accomplish this hybrid identity scenario we recommend two options:

- Synchronize your on-premises directory with your cloud directory using Azure AD Connect
- Federate your on-premises identity with your cloud directory using Active Directory Federation Services (AD FS)

Organizations that fail to integrate their on-premises identity with their cloud identity will experience increased administrative overhead in managing accounts, which increases the likelihood of mistakes and security breaches.

For more information on Azure AD synchronization, please read the article Integrating your on-premises identities with Azure Active Directory.

Enable Single Sign-On (SSO)

When you have multiple directories to manage, this becomes an administrative problem not only for IT, but also for end users that will have to remember multiple passwords. By using SSO you will provide your users the ability of use the same set of credentials to sign-in and access the resources that they need, regardless where this resource is located on-premises or in the cloud.

Use SSO to enable users to access their SaaS applications based on their organizational account in Azure AD. This is applicable not only for Microsoft SaaS apps, but also other apps, such as Google Apps and Salesforce. Your application can be configured to use Azure AD as a SAML-based identity provider. As a security control, Azure AD will not issue a token allowing them to sign into the application unless they have been granted access using Azure AD. You may grant access directly, or through a group that they are a member of.

> **NOTE**
> the decision to use SSO will impact how you integrate your on-premises directory with your cloud directory. If you want SSO, you will need to use federation, because directory synchronization will only provide same sign-on experience.

Organizations that do not enforce SSO for their users and applications are more exposed to scenarios where users will have multiple passwords which directly increases the likelihood of users reusing passwords or using weak passwords.

You can learn more about Azure AD SSO by reading the article AD FS management and customization with Azure AD Connect.

Deploy password management

In scenarios where you have multiple tenants or you want to enable users to reset their own password, it is important that you use appropriate security policies to prevent abuse. In Azure you can leverage the self-service password reset capability and customize the security options to meet your business requirements.

It is particularly important to obtain feedback from these users and learn from their experiences as they try to perform these steps. Based on these experiences, elaborate a plan to mitigate potential issues that may occur during the deployment for a larger group. It is also recommended that you use the Password Reset Registration Activity report to monitor the users that are registering.

Organizations that want to avoid password change support calls but do enable users to reset their own passwords are more susceptible to a higher call volume to the service desk due to password issues. In organizations that have multiple tenants, it is imperative that you implement this type of capability and enable users to perform password reset within security boundaries that were established in the security policy.

You can learn more about password reset by reading the article Deploying Password Management and training users to use it.

Enforce multi-factor authentication (MFA) for users

For organizations that need to be compliant with industry standards, such as PCI DSS version 3.2, multi-factor authentication is a must have capability for authenticate users. Beyond being compliant with industry standards, enforcing MFA to authenticate users can also help organizations to mitigate credential theft type of attack, such as

Pass-the-Hash (PtH).

By enabling Azure MFA for your users, you are adding a second layer of security to user sign-ins and transactions. In this case, a transaction might be accessing a document located in a file server or in your SharePoint Online. Azure MFA also helps IT to 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.

For more information on Azure MFA, please read the article Getting started with Azure Multi-Factor Authentication in the cloud.

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. Azure Role-Based Access Control (RBAC) can be used 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 leverage built in RBAC roles in Azure to assign privileges to users. Consider using *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 needs to manage VMs and storage account, consider adding them to *Virtual Machine Contributor* role.

Organizations that do not enforce data access control by leveraging capabilities such as RBAC may be giving more privileges than necessary to their users. This can lead to data compromise by allow users access to certain types of types of data (e.g., high business impact) that they shouldn't have in the first place.

You can learn more about Azure RBAC by reading the article Azure Role-Based Access Control.

Control locations where resources are created using resource manager

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.

To achieve this, organizations can create security policies that have definitions that describe the actions or resources that are specifically denied. You assign those policy definitions at the desired scope, such as the subscription, resource group, or an individual resource.

> **NOTE**
> this is not the same as RBAC, it actually leverages RBAC to authenticate the users that have privilege to create those resources.

Leverage Azure Resource Manager to create custom policies also for scenarios where the organization wants to allow operations only when the appropriate cost center is associated; otherwise, they will deny the request.

Organizations that are not controlling how resources are created are more susceptible to users that may abuse the

service by creating more resources than they need. Hardening the resource creation process is an important step to secure a multi-tenant scenario.

You can learn more about creating policies with Azure Resource Manager by reading the article Use Policy to manage resources and control access.

Guide developers to leverage identity capabilities for SaaS apps

User identity will be leveraged in many scenarios when users access SaaS apps that can be integrated with on-premises or cloud directory. First and foremost, we recommend that developers use a secure methodology to develop these apps, such as Microsoft Security Development Lifecycle (SDL). 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, as well as open source libraries for different platforms.

Make sure to register any application that outsources authentication to Azure AD, this is a mandatory procedure. The reason behind this is because Azure AD needs to coordinate the communication with the application when handling sign-on (SSO) or exchanging tokens. The user's session expires when the lifetime of the token issued by Azure AD expires. Always evaluate if your application should use this time or if you can reduce this time. Reducing the lifetime can act as a security measure that will force users to sign out based on a period of inactivity.

Organizations that do not enforce identity control to access apps and do not guide their developers on how to securely integrate apps with their identity management system may be more susceptible to credential theft type of attack, such as weak authentication and session management described in Open Web Application Security Project (OWASP) Top 10.

You can learn more about authentication scenarios for SaaS apps by reading Authentication Scenarios for Azure AD.

Actively monitor for suspicious activities

According to Verizon 2016 Data Breach report, credential compromise is still in the rise and becoming one of the most profitable businesses for cyber criminals. For this reason, it is important to have an active identity monitor system in place that can quickly detect suspicious behavior activity and trigger an alert for further investigation. Azure AD has two major capabilities that can help organizations monitor their identities: Azure AD Premium anomaly reports and Azure AD identity protection capability.

Make sure to use the anomaly reports to identify attempts to sign in without being traced, brute force attacks against a particular account, attempts to sign in from multiple locations, sign in from infected devices and suspicious IP addresses. Keep in mind that these are reports. In other words, you must have processes and procedures in place for IT admins to run these reports on the daily basis or on demand (usually in an incident response scenario).

In contrast, Azure AD identity protection is an active monitoring system and it will flag the current risks on its own dashboard. Besides that, you will also receive daily summary notifications via email. We recommend that you adjust the risk level according to your business requirements. The risk level for a risk event is an indication (High, Medium, or Low) of the severity of the risk event. The risk level helps Identity Protection users prioritize the actions they must take to reduce the risk to their organization.

Organizations that do not actively monitor their identity systems are at risk of having user credentials compromised. Without knowledge that suspicious activities are taking place using these credentials, organizations won't be able to mitigate this type of threat. You can learn more about Azure Identity protection by reading Azure Active Directory Identity Protection.

Internet of Things Security Best Practices

11/15/2016 • 2 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes • Barclay Neira

---

Securing the Internet of Things (IoT) infrastructure is a critical undertaking for anyone involved with IoT solutions. Because of the number of devices involved and the distributed nature of these devices, the impact a security event related to compromise of millions of IoT devices is non-trivial and can have widespread impact.

For this reason, IoT security needs a security-in-depth approach. Data needs to be secure in the cloud and as it moves over private and public networks. Methods need to be in place to securely provision the IoT devices themselves. Each layer, from device, to network, to cloud back-end needs strong security assurances.

IoT best practices can be categorized in the following way:

- IoT hardware manufacturer or integrator
- IoT solution developer
- IoT solution deployer
- IoT solution operator

This article summarizes Internet of Things Security Best Practices. Please refer to that article for more detailed information.

IoT hardware manufacturer or integrator

Follow the best practices below if you are an IoT hardware manufacture or a hardware integrator:

- **Scope hardware to minimum requirements**: the hardware design should include minimum features required for operation of the hardware, and nothing more.
- **Make hardware tamper proof**: build in mechanisms to detect physical tampering of hardware, such as opening the device cover, removing a part of the device, etc.
- **Build around secure hardware**: if COGS permit, build security features such as secure and encrypted storage and Trusted Platform Module (TPM)-based boot functionality.
- **Make upgrades secure**: upgrading firmware during lifetime of the device is inevitable.

IoT solution developer

Follow the best practices below if you are an IoT solution developer:

- **Follow secure software development methodology**: developing secure software requires ground-up thinking about security from the inception of the project all the way to its implementation, testing, and deployment.
- **Choose open source software with care**: open source software provides an opportunity to quickly develop solutions.
- **Integrate with care**: many of the software security flaws exist at the boundary of libraries and APIs.

IoT solution deployer

Follow the best practices below if you are an IoT solution deployer:

- **Deploy hardware securely**: IoT deployments may require hardware to be deployed in unsecure locations, such as in public spaces or unsupervised locales.
- **Keep authentication keys safe**: during deployment, each device requires device IDs and associated authentication keys generated by the cloud service. Keep these keys physically safe even after the deployment. Any compromised key can be used by a malicious device to masquerade as an existing device.

IoT solution operator

Follow the best practices below if you are an IoT solution operator:

- **Keep systems up to date**: ensure device operating systems and all device drivers are updated to the latest versions.
- **Protect against malicious activity**: if the operating system permits, place the latest anti-virus and anti-malware capabilities on each device operating system.
- **Audit frequently**: auditing IoT infrastructure for security related issues is key when responding to security incidents.
- **Physically protect the IoT infrastructure**: the worst security attacks against IoT infrastructure are launched using physical access to devices.
- **Protect cloud credentials**: cloud authentication credentials used for configuring and operating an IoT deployment are possibly the easiest way to gain access and compromise an IoT system.

Internet of Things security overview

11/22/2016 • 2 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil

---

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 Suite packages 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 suite is a powerful solution 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

A practical guide to designing secure health care solutions in Azure

11/15/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear

---

Health industry startups, system integrators (SIs), independent software vendors (ISVs), and healthcare organizations considering a move to Azure are looking for guidance that helps them incorporate security controls to meet their compliance obligations.

A Practical Guide to Designing Secure Health Care Solutions in Microsoft Azure helps you understand how you can improve security for your solutions by using the Azure services and features that you can configure based on your requirements. The content is divided into three major sections:

1. Considerations guidance for using cloud technology, including risk management, shared responsibility, establishing an information security management system, understanding industry and local regulations, and establishing standard operating procedures.
2. Key security principles that are both aligned to a standard information security management standard, such as ISO 27001, and standard development processes, such as Microsoft's Security Development Lifecycle (SDL).
3. Applying the key principles to use cases by demonstrating alignment from a solution architect perspective, where requirements for the solutions are aligned to the information security management standard.

We hope you find A Practical Guide to Designing Secure Health Care Solutions helpful and if you have any questions or suggestions, let us know by leaving a comment below.

Security architecture overview

11/22/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear • unknown

---

Having a strong architectural foundation is one of the keys to success when it comes to secure solution deployments in Azure. With this knowledge you're able to better understand your requirements by knowing the right questions to ask and more equipped to find the right answers to your questions. Getting right answers to the right questions goes a long way toward optimizing the security of your deployments.

In this section you'll see articles on Azure Security Architecture that will help you build secure solutions. A popular collection of Azure security best practices and patterns is also included. At this time, we have the following articles – make sure to visit our site and the Azure Security Team blog for updates on a regular basis:

- Data Classification for Cloud Readiness
- Application Architecture on Microsoft Azure
- Azure Security Best Practices and Patterns

Data classification for Azure

11/22/2016 • 16 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Barclay Neira

---

This article provides an introduction to the fundamentals of data classification and highlights its value, specifically in the context of cloud computing and using Microsoft Azure.

Data classification fundamentals

Successful data classification in an organization requires broad awareness of your organization's needs and a thorough understanding of where your data assets reside.

Data exists in one of three basic states:

- At rest
- In process
- In transit

All three states require unique technical solutions for data classification, but the applied principles of data classification should be the same for each. Data that is classified as confidential needs to stay confidential when at rest, in process, and in transit.

Data can also be either structured or unstructured. Typical classification processes for the structured data found in databases and spreadsheets are less complex and time-consuming to manage than those for unstructured data such as documents, source code, and email.

> **TIP**
> for more information regarding Azure capabilities and best practices for data encryption read Azure Data Encryption Best Practices

In general, organizations will have more unstructured data than structured data. Regardless of whether data is structured or unstructured, it is important for you to manage data sensitivity. When properly implemented, data classification helps ensure that sensitive or confidential data assets are managed with greater oversight than data assets that are considered public or free to distribute.

Controlling access to data

Authentication and authorization are often confused with each other and their roles misunderstood. In reality they are quite different, as shown in the following figure.

Authentication

Authentication typically consists of at least two parts: a username or user ID to identify a user and a token, such as a password, to confirm that the username credential is valid. The process does not provide the authenticated user with access to any items or services; it verifies that the user is who they say they are.

> **TIP**
> Azure Active Directory provides cloud-based identity services that allow you to authenticate and authorize users.

Authorization

Authorization is the process of providing an authenticated user the ability to access an application, data set, data file, or some other object. Assigning authenticated users the rights to use, modify, or delete items that they can access requires attention to data classification.

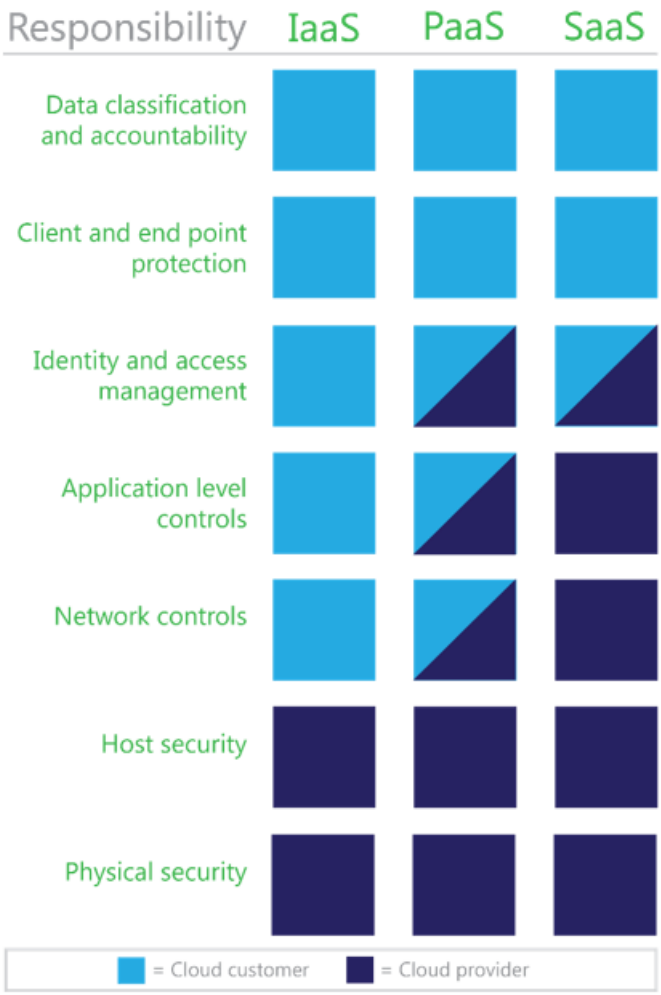
Successful authorization requires implementation of a mechanism to validate individual users' needs to access files and information based on a combination of role, security policy, and risk policy considerations. For example, data from specific line-of-business (LOB) applications might not need to be accessed by all employees, and only a small subset of employees will likely need access to human resources (HR) files. But for organizations to control who can access data, as well as when and how, an effective system for authenticating users must be in place.

> **TIP**
> in Microsoft Azure, make sure to leverage Azure Role-Based Access Control (RBAC) to grant only the amount of access that users need to perform their jobs. Read Use role assignments to manage access to your Azure Active Directory resources for more information.

Roles and responsibilities in cloud computing

Although cloud providers can help manage risks, customers need to ensure that data classification management and enforcement is properly implemented to provide the appropriate level of data management services.

Data classification responsibilities will vary based on which cloud service model is in place, as shown in the following figure. The three primary cloud service models are infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Implementation of data classification mechanisms will also vary based on the reliance on and expectations of the cloud provider.



Although you are responsible for classifying your data, cloud providers should make written commitments about how they will secure and maintain the privacy of the customer data stored within their cloud.

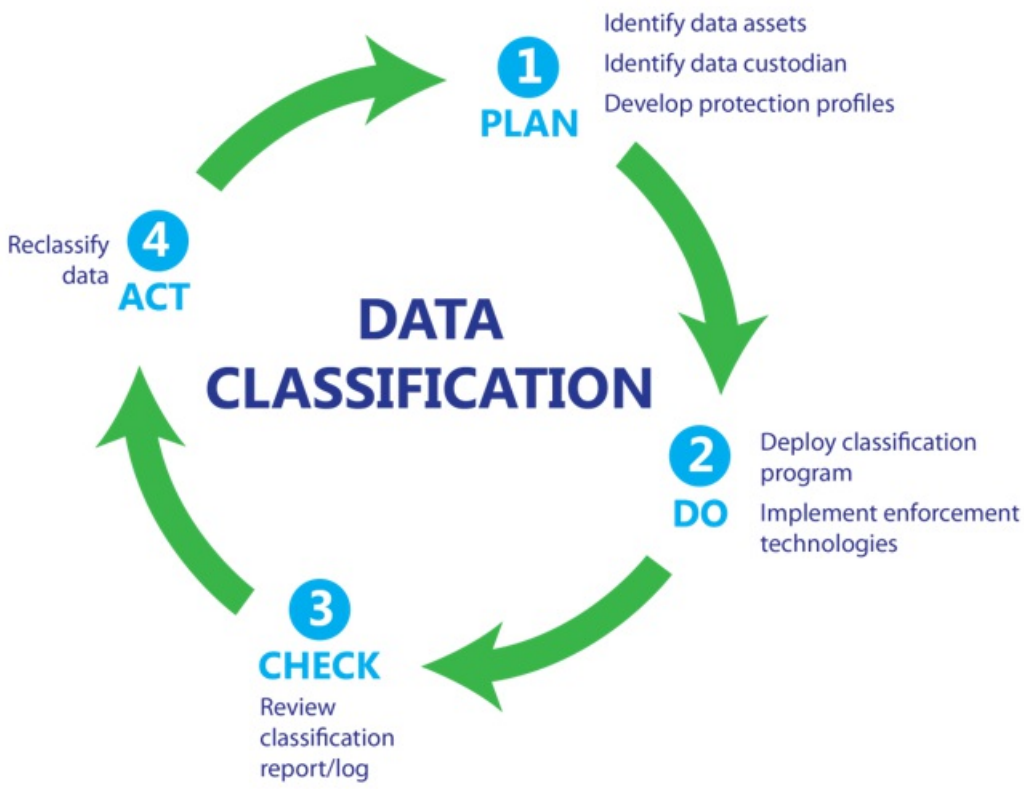
- **IaaS providers** requirements are limited to ensuring that the virtual environment can accommodate data classification capabilities and customer compliance requirements. IaaS providers have a smaller role in data classification because they only need to ensure that customer data addresses compliance requirements. However, providers must still ensure that their virtual environments address data classification requirements in addition to securing their data centers.
- **PaaS providers** responsibilities may be mixed, because the platform could be used in a layered approach to provide security for a classification tool. PaaS providers may be responsible for authentication and possibly some authorization rules, and must provide security and data classification capabilities to their application layer. Much like IaaS providers, PaaS providers need to ensure that their platform complies with any relevant data classification requirements.
- **SaaS providers** will frequently be considered as part of an authorization chain, and will need to ensure that the data stored in the SaaS application can be controlled by classification type. SaaS applications can be used for LOB applications, and by their very nature need to provide the means to authenticate and authorize data that is used and stored.

Classification process

Many organizations that understand the need for data classification and want to implement it face a basic challenge: where to begin?

One effective and simple way to implement data classification is to use the PLAN, DO, CHECK, ACT model from MOF. The following figure charts the tasks that are required to successfully implement data classification in this model.

1. **PLAN**. Identify data assets, a data custodian to deploy the classification program, and develop protection profiles.
2. **DO**. After data classification policies are agreed upon, deploy the program and implement enforcement technologies as needed for confidential data.
3. **CHECK**. Check and validate reports to ensure that the tools and methods being used are effectively addressing the classification policies.
4. **ACT**. Review the status of data access and review files and data that require revision using a reclassification and revision methodology to adopt changes and to address new risks.



Select a terminology model that addresses your needs

Several types of processes exist for classifying data, including manual processes, location-based processes that classify data based on a user's or system's location, application-based processes such as database-specific classification, and automated processes used by various technologies, some of which are described in the "Protecting confidential data" section later in this article.

This article introduces two generalized terminology models that are based on well-used and industry-respected models. These terminology models, both of which provide three levels of classification sensitivity, are shown in the following table.

> **NOTE**
> when classifying a file or resource that combines data that would typically be classified at differing levels, the highest level of classification present should establish the overall classification. For example, a file containing sensitive and restricted data should be classified as restricted.

SENSITIVITY	TERMINOLOGY MODEL 1	TERMINOLOGY MODEL 2
High	Confidential	Restricted

SENSITIVITY	TERMINOLOGY MODEL 1	TERMINOLOGY MODEL 2
Medium	For internal use only	Sensitive
Low	Public	Unrestricted

Confidential (restricted)

Information that is classified as confidential or restricted includes data that can be catastrophic to one or more individuals and/or organizations if compromised or lost. Such information is frequently provided on a "need to know" basis and might include:

- Personal data, including personally identifiable information such as Social Security or national identification numbers, passport numbers, credit card numbers, driver's license numbers, medical records, and health insurance policy ID numbers.
- Financial records, including financial account numbers such as checking or investment account numbers.
- Business material, such as documents or data that is unique or specific intellectual property.
- Legal data, including potential attorney-privileged material.
- Authentication data, including private cryptography keys, username password pairs, or other identification sequences such as private biometric key files.

Data that is classified as confidential frequently has regulatory and compliance requirements for data handling.

For internal use only (sensitive)

Information that is classified as being of medium sensitivity includes files and data that would not have a severe impact on an individual and/or organization if lost or destroyed. Such information might include:

- Email, most of which can be deleted or distributed without causing a crisis (excluding mailboxes or email from individuals who are identified in the confidential classification).
- Documents and files that do not include confidential data.

Generally, this classification includes anything that is not confidential. This classification can include most business data, because most files that are managed or used day-to-day can be classified as sensitive. With the exception of data that is made public or is confidential, all data within a business organization can be classified as sensitive by default.

Public (unrestricted)

Information that is classified as public includes data and files that are not critical to business needs or operations. This classification can also include data that has deliberately been released to the public for their use, such as marketing material or press announcements. In addition, this classification can include data such as spam email messages stored by an email service.

Define data ownership

It's important to establish a clear custodial chain of ownership for all data assets. The following table identifies different data ownership roles in data classification efforts and their respective rights.

ROLE	CREATE	MODIFY/DELETE	DELEGATE	READ	ARCHIVE/RESTORE
Owner	X	X	X	X	X
Custodian			X		
Administrator					X

ROLE	CREATE	MODIFY/DELETE	DELEGATE	READ	ARCHIVE/RESTORE
User*		X		X	

**Users may be granted additional rights such as edit and delete by a custodian*

> **NOTE**
> this table does not provide an exhaustive list of roles and rights, but merely a representative sample.

The **data asset owner** is the original creator of the data, who can delegate ownership and assign a custodian. When a file is created, the owner should be able to assign a classification, which means that they have a responsibility to understand what needs to be classified as confidential based on their organization's policies. All of a data asset owner's data can be auto-classified as for internal use only (sensitive) unless they are responsible for owning or creating confidential (restricted) data types. Frequently, the owner's role will change after the data is classified. For example, the owner might create a database of classified information and relinquish their rights to the data custodian.

> **NOTE**
> data asset owners often use a mixture of services, devices, and media, some of which are personal and some of which belong to the organization. A clear organizational policy can help ensure that usage of devices such as laptops and smart devices is in accordance with data classification guidelines.

The **data asset custodian** is assigned by the asset owner (or their delegate) to manage the asset according to agreements with the asset owner or in accordance with applicable policy requirements. Ideally, the custodian role can be implemented in an automated system. An asset custodian ensures that necessary access controls are provided and is responsible for managing and protecting assets delegated to their care. The responsibilities of the asset custodian could include:

- Protecting the asset in accordance with the asset owner's direction or in agreement with the asset owner
- Ensuring that classification policies are complied with
- Informing asset owners of any changes to agreed-upon controls and/or protection procedures prior to those changes taking effect
- Reporting to the asset owner about changes to or removal of the asset custodian's responsibilities
- An **administrator** represents a user who is responsible for ensuring that integrity is maintained, but they are not a data asset owner, custodian, or user. In fact, many administrator roles provide data container management services without having access to the data. The administrator role includes backup and restoration of the data, maintaining records of the assets, and choosing, acquiring, and operating the devices and storage that house the assets.
- The asset user includes anyone who is granted access to data or a file. Access assignment is often delegated by the owner to the asset custodian.

Implementation

Management considerations apply to all classification methodologies. These considerations need to include details about who, what, where, when, and why a data asset would be used, accessed, changed, or deleted. All asset management must be done with an understanding of how an organization views its risks, but a simple methodology can be applied as defined in the data classification process. Additional considerations for data classification include the introduction of new applications and tools, and managing change after a classification method is implemented.

Reclassification

Reclassifying or changing the classification state of a data asset needs to be done when a user or system determines that the data asset's importance or risk profile has changed. This effort is important for ensuring that the classification status continues to be current and valid. Most content that is not classified manually can be classified automatically or based on usage by a data custodian or data owner.

Manual data reclassification

Ideally, this effort would ensure that the details of a change are captured and audited. The most likely reason for manual reclassification would be for reasons of sensitivity, or for records kept in paper format, or a requirement to review data that was originally misclassified. Because this paper considers data classification and moving data to the cloud, manual reclassification efforts would require attention on a case-by-case basis and a risk management review would be ideal to address classification requirements. Generally, such an effort would consider the organization's policy about what needs to be classified, the default classification state (all data and files being sensitive but not confidential), and take exceptions for high-risk data.

Automatic data reclassification

Automatic data reclassification uses the same general rule as manual classification. The exception is that automated solutions can ensure that rules are followed and applied as needed. Data classification can be done as part of a data classification enforcement policy, which can be enforced when data is stored, in use, and in transit using authorization technology.

- Application-based. Using certain applications by default sets a classification level. For example, data from customer relationship management (CRM) software, HR, and health record management tools is confidential by default.
- Location-based. Data location can help identify data sensitivity. For example, data that is stored by an HR or financial department is more likely to be confidential in nature.

Data retention, recovery, and disposal

Data recovery and disposal, like data reclassification, is an essential aspect of managing data assets. The principles for data recovery and disposal would be defined by a data retention policy and enforced in the same manner as data reclassification; such an effort would be performed by the custodian and administrator roles as a collaborative task.

Failure to have a data retention policy could mean data loss or failure to comply with regulatory and legal discovery requirements. Most organizations that do not have a clearly defined data retention policy tend to use a default "keep everything" retention policy. However, such a retention policy has additional risks in cloud services scenarios.

For example, a data retention policy for cloud service providers can be considered as for "the duration of the subscription" (as long as the service is paid for, the data is retained). Such a pay-for-retention agreement may not address corporate or regulatory retention policies. Defining a policy for confidential data can ensure that data is stored and removed based on best practices. In addition, an archival policy can be created to formalize an understanding about what data should be disposed of and when.

Data retention policy should address the required regulatory and compliance requirements, as well as corporate legal retention requirements. Classified data might provoke questions about retention duration and exceptions for data that has been stored with a provider; such questions are more likely for data that has not been classified correctly.

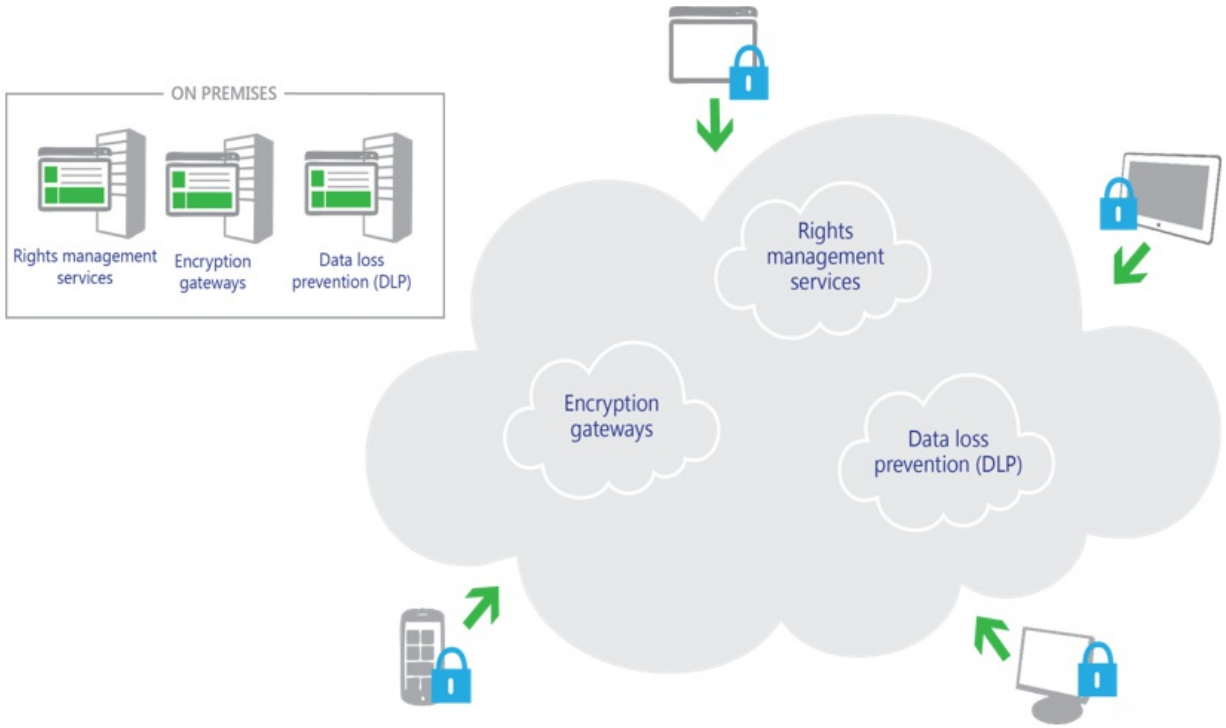
> **TIP**
> learn more about Azure Data Retention policies and more by reading the Microsoft Online Subscription Agreement

Protecting confidential data

After data is classified, finding and implementing ways to protect confidential data becomes an integral part of any data protection deployment strategy. Protecting confidential data requires additional attention to how data is stored and transmitted in conventional architectures as well as in the cloud.

This section provides basic information about some technologies that can automate enforcement efforts to help protect data that has been classified as confidential.

As the following figure shows, these technologies can be deployed as on-premises or cloud-based solutions—or in a hybrid fashion, with some of them deployed on-premises and some in the cloud. (Some technologies, such as encryption and rights management, also extend to user devices.)



Rights management software

One solution for preventing data loss is rights management software. Unlike approaches that attempt to interrupt the flow of information at exit points in an organization, rights management software works at deep levels within data storage technologies. Documents are encrypted, and control over who can decrypt them uses access controls that are defined in an authentication control solution such as a directory service.

> **TIP**
> you can use Azure Rights Management (Azure RMS) as the information protection solution to protect data in different scenarios. Read What is Azure Rights Management? for more information about this Azure solution.

Some of the benefits of rights management software include:

- Safeguarded sensitive information. Users can protect their data directly using rights management-enabled applications. No additional steps are required—authoring documents, sending email, and publishing data offer a consistent data protection experience.
- Protection travels with the data. Customers remain in control of who has access to their data, whether in the cloud, existing IT infrastructure, or at the user's desktop. Organizations can choose to encrypt their data and restrict access according to their business requirements.
- Default information protection policies. Administrators and users can use standard policies for many common

business scenarios, such as "Company Confidential–Read Only" and "Do Not Forward." A rich set of usage rights are supported such as read, copy, print, save, edit, and forward to allow flexibility in defining custom usage rights.

> **TIP**
> you can protect data in Azure Storage by using Azure Storage Service Encryption for Data at Rest. You can also use Azure Disk Encryption to help protect data contained on virtual disks used for Azure Virtual Machines.

Encryption gateways

Encryption gateways operate in their own layers to provide encryption services by rerouting all access to cloud-based data. This approach should not be confused with that of a virtual private network (VPN). Encryption gateways are designed to provide a transparent layer to cloud-based solutions.

Encryption gateways can provide a means to manage and secure data that has been classified as confidential by encrypting the data in transit as well as data at rest.

Encryption gateways are placed into the data flow between user devices and application data centers to provide encryption/decryption services. These solutions, like VPNs, are predominantly on-premises solutions. They are designed to provide a third party with control over encryption keys, which helps reduce the risk of placing both the data and key management with one provider. Such solutions are designed, much like encryption, to work seamlessly and transparently between users and the service.

> **TIP**
> you can use Azure ExpressRoute to extend your on-premises networks into the Microsoft cloud over a dedicated private connection. Read ExpressRoute technical overview for more information about this capability. Another options for cross premises connectivity between your on-premises network and Azure is a site-to-site VPN.

Data loss prevention

Data loss (sometimes referred to as data leakage) is an important consideration, and the prevention of external data loss via malicious and accidental insiders is paramount for many organizations.

Data loss prevention (DLP) technologies can help ensure that solutions such as email services do not transmit data that has been classified as confidential. Organizations can take advantage of DLP features in existing products to help prevent data loss. Such features use policies that can be easily created from scratch or by using a template supplied by the software provider.

DLP technologies can perform deep content analysis through keyword matches, dictionary matches, regular expression evaluation, and other content examination to detect content that violates organizational DLP policies. For example, DLP can help prevent the loss of the following types of data:

- Social Security and national identification numbers
- Banking information
- Credit card numbers
- IP addresses

Some DLP technologies also provide the ability to override the DLP configuration (for example, if an organization needs to transmit Social Security number information to a payroll processor). In addition, it's possible to configure DLP so that users are notified before they even attempt to send sensitive information that should not be transmitted.

TIP

you can use Office 365 DLP capabilities to protect your documents. Read Office 365 compliance controls: Data Loss Prevention for more information.

See also

- Azure Data Encryption Best Practices
- Azure Identity Management and access control security best practices
- Azure Security Team Blog
- Microsoft Security Response Center

Application architecture on Azure

11/22/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • TerryLanfear • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • curtand

---

To help secure your cloud-based solutions on Microsoft Azure, a strong architectural foundation is critical. Architects, designers, and implementers all benefit from a strong knowledge of application and services architecture. This foundational knowledge helps you understand all the components of your cloud-based solutions and make it easier to integrate security into all aspects of your design and implementation.

We have the following to help you with your architectural investigations and designs:

- Architectural infographics
- Architectural blueprints
- Cloud and enterprise symbol set
- 3D blueprint Visio template

Architectural infographics

Microsoft publishes several architectural related posters/infographics. They include:

- Building Real-World Cloud Applications
- Scaling with Cloud Services

Architectural blueprints

Microsoft publishes a set of high-level architectural blueprints showing how to build specific types of systems using Microsoft products. Each blueprint includes a:

- Flat 2D Visio 2003-based file that you can download and modify
- Colorful 3D perspective PDF file to introduce the blueprint to less technical audiences
- Video that walks through the 3D version

Cloud and enterprise symbol set

View the Visio and symbols training video and then download the Cloud and Enterprise Symbol set to help create technical materials that describe Azure, Windows Server, SQL Server and more. You can use the symbols in architecture diagrams, training materials, presentations, datasheets, infographics, whitepapers, and even third party books if the book trains people to use Microsoft products. However, they are not meant for use in user interfaces.

3D blueprint Visio template

The 3D versions of the Microsoft Architecture Blueprints were initially created in a non-Microsoft tool. A new Visio 2013 (and later) template shipped on August 5, 2015 as part of a Microsoft Architecture certification course distributed on EDX.ORG.

The template is also available outside the course.

- View the training video first so you know what it can do

- Download the Microsoft 3d Blueprint Visio Template
- Download the Cloud and Enterprise Symbols to use with the 3D template

Azure security best practices and patterns

11/15/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Matthew Baldwin • TerryLanfear

---

We currently have the following Azure security best practices and patterns articles. Make sure to visit this site periodically to see updates to our growing list of Azure security best practices and patterns:

- Azure network security best practices
- Azure data security and encryption best practices
- Identity management and access control security best practices
- Internet of Things security best practices
- Best practices for software update on Microsoft Azure IaaS
- Azure boundary security best practices
- Implementing a secure hybrid network architecture in Azure

Azure provides a secure platform on which you can build your solutions. We also provide services and technologies to make your solutions on Azure more secure. Because of the many options available to you, many of you have voiced an interest in what Microsoft recommends as best practices and patterns for improving security.

We understand your interest and have created a collection of documents that describe things you can do, given the right context, to improve the security of Azure deployments.

In these best practices and patterns articles, we discuss a collection of best practices and useful patterns for specific topics. These best practices and patterns are derived from our experiences with these technologies and the experiences of customers like yourself.

For each best practice we strive to 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

We look forward to including many more articles on Azure security architecture and best practices. If there are topics that you'd like us to include, let us know in the discussion area at the bottom of this page.

Disaster recovery and high availability for applications built on Microsoft Azure

11/15/2016 • 12 min to read • Edit on GitHub

Contributors

adamglick • Ralph Squillace • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • James Dunn • Simon Rolfe

Introduction

This article focuses on high availability for applications running in Azure. An overall strategy for high availability also includes the aspect of disaster recovery. Planning for failures and disasters in the cloud requires you to recognize the failures quickly. You then implement a strategy that matches your tolerance for the application's downtime. Additionally, you have to consider the extent of data loss the application can tolerate without causing adverse business consequences as it is restored.

Most companies say they are prepared for temporary and large-scale failures. However, before you answer that question for yourself, does your company rehearse these failures? Do you test the recovery of databases to ensure you have the correct processes in place? Probably not. That's because successful disaster recovery starts with lots of planning and architecting to implement these processes. Just like many other non-functional requirements, such as security, disaster recovery rarely gets the up-front analysis and time allocation it requires. Also, most companies don't have the budget for geographically distributed regions with redundant capacity. Consequently, even mission critical applications are frequently excluded from proper disaster recovery planning.

Cloud platforms, such as Azure, provide geographically dispersed regions around the world. These platforms also provide capabilities that support availability and a variety of disaster recovery scenarios. Now, every mission critical cloud application can be given due consideration for disaster proofing of the system. Azure has resiliency and disaster recovery built in to many of its services. You must study these platform features carefully, and supplement with application strategies.

This article outlines the necessary architecture steps you must take to disaster-proof an Azure deployment. Then you can implement the larger business continuity process. A business continuity plan is a roadmap for continuing operations under adverse conditions. This could be a failure with technology, such as a downed service, or a natural disaster, such as a storm or power outage. Application resiliency for disasters is only a subset of the larger disaster recovery process, as described in this NIST document: Contingency Planning Guide for Information Technology Systems.

The following sections define different levels of failures, techniques to deal with them, and architectures that support these techniques. This information provides input to your disaster recovery processes and procedures, to ensure your disaster recovery strategy works correctly and efficiently.

Characteristics of resilient cloud applications

A well architected application can withstand capability failures at a tactical level, and it can also tolerate strategic system-wide failures at the region level. The following sections define the terminology referenced throughout the document to describe various aspects of resilient cloud services.

High availability

A highly available cloud application implements strategies to absorb the outage of dependencies, like the managed services offered by the cloud platform. Despite possible failures of the cloud platform's capabilities, this approach

permits the application to continue to exhibit the expected functional and non-functional systemic characteristics. This is covered in-depth in the Channel 9 video series, Failsafe: Guidance for Resilient Cloud Architectures.

When you implement the application, you must consider the probability of a capability outage. Additionally, consider the impact an outage will have on the application from the business perspective, before diving deep into the implementation strategies. Without due consideration to the business impact and the probability of hitting the risk condition, the implementation can be expensive and potentially unnecessary.

Consider an automotive analogy for high availability. Even quality parts and superior engineering does not prevent occasional failures. For example, when your car gets a flat tire, the car still runs, but it is operating with degraded functionality. If you planned for this potential occurrence, you can use one of those thin-rimmed spare tires until you reach a repair shop. Although the spare tire does not permit fast speeds, you can still operate the vehicle until you replace the tire. Similarly, a cloud service that plans for potential loss of capabilities can prevent a relatively minor problem from bringing down the entire application. This is true even if the cloud service must run with degraded functionality.

There are a few key characteristics of highly available cloud services: availability, scalability, and fault tolerance. Although these characteristics are interrelated, it is important to understand each, and how they contribute to the overall availability of the solution.

Availability

An available application considers the availability of its underlying infrastructure and dependent services. Available applications remove single points of failure through redundancy and resilient design. When you broaden the scope to consider availability in Azure, it is important to understand the concept of the effective availability of the platform. Effective availability considers the Service Level Agreements (SLA) of each dependent service, and their cumulative effect on the total system availability.

System availability is the measure of the percentage of a time window the system will be able to operate. For example, the availability SLA of at least two instances of a web or worker role in Azure is 99.95 percent (out of 100 percent). It does not measure the performance or functionality of the services running on those roles. However, the effective availability of your cloud service is also affected by the various SLAs of the other dependent services. The more moving parts within the system, the more care you must take to ensure the application can resiliently meet the availability requirements of its end users.

Consider the following SLAs for an Azure service that uses Azure services: Compute, Azure SQL Database, and Azure Storage.

AZURE SERVICE	SLA	POTENTIAL MINUTES DOWNTIME/MONTH (30 DAYS)
Compute	99.95%	21.6 minutes
SQL Database	99.99%	4.3 minutes
Storage	99.90%	43.2 minutes

You must plan for all services to potentially go down at different times. In this simplified example, the total number of minutes per month that the application could be down is 108 minutes. A 30-day month has a total of 43,200 minutes. 108 minutes is .25 percent of the total number of minutes in a 30-day month (43,200 minutes). This gives you an effective availability of 99.75 percent for the cloud service.

However, using availability techniques described in this paper can improve this. For example, if you design your application to continue running when the SQL Database is unavailable, you can remove that from the equation. This might mean that the application runs with reduced capabilities, so there are also business requirements to consider. For a complete list of Azure SLAs, see Service Level Agreements.

Scalability

Scalability directly affects availability. An application that fails under increased load is no longer available. Scalable applications are able to meet increased demand with consistent results, in acceptable time windows. When a system is scalable, it scales horizontally or vertically to manage increases in load while maintaining consistent performance. In basic terms, horizontal scaling adds more machines of the same size (processor, memory, and bandwidth), while vertical scaling increases the size of the existing machines. For Azure, you have vertical scaling options for selecting various machine sizes for compute. However, changing the machine size requires a re-deployment. Therefore, the most flexible solutions are designed for horizontal scaling. This is especially true for compute, because you can easily increase the number of running instances of any web or worker role. These additional instances handle increased traffic through the Azure Web portal, PowerShell scripts, or code. Base this decision on increases in specific monitored metrics. In this scenario, user performance or metrics do not suffer a noticeable drop under load. Typically, the web and worker roles store any state externally. This allows for flexible load balancing and graceful handling of any changes to instance counts. Horizontal scaling also works well with services, such as Azure Storage, which do not provide tiered options for vertical scaling.

Cloud deployments should be seen as a collection of scale-units. This allows the application to be elastic in servicing the throughput needs of end users. The scale units are easier to visualize at the web and application server level. This is because Azure already provides stateless compute nodes through web and worker roles. Adding more compute scale-units to the deployment will not cause any application state management side effects, because compute scale-units are stateless. A storage scale-unit is responsible for managing a partition of data (structured or unstructured). Examples of storage scale-units include Azure Table partition, Azure Blob container, and Azure SQL Database. Even the usage of multiple Azure Storage accounts has a direct impact on the application scalability. You must design a highly scalable cloud service to incorporate multiple storage scale-units. For instance, if an application uses relational data, partition the data across several SQL databases. Doing so allows the storage to keep up with the elastic compute scale-unit model. Similarly, Azure Storage allows data partitioning schemes that require deliberate designs to meet the throughput needs of the compute layer. For a list of best practices for designing scalable cloud services, see Best Practices for the Design of Large-Scale Services on Azure Cloud Services.

Fault tolerance

Applications should assume that every dependent cloud capability can and will go down at some point in time. A fault tolerant application detects and maneuvers around failed elements, to continue and return the correct results within a specific timeframe. For transient error conditions, a fault tolerant system will employ a retry policy. For more serious faults, the application can detect problems and fail over to alternative hardware or contingency plans until the failure is corrected. A reliable application can properly manage the failure of one or more parts, and continue operating properly. Fault tolerant applications can use one or more design strategies, such as redundancy, replication, or degraded functionality.

Disaster recovery

A cloud deployment might cease to function due to a systemic outage of the dependent services or the underlying infrastructure. Under such conditions, a business continuity plan triggers the disaster recovery process. This process typically involves both operations personnel and automated procedures in order to reactivate the application in an available region. This requires the transfer of application users, data, and services to the new region. It also involves the use of backup media or ongoing replication.

Consider the previous analogy that compared high availability to the ability to recover from a flat tire through the use of a spare. In contrast, disaster recovery involves the steps taken after a car crash, where the car is no longer operational. In that case, the best solution is to find an efficient way to change cars, by calling a travel service or a friend. In this scenario, there is likely going to be a longer delay in getting back on the road. There is also more complexity in repairing and returning to the original vehicle. In the same way, disaster recovery to another region is a complex task that typically involves some downtime and potential loss of data. To better understand and evaluate disaster recovery strategies, it is important to define two terms: recovery time objective (RTO) and recovery point objective (RPO).

Recovery time objective

The RTO is the maximum amount of time allocated for restoring application functionality. This is based on business requirements, and it is related to the importance of the application. Critical business applications require a low RTO.

Recovery point objective

The RPO is the acceptable time window of lost data due to the recovery process. For example, if the RPO is one hour, you must completely back up or replicate the data at least every hour. Once you bring up the application in an alternate region, the backup data may be missing up to an hour of data. Like RTO, critical applications target a much smaller RPO.

Checklist

Let's summarize the key points that have been covered in this article (and its related articles on high availability and disaster recovery for Azure applications). This summary will act as a checklist of items you should consider for your own availability and disaster recovery planning. These are best practices that have been useful for customers seeking to get serious about implementing a successful solution.

1. Conduct a risk assessment for each application, because each can have different requirements. Some applications are more critical than others and would justify the extra cost to architect them for disaster recovery.
2. Use this information to define the RTO and RPO for each application.
3. Design for failure, starting with the application architecture.
4. Implement best practices for high availability, while balancing cost, complexity, and risk.
5. Implement disaster recovery plans and processes.
   - Consider failures that span the module level all the way to a complete cloud outage.
   - Establish backup strategies for all reference and transactional data.
   - Choose a multi-site disaster recovery architecture.
6. Define a specific owner for disaster recovery processes, automation, and testing. The owner should manage and own the entire process.
7. Document the processes so they are easily repeatable. Although there is one owner, multiple people should be able to understand and follow the processes in an emergency.
8. Train the staff to implement the process.
9. Use regular disaster simulations for both training and validation of the process.

Summary

When hardware or applications fail within Azure, the techniques and strategies for managing them are different than when failure occurs on on-premises systems. The main reason for this is that cloud solutions typically have more dependencies on infrastructure that is distributed across an Azure region, and managed as separate services. You must deal with partial failures using high availability techniques. To manage more severe failures, possibly due to a disaster event, use disaster recovery strategies.

Azure detects and handles many failures, but there are many types of failures that require application-specific strategies. You must actively prepare for and manage the failures of applications, services, and data.

When creating your application's availability and disaster recovery plan, consider the business consequences of the application's failure. Defining the processes, policies, and procedures to restore critical systems after a catastrophic event takes time, planning, and commitment. And once you establish the plans, you cannot stop there. You must regularly analyze, test, and continually improve the plans based on your application portfolio, business needs, and the technologies available to you. Azure provides new capabilities and raises new challenges to creating robust applications that withstand failures.

Additional resources

High availability for applications built on Microsoft Azure

Disaster recovery for applications built on Microsoft Azure

Azure resiliency technical guidance

Overview: Cloud business continuity and database disaster recovery with SQL Database

High availability and disaster recovery for SQL Server in Azure Virtual Machines

Failsafe: Guidance for resilient cloud architectures

Best Practices for the design of large-scale services on Azure Cloud Services

Next steps

This article is part of a series of articles focused on disaster recovery and high availability for Azure applications. The next article in this series is High availability for applications built on Microsoft Azure.

Microsoft Trust Center

11/22/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil

---

The Azure Security Information site on Azure.com gives you the information you need to plan, design, deploy, configure, and manage your cloud solutions securely. With the Microsoft Trust center, you also have the information you need to be confident that the Azure platform on which you run your services is secure.

We know that when you entrust your applications and data to Azure, you're going to have questions. Where is it? Who can access it? What is Microsoft doing to protect it? How can you verify that Microsoft is doing what it says?

And we have answers. Because it's your data, you decide who has access, and you work with us to decide where it is located. To safeguard your data, we use state-of-the-art security technology and world-class cryptography. Our compliance is independently audited, and we're transparent on many levels—from how we handle legal demands for your customer data to the security of our code.

Here's what you find at the Microsoft Trust Center:

- Security – Learn how all the Microsoft Cloud services are secured.
- Privacy – Understand how Microsoft ensures privacy of your Data in the Microsoft cloud.
- Compliance – Discover how Microsoft helps organizations comply with national, regional, and industry-specific requirements governing the collection and use of individuals' data.
- Transparency – View how Microsoft believes that you control your data in the cloud and how Microsoft helps you know as much as possible about how that data is handled.
- Products and Services – See all the Microsoft Cloud products and services in one place
- Service Trust Portal – Obtain copies of independent audit reports of Microsoft cloud services, risk assessments, security best practices, and related materials.
- What's New – Find out what's new in Microsoft Cloud Trust
- Resources – Investigate white papers, videos, and case studies on Microsoft Trusted Cloud

The Microsoft Trust Center has what you need to understand what we do to secure the Microsoft Cloud.

Microsoft Security Response Center

11/15/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes

---

The Microsoft Security Response Center (MSRC) is led by some of the world's most experienced security experts. These experts identify, monitor, respond to and resolve security incidents and on-premises and cloud vulnerabilities around the clock, each day of the year.

In addition to the continuous work the MSRC does in the background, the MSRC team has a number of resources available to you so that you can understand how to secure your Azure assets and deployments more effectively.

The MSRC Blog

The MSRC blog is the place to go to get the latest news on what the MSRC is doing to help protect you against cloud threats.

White Papers

The MSRC has published a number of white papers that will help you understand what they do and how they do it. Some provide insights into how we secure the Microsoft cloud and include useful information on how you can employ the same security configurations.

Security Researcher Engagement and Bounty Programs

The MSRC supports collaboration and relationships with security researchers globally to advance Microsoft product security.

Microsoft bounty programs pay researchers for novel exploitation techniques, defensive ideas that mitigate novel exploitations, and identification of critical vulnerabilities in Microsoft on-premises and cloud software.

Learn more about these programs at the MSRC Bug Bounty page and the MSRC blog.

To learn more about the MSRC, please visit the MSRC home page.

Pen Testing

11/15/2016 • 2 min to read • Edit on GitHub

Contributors

Yuri Diogenes • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • TerryLanfear

---

One of the great things about using Microsoft Azure for application testing and deployment is that you don't need to put together an on-premises infrastructure to develop, test and deploy your applications. All the infrastructure is taken care of by the Microsoft Azure platform services. 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 us improve our platform and guides our actions in terms of improving security controls, introducing new security controls, and improving our security processes.

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.

When you pen test your applications, it might look like an attack to us. We continuously monitor for attack patterns and will initiate an incident response process if we need to. It doesn't help you and it doesn't help us if we trigger an incident response due to your own due diligence pen testing.

What to do?

When you're ready to pen test your Azure-hosted applications, you need to let us know. Once we know that you're going to be performing specific tests, we won't inadvertently shut you down (such as blocking the IP address that you're testing from), as long as your tests conform to the Azure pen testing terms and conditions. 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.

Are you ready to get started with pen testing your applications hosted in Microsoft Azure? If so, then head on over to the Penetration Test Overview page (and click the Create a Testing Request button at the bottom of the page. You'll also find more information on the pen testing terms and conditions and helpful links on how you can report security flaws related to Azure or any other Microsoft service.

Introduction to Azure Security Center

11/22/2016 • 5 min to read • Edit on GitHub

Contributors

TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • 4c74356b41 • Rebecca Agiewich

Learn about Azure Security Center, its key capabilities, and how it works.

> **NOTE**
> This document introduces the service by using an example deployment.

What is 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.

Key capabilities

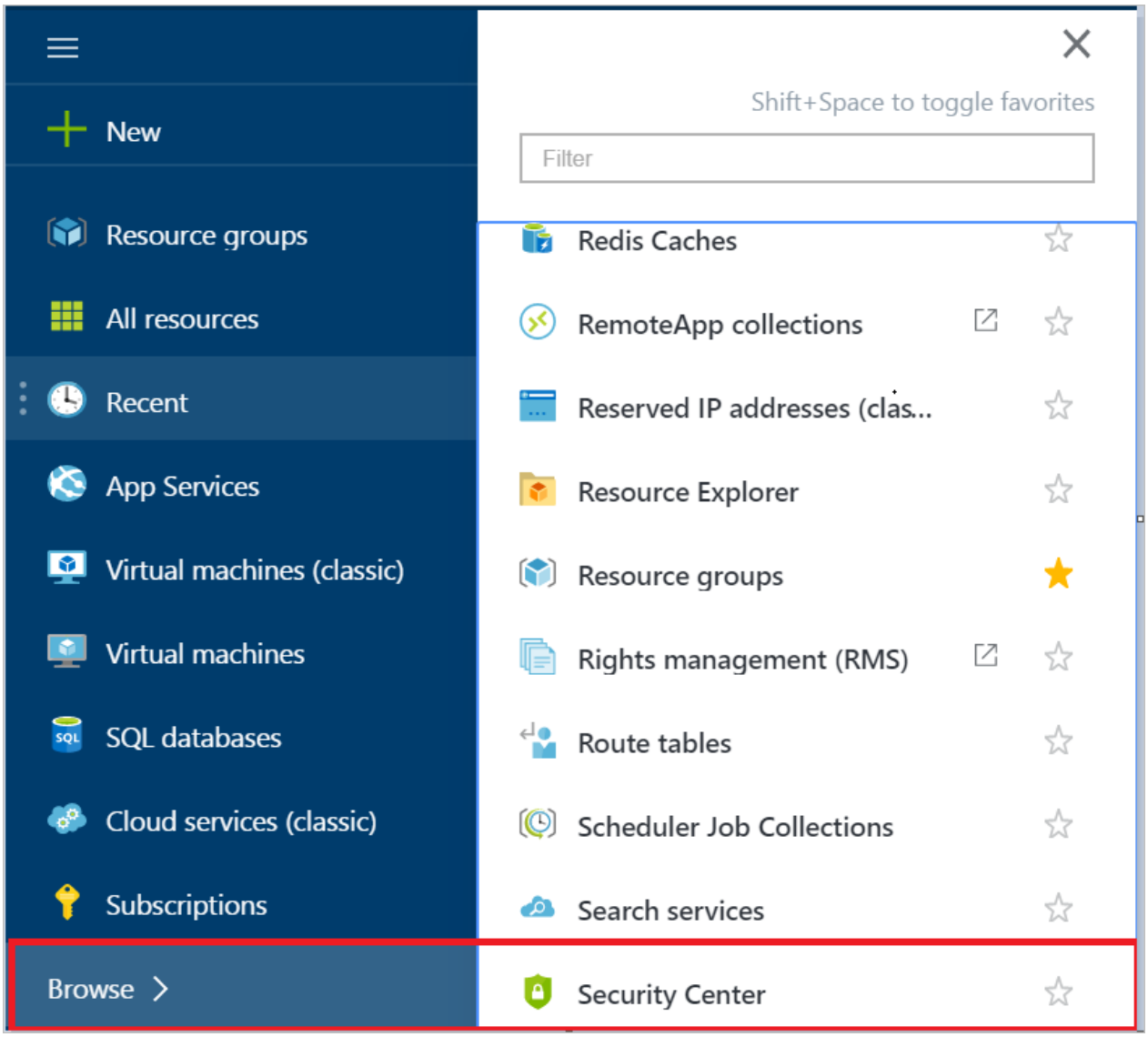
Security Center delivers easy-to-use and effective threat prevention, detection, and response capabilities that are built in to Azure. Key capabilities are:

STAGE	CAPABILITY
Prevent	Monitors the security state of your Azure resources
Prevent	Defines policies for your Azure subscriptions and resource groups based on your company's security requirements, the types of applications that you use, and the sensitivity of your data
Prevent	Uses policy-driven security recommendations to guide service owners through the process of implementing needed controls
Prevent	Rapidly deploys security services and appliances from Microsoft and partners
Detect	Automatically collects and analyzes security data from your Azure resources, the network, and partner solutions like antimalware programs and firewalls
Detect	Leverages global threat intelligence from Microsoft products and services, the Microsoft Digital Crimes Unit (DCU), the Microsoft Security Response Center (MSRC), and external feeds

STAGE	CAPABILITY
Detect	Applies advanced analytics, including machine learning and behavioral analysis
Respond	Provides prioritized security incidents/alerts
Respond	Offers insights into the source of the attack and impacted resources
Respond	Suggests ways to stop the current attack and help prevent future attacks

Introductory walkthrough

You access Security Center from the Azure portal. Sign in to the portal, select **Browse**, and then scroll to the **Security Center** option or select the **Security Center** tile that you previously pinned to the portal dashboard.



From Security Center, you can set security policies, monitor security configurations, and view security alerts.

Security policies

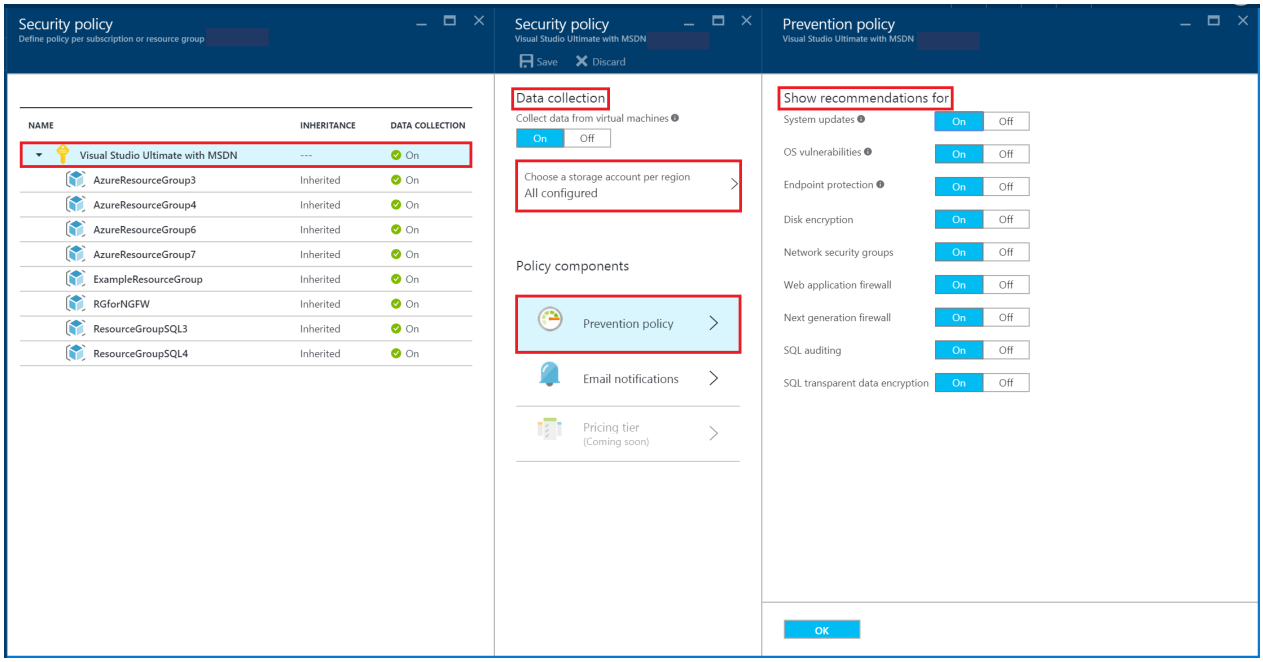
You can define policies for your Azure subscriptions and resource groups according to your company's security requirements. You can also tailor them to the types of applications you're using or to the sensitivity of the data in each subscription. For example, resources used for development or testing may have different security

requirements than those used for production applications. Likewise, applications with regulated data like PII may require a higher level of security.

> **NOTE**
> To modify a security policy at the subscription level or the resource group level, you must be the Owner of the subscription or a Contributor to it.

On the **Security Center** blade, select the **Policy** tile for a list of your subscriptions and resource groups.

On the **Security policy** blade, select a subscription to view the policy details.



Data collection (see above) enables data collection for a security policy. Enabling provides:

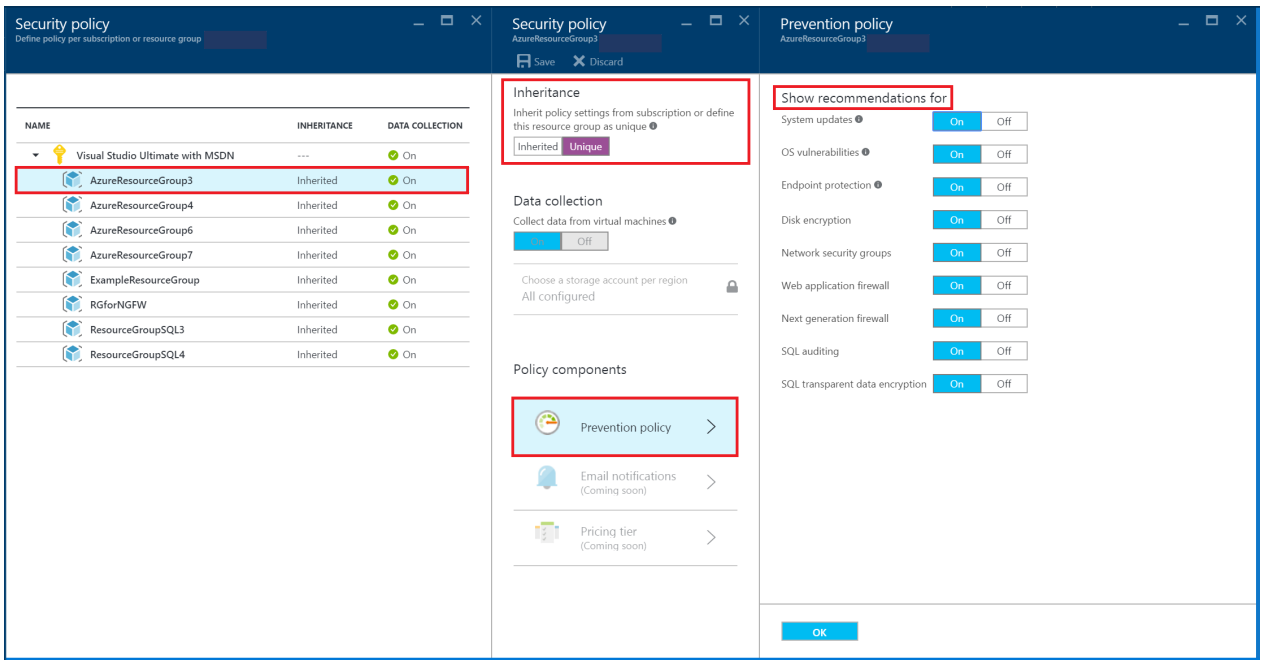
- Daily scanning of all supported virtual machines (VMs) for security monitoring and recommendations.
- Collection of security events for analysis and threat detection.

Choose a storage account per region (see above) lets you choose, for each region in which you have VMs running, the storage account where data collected from those VMs is stored. If you do not choose a storage account for each region, it is created for you. The data that's collected is logically isolated from other customers' data for security reasons.

> **NOTE**
> Data collection and choosing a storage account per region is configured at the subscription level.

Select **Prevention policy** (see above) to open the **Prevention policy** blade. **Show recommendations for** lets you choose the security controls that you want to monitor and recommend based on the security needs of the resources within the subscription.

Next, select a resource group to view policy details.



Inheritance (see above) lets you define the resource group as:

- Inherited (default) which means all security policies for this resource group are inherited from the subscription level.
- Unique which means the resource group has a custom security policy. You need to make changes under **Show recommendations for**.

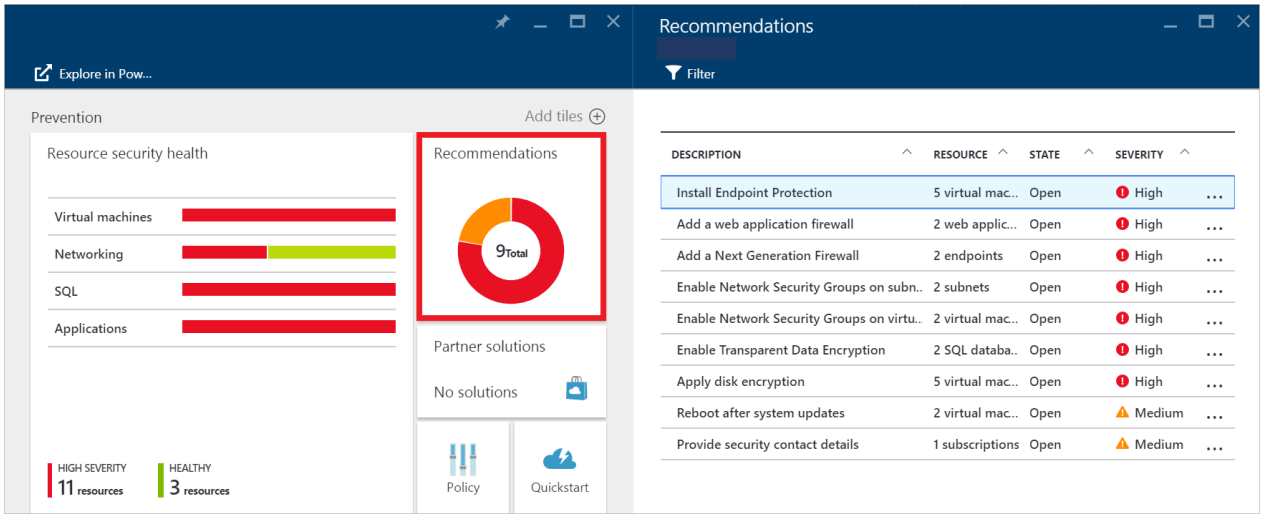
> **NOTE**
> If there is a conflict between subscription level policy and resource group level policy, the resource group level policy takes precedence.

Security recommendations

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

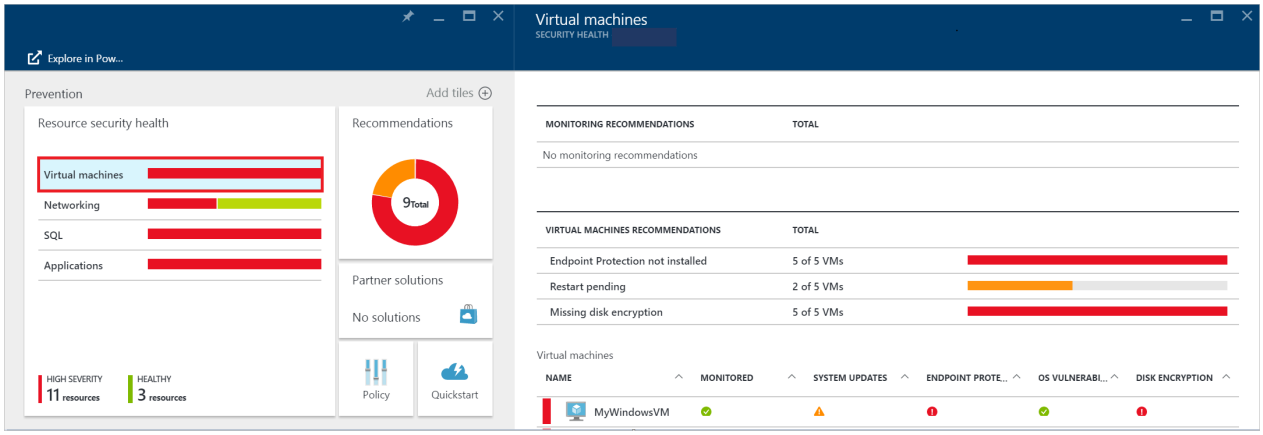
Click the **Recommendations** tile for a list of recommendations. Click each recommendation to view additional information or to take action to resolve the issue.



Resource health

The **Resource security health** tile shows the overall security posture of the environment by resource type, including VMs, web applications, and other resources.

Select a resource type on the **Resource security health** tile to view more information, including a list of any potential security vulnerabilities that have been identified. (**Virtual machines** is selected in the example below.)

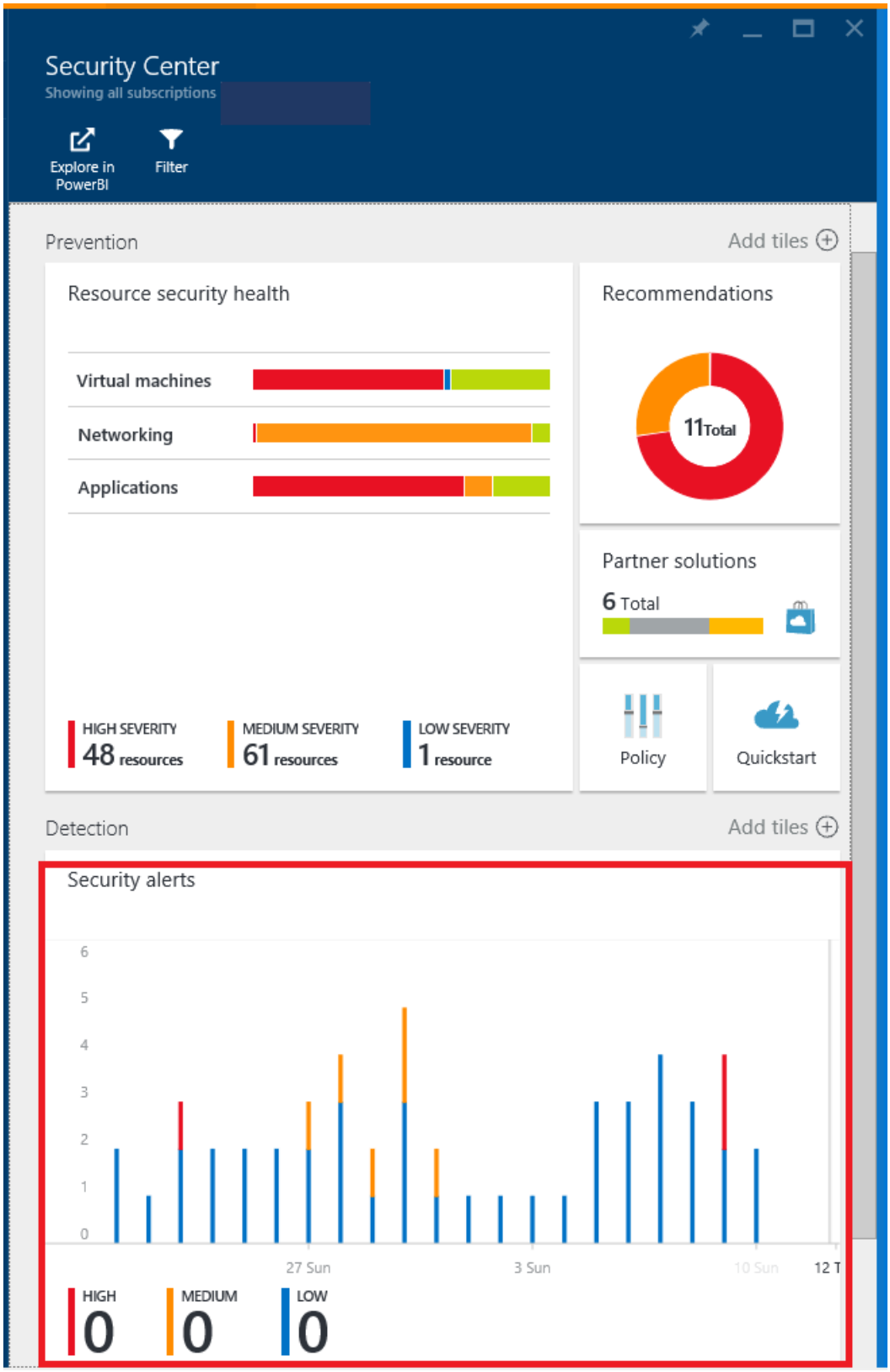


Security alerts

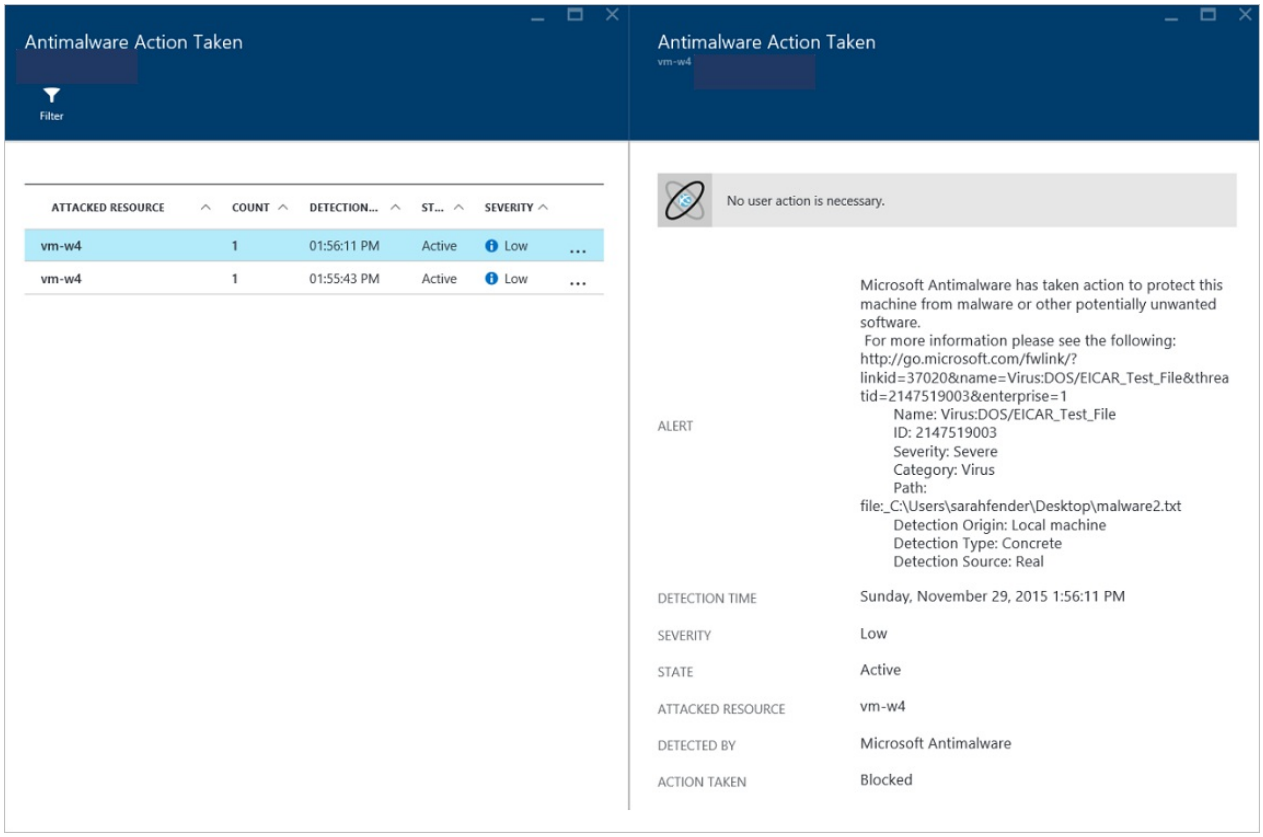
Security Center automatically collects, analyzes, and integrates log data from your Azure resources, the network, and partner solutions like antimalware programs and firewalls. When threats are detected, a security alert is created. Examples include detection of:

- 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

Clicking the **Security alerts** tile displays a list of prioritized alerts.



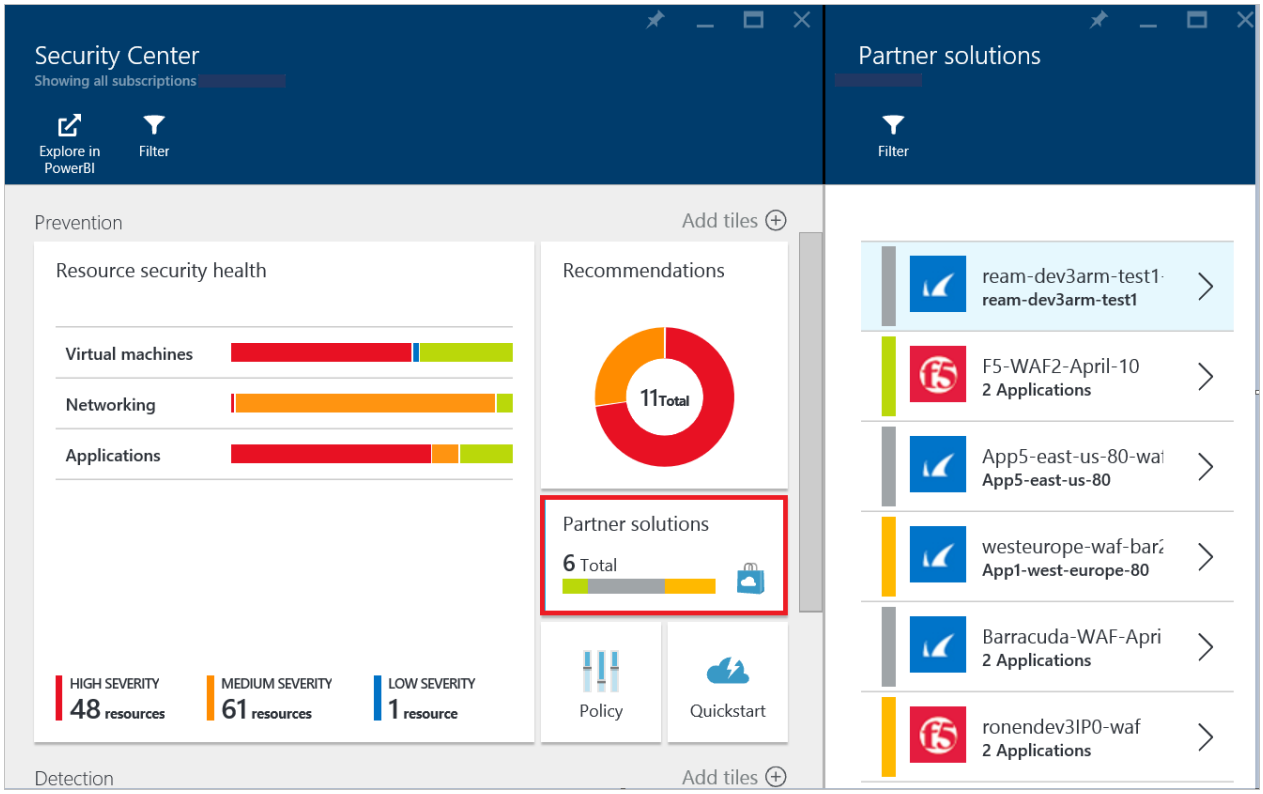
Selecting an alert shows more information about the attack and suggestions for how to remediate it.



Partner solutions

The **Partner solutions** tile lets you monitor at a glance the health status of your partner solutions integrated with your Azure subscription. Security Center displays alerts coming from the solutions.

Select the **Partner solutions** tile. A blade opens displaying a list of all connected partner solutions.



Get started

To get started with Security Center, you need a subscription to Microsoft Azure. Security Center is enabled with your Azure subscription. If you do not have a subscription, you can sign up for a free trial.

You access Security Center from the Azure portal. See the portal documentation to learn more.

Getting started with Azure Security Center quickly guides you through the security-monitoring and policy-management components of Security Center.

See also

In this document, you were introduced to Security Center, its key capabilities, and how to get started. To learn more, see the following:

- Setting security policies in Azure Security Center — Learn how to configure security policies for your Azure subscriptions and resource groups.
- Managing security recommendations in Azure Security Center — Learn how recommendations help you protect your Azure resources.
- Security health monitoring in Azure Security Center — Learn how to monitor the health of your Azure resources.
- Managing and responding to security alerts in Azure Security Center — Learn how to manage and respond to security alerts.
- Monitoring partner solutions with Azure Security Center — Learn how to monitor the health status of your partner solutions.
- Azure Security Center FAQ — Find frequently asked questions about using the service.
- Azure Security blog — Get the latest Azure security news and information.

What is Azure Key Vault?

11/15/2016 • 3 min to read • Edit on GitHub

Contributors

cabailey • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Matthew Baldwin • Dene Hager

Azure Key Vault is available in most regions. For more information, see the Key Vault pricing page.

Introduction

Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. By using 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. If you choose to do this, Microsoft processes your keys in FIPS 140-2 Level 2 validated HSMs (hardware and firmware).

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 seamlessly migrate them to production keys. Security administrators can grant (and revoke) permission to keys, as needed.

Use the following table to better understand how Key Vault can help to meet the needs of developers and security administrators.

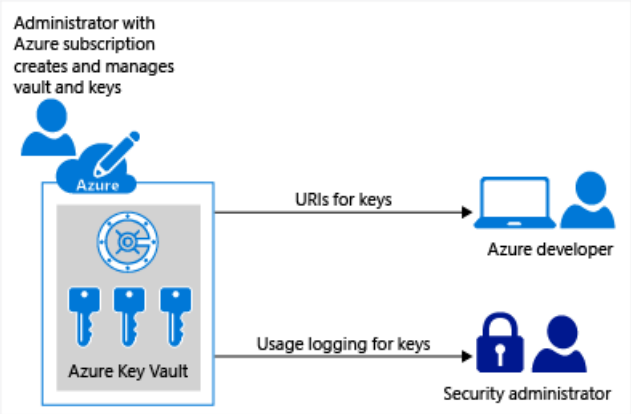
ROLE	PROBLEM STATEMENT	SOLVED BY AZURE KEY VAULT
Developer for an Azure application	"I want to write an application for Azure that uses keys for signing and encryption, but I want these keys to be external from my application so that the solution is suitable for an application that is geographically distributed. I also want these keys and secrets to be protected, without having to write the code myself. I also want these keys and secrets to be easy for me to use from my applications, with optimal performance."	√ Keys are stored in a vault and invoked by URI when needed. √ Keys are safeguarded by Azure, using industry-standard algorithms, key lengths, and hardware security modules (HSMs). √ Keys are processed in HSMs that reside in the same Azure datacenters as the applications. This provides better reliability and reduced latency than if the keys reside in a separate location, such as on-premises.
Developer for Software as a Service (SaaS)	"I don't want the responsibility or potential liability for my customers' tenant keys and secrets. I want the customers to own and manage their keys so that I can concentrate on doing what I do best, which is providing the core software features."	√ Customers can import their own keys into Azure, and manage them. When a SaaS application needs to perform cryptographic operations by using their customers' keys, Key Vault does these operations on behalf of the application. The application does not see the customers' keys.

ROLE	PROBLEM STATEMENT	SOLVED BY AZURE KEY VAULT
Chief security officer (CSO)	"I want to know that our applications comply with FIPS 140-2 Level 2 HSMs for secure key management. I want to make sure that my organization is in control of the key life cycle and can monitor key usage. And although we use multiple Azure services and resources, I want to manage the keys from a single location in Azure."	√ HSMs are FIPS 140-2 Level 2 validated. √ Key Vault is designed so that Microsoft does not see or extract your keys. √ Near real-time logging of key usage. √ The vault provides a single interface, regardless of how many vaults you have in Azure, which regions they support, and which applications use them.

Anybody with an Azure subscription can create and use key vaults. Although Key Vault benefits developers and security administrators, it could be implemented and managed by an organization's administrator who manages other Azure services for an organization. For example, this administrator would sign in with an Azure subscription, create a vault for the organization in which to store keys, and then be responsible for operational tasks, such as:

- Create or import a key or secret
- Revoke or delete a key or secret
- Authorize users or applications to access the key vault, so they can then manage or use its keys and secrets
- Configure key usage (for example, sign or encrypt)
- Monitor key usage

This administrator would then provide developers with URIs to call from their applications, and provide their security administrator with key usage logging information.



Developers can also manage the keys directly, by using APIs. For more information, see the Key Vault developer's guide.

Next Steps

For a getting started tutorial for an administrator, see Get Started with Azure Key Vault.

For more information about usage logging for Key Vault, see Azure Key Vault Logging.

For more information about using keys and secrets with Azure Key Vault, see About Keys, Secrets, and Certificates.

What is Log Analytics?

11/15/2016 • 3 min to read • Edit on GitHub

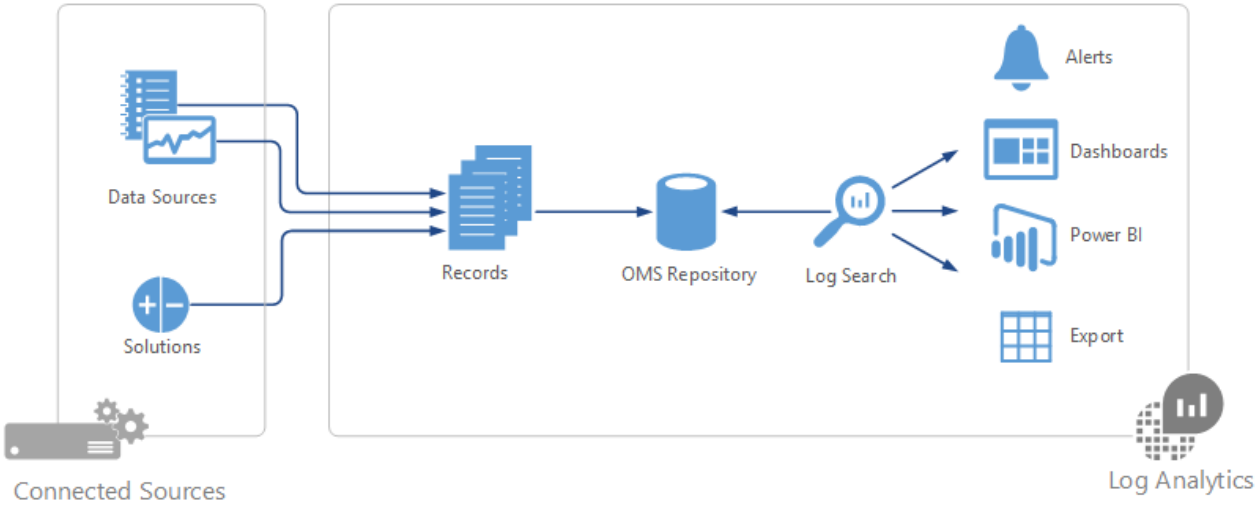
Contributors

Brian Wren • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Bill Anderson

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

Log Analytics components

At the center of Log Analytics is the OMS repository which is hosted in the Azure cloud. Data is collected into the repository from connected sources by configuring data sources and adding solutions to your subscription. Data sources and solutions will each create different record types that have their own set of properties but may still be analyzed together in queries to the repository. This allows you to use the same tools and methods to work with different kinds of data collected by different sources.



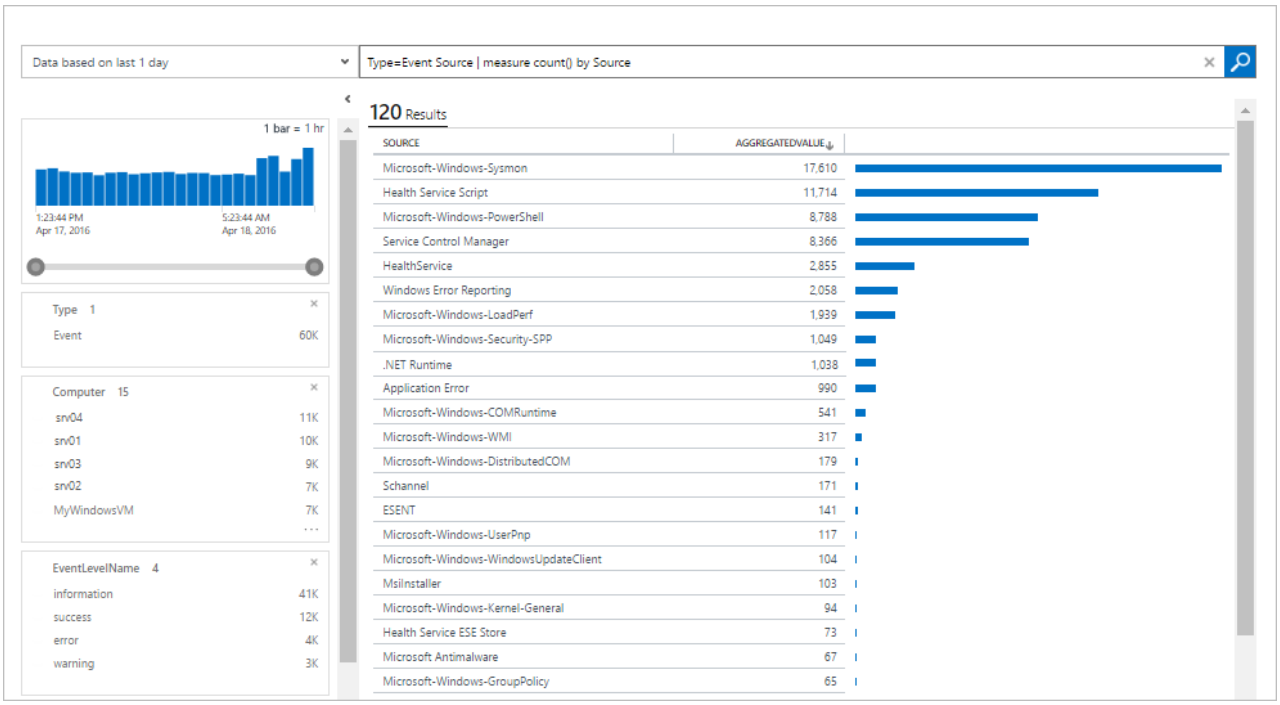
Connected sources are the computers and other resources that generate data collected by Log Analytics. This can include agents installed on Windows and Linux computers that connect directly or agents in a connected System Center Operations Manager management group. Log Analytics can also collect data from Azure storage.

Data sources are the different kinds of data collected from each connected source. This includes 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.

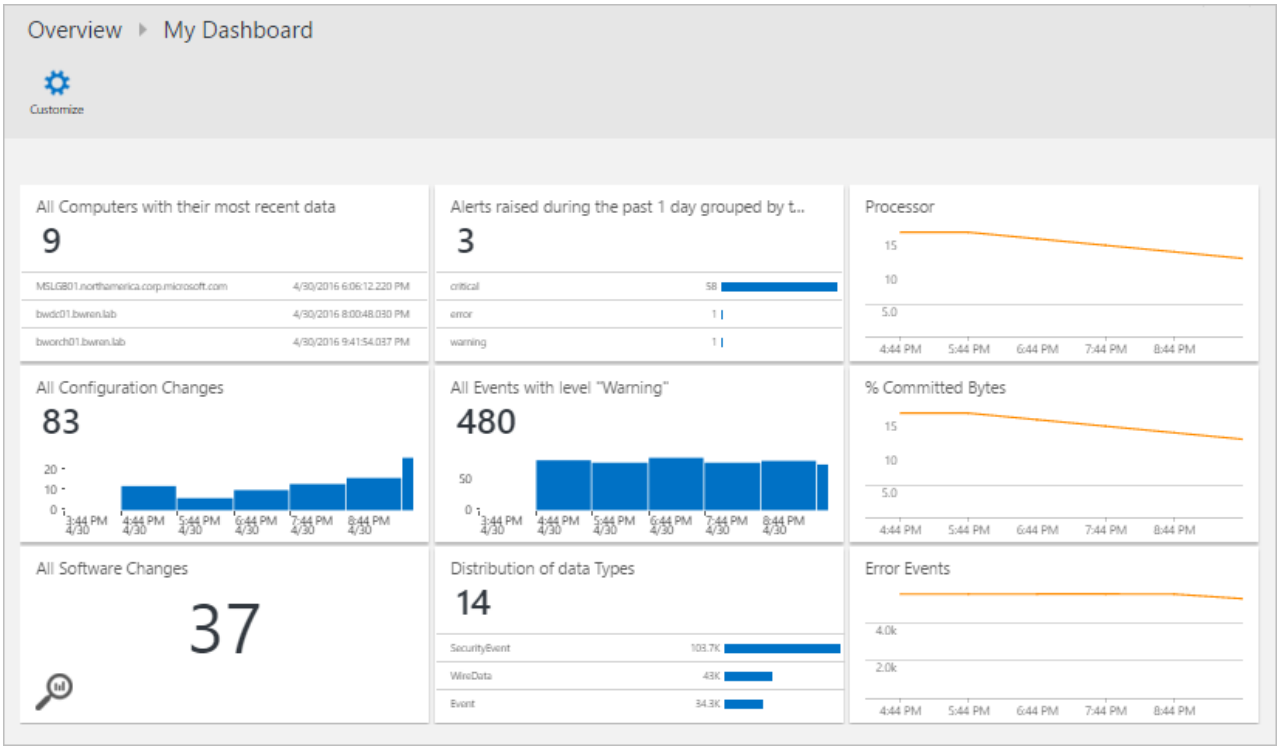
Analyzing Log Analytics data

Most of your interaction with Log Analytics will be through the OMS 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 leverage log searches where you construct queries to analyze collected data, dashboards which you can customize with graphical views of your most valuable searches, and solutions which provide additional functionality and analysis tools.

Log Analytics provides a query syntax to quickly retrieve and consolidate data in the repository. You can create and save Log Searches to directly analyze data in the OMS portal or have log searches run automatically to create an alert if the results of the query indicate an important condition.



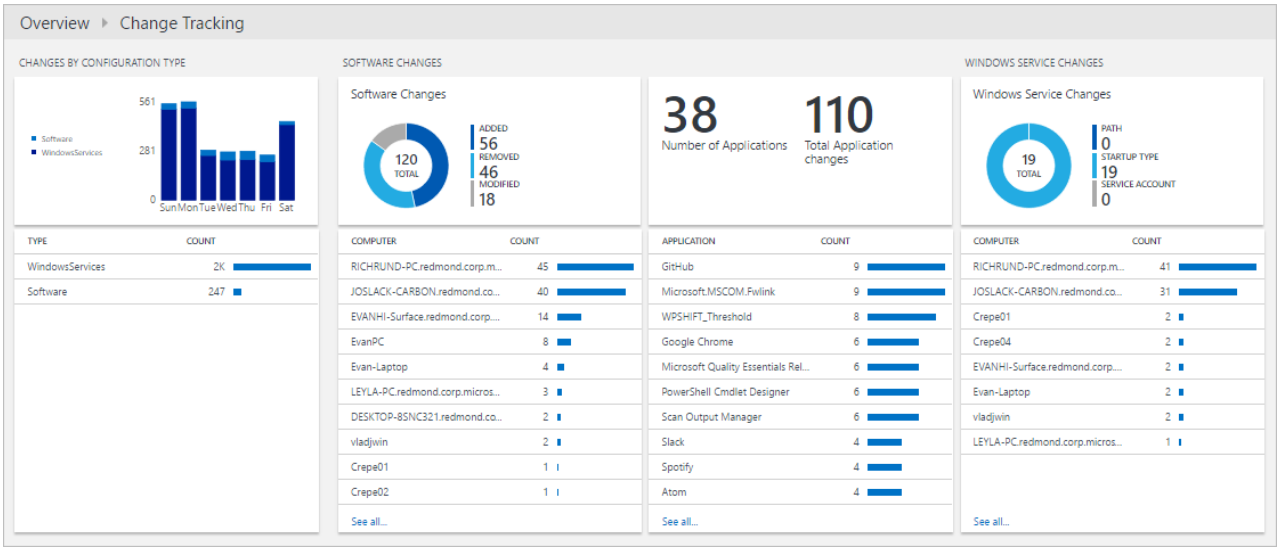
To give a quick graphical view of the health of your overall environment, you can add visualizations for saved log searches to your dashboard.



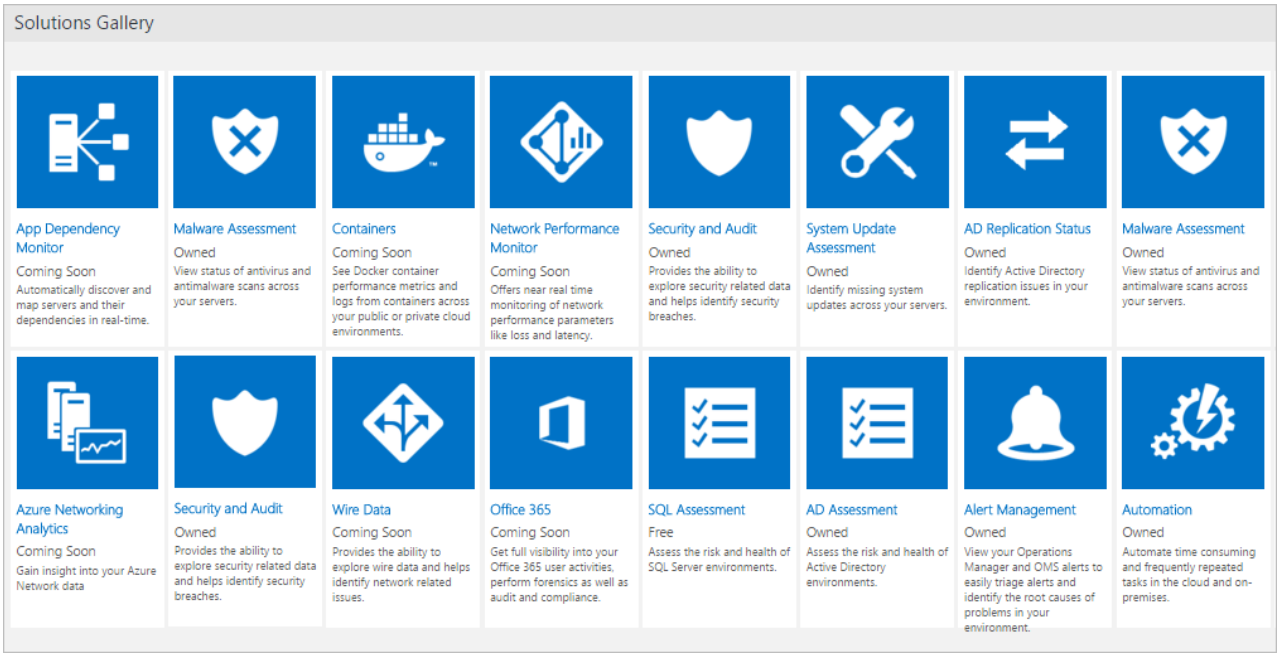
In order to analyze data outside of Log Analytics, you can export the data from the OMS repository into tools such as Power BI or Excel. You can also leverage the Log Search API to build custom solutions that leverage Log Analytics data or to integrate with other systems.

Solutions

Solutions add functionality to Log Analytics. They primarily run in the cloud and provide analysis of data collected in the OMS repository. They may also define new record types to be collected that can be analyzed with Log Searches or by additional user interface provided by the solution in the OMS dashboard.



Solutions are available for a variety of functions, and you can easily browse available solutions and add them to your OMS workspace from the Solutions Gallery. Many will be automatically deployed and start working immediately while others may require some configuration.

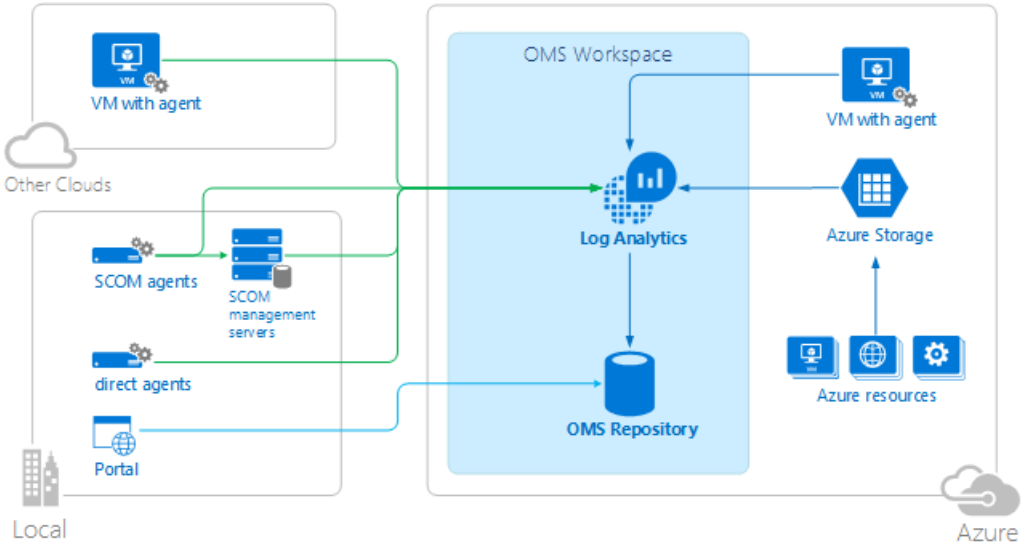


Log Analytics architecture

The deployment requirements of Log Analytics are minimal since the central components are hosted in the Azure cloud. This includes the repository in addition to the services that allow you to correlate and analyze collected data. The portal can be accessed from any browser so there is no requirement for client software.

You must install agents on Windows and Linux computers, but there is no additional agent required for computers that are already members of a connected SCOM management group. SCOM agents will continue to communicate with management servers which will forward their data to Log Analytics. Some solutions though will require agents to communicate directly with Log Analytics. The documentation for each solution will specify its communication requirements.

When you sign up for Log Analytics, you will create an OMS workspace. You can think of the workspace as a unique OMS environment with its own data repository, data sources, and solutions. You may create multiple workspaces in your subscription to support multiple environments such as production and test.



Next steps

- Sign up for a free Log Analytics account to test in your own environment.
- View the different Data Sources available to collect data into the OMS repository.
- Browse the available solutions in the Solutions Gallery to add functionality to Log Analytics.

What is Azure Multi-Factor Authentication?

11/15/2016 • 6 min to read • Edit on GitHub

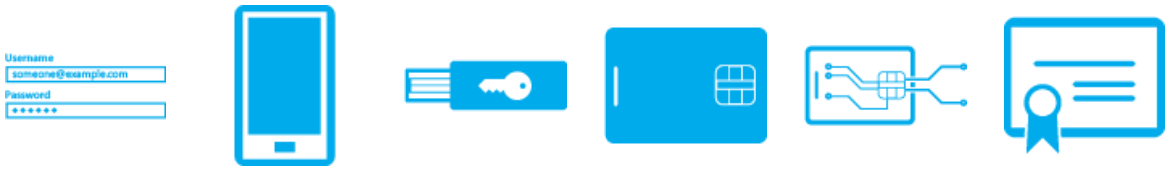
Contributors

Kelly Gremban • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Bill Mathers • unknown • Cody Mansfield • Glen Little • Steven M. Powell • femila • Matthew Baldwin • Dene Hager • curtand

---

Two-step verification 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:

- Something you know (typically a password)
- Something you have (a trusted device that is not easily duplicated, like a phone)
- Something you are (biometrics)



Azure Multi-Factor Authentication (MFA) is Microsoft's two-step verification solution. Azure 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 methods, including phone call, text message, or mobile app verification.

Why use Azure Multi-Factor Authentication?

Today, more than ever, people are increasingly connected. With smart phones, tablets, laptops, and PCs, people have several different options on how they are going to connect and stay connected at any time. People can access their accounts and applications from anywhere, which means that they can get more work done and serve their customers better.

Azure Multi-Factor Authentication is an easy to use, scalable, and reliable solution that provides a second method of authentication so your users are always protected.

			
Easy to use	**Scalable**	**Always Protected**	**Reliable**

- **Easy to Use** - Azure Multi-Factor Authentication is simple to set up and use. The extra protection that comes with Azure Multi-Factor Authentication allows users to manage their own devices. Best of all, in many instances it can be set up with just a few simple clicks.
- **Scalable** - Azure Multi-Factor Authentication uses the power of the cloud and integrates with your on-premises AD and custom apps. This protection is even extended to your high-volume, mission-critical scenarios.
- **Always Protected** - Azure Multi-Factor Authentication provides strong authentication using the highest industry standards.
- **Reliable** - We guarantee 99.9% availability of Azure Multi-Factor Authentication. The service is considered unavailable when it is unable to receive or process verification requests for the two-step verification.

How Azure Multi-Factor Authentication works

The security of two-step verification lies in its layered approach. Compromising multiple verification methods presents a significant challenge for attackers. Even if an attacker manages to learn your password, it is useless without also having possession of the trusted device. Should you lose the device, the person who finds it can't use it unless he or she also knows your password.

Methods available for Multi-Factor Authentication

When a user signs in, an additional verification request is sent to the user. The following are a list of methods that can be used for this second verification.

VERIFICATION METHOD	DESCRIPTION
Phone call	A call is placed to a user's phone asking them to verify that they are signing. Press the # key to complete the verification process. This option is configurable and can be changed to a code that you specify.
Text message	A text message is sent to a user's smart phone with a 6-digit code. Enter this code in to complete the verification process.
Mobile app notification	A verification request is sent to a user's smart phone asking them complete the verification by selecting **Verify** from the mobile app. This occurs if app notification is the primary verification method. If they receive this notification when they are not signing in, they can report it as fraud.
Verification code with mobile app	The mobile app on a user's device generates a verification code. This occurs if you selected a verification code as your primary verification method.

For the mobile app verification methods, Azure Multi-Factor Authentication works with third-party authentication apps for smart phones. However, we recommend the Microsoft Authenticator app, which is available for Windows Phone, Android, and IOS.

Available versions of Azure Multi-Factor Authentication

Azure Multi-Factor Authentication is available in three different versions.

VERSION	DESCRIPTION
Multi-Factor Authentication for Office 365	This version works exclusively with Office 365 applications and is managed from the Office 365 portal. So administrators can now help secure their Office 365 resources with two-step verification. This version comes with an Office 365 subscription.

VERSION	DESCRIPTION
Multi-Factor Authentication for Azure Administrators	The same subset of two-step verification capabilities for Office 365 is available at no cost to all Azure administrators. Every administrative account of an Azure subscription can enable this functionality for additional protection. An administrator that wants to access Azure portal to create a VM or web site, manage storage, or use any other Azure service can add MFA to his administrator account.
Azure Multi-Factor Authentication	Azure Multi-Factor Authentication offers the richest set of capabilities. It provides additional configuration options via the Azure classic portal, advanced reporting, and support for a range of on-premises and cloud applications. Azure Multi-Factor Authentication comes as part of Azure Active Directory Premium and Enterprise Mobility Suite, and can be deployed either in the cloud or on premises.

Feature comparison of versions

The following table provides a list of the features that are available in the various versions of Azure Multi-Factor Authentication.

> **NOTE**
>
> This comparison table discusses the features that are part of each subscription. If you have Azure AD Premium or Enterprise Mobility Suite, some features may not be available depending on whether you use MFA in the cloud or MFA on-premises.

FEATURE	MULTI-FACTOR AUTHENTICATION FOR OFFICE 365 (INCLUDED IN OFFICE 365 SKUS)	MULTI-FACTOR AUTHENTICATION FOR AZURE ADMINISTRATORS (INCLUDED WITH AZURE SUBSCRIPTION)	AZURE MULTI-FACTOR AUTHENTICATION (INCLUDED IN AZURE AD PREMIUM AND ENTERPRISE MOBILITY SUITE)
Administrators can protect accounts with MFA	●	● (Available only for Azure Administrator accounts)	●
Mobile app as a second factor	●	●	●
Phone call as a second factor	●	●	●
SMS as a second factor	●	●	●
App passwords for clients that don't support MFA	●	●	●
Admin control over authentication methods	●	●	●
PIN mode			●
Fraud alert			●

FEATURE	MULTI-FACTOR AUTHENTICATION FOR OFFICE 365 (INCLUDED IN OFFICE 365 SKUS)	MULTI-FACTOR AUTHENTICATION FOR AZURE ADMINISTRATORS (INCLUDED WITH AZURE SUBSCRIPTION)	AZURE MULTI-FACTOR AUTHENTICATION (INCLUDED IN AZURE AD PREMIUM AND ENTERPRISE MOBILITY SUITE)
MFA Reports			•
One-Time Bypass			•
Custom greetings for phone calls			•
Customization of caller ID for phone calls			•
Event Confirmation			•
Trusted IPs			•
Remember MFA for trusted devices	•	•	•
MFA SDK			• requires Multi-Factor Auth provider and full Azure subscription
MFA for on-premises applications using MFA server			•

How to get Azure Multi-Factor Authentication

If you would like the full functionality offered by Azure Multi-Factor Authentication, there are several options:

1. Purchase Azure Multi-Factor Authentication licenses and assign them to your users.
2. Purchase licenses that have Azure Multi-Factor Authentication bundled within them such as Azure Active Directory Premium, Enterprise Mobility Suite, or Enterprise Cloud Suite and assign them to your users.
3. Create an Azure Multi-Factor Authentication Provider within an Azure subscription. When using an Azure Multi-Factor Authentication Provider, there are two usage models available that are billed through your Azure subscription:
   - **Per User**. For enterprises that want to enable two-step verification for a fixed number of employees who regularly need authentication.
   - **Per Authentication**. For enterprises that want to enable two-step verification for a large group of external users who infrequently need authentication.

Azure Multi-Factor Authentication provides selectable verification methods for both cloud and server. This means that you can choose which methods are available for your users. This feature is currently in public preview for the cloud version of multi-factor authentication. For more information, see selectable verification methods.

For pricing details, see Azure MFA Pricing.

Next steps

To get started with Azure Multi-Factor Authentication, your first step is to choose between MFA in the cloud or on-premises

What is Azure Active Directory?

11/22/2016 • 5 min to read • Edit on GitHub

Contributors

curtand • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • MarkusVi • Sonia Wadhwa • Dene Hager • dstrockis • swkrish

---

Azure Active Directory (Azure AD) is Microsoft's multi-tenant cloud based directory and identity management service.

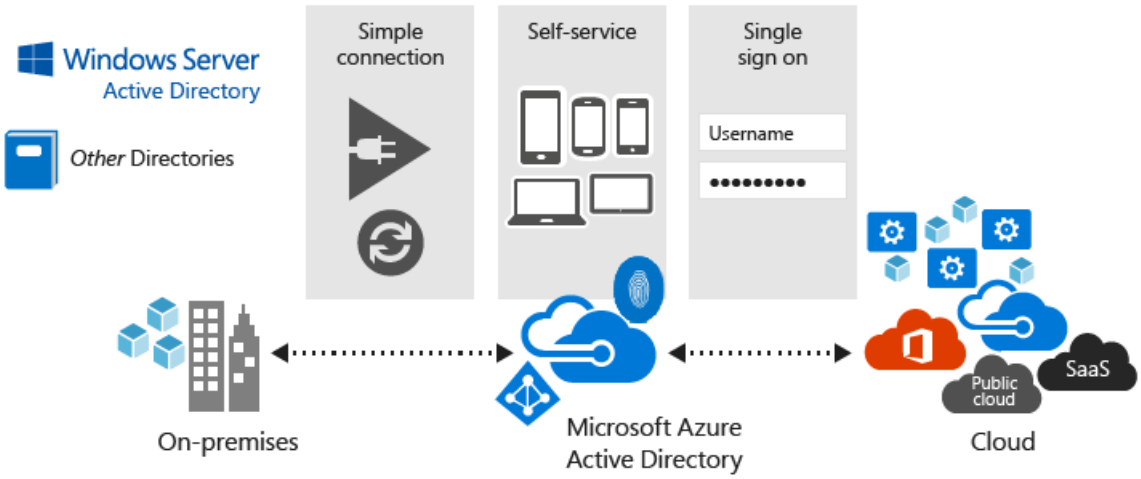
For IT Admins, Azure AD provides an affordable, easy to use solution to give employees and business partners single sign-on (SSO) access to thousands of cloud SaaS Applications like Office365, Salesforce.com, DropBox, and Concur.

For application developers, Azure AD lets you focus on building your application by making it fast and simple to integrate with a world class identity management solution used by millions of organizations around the world.

Azure AD also includes a full suite of identity management capabilities including multi-factor authentication, device registration, self-service password management, self-service group management, privileged account management, role based access control, application usage monitoring, rich auditing and security monitoring and alerting. These capabilities can help secure cloud based applications, streamline IT processes, cut costs and help ensure that corporate compliance goals are met.

Additionally, with just four clicks, Azure AD can be integrated with an existing Windows Server Active Directory, giving organizations the ability to leverage their existing on-premises identity investments to manage access to cloud based SaaS applications.

If you are an Office365, Azure or Dynamics CRM Online customer, you might not realize that you are already using Azure AD. Every Office365, Azure and Dynamics CRM tenant is actually already an Azure AD tenant. Whenever you want you can start using that tenant to manage access to thousands of other cloud applications Azure AD integrates with!



How reliable is Azure AD?

The multi-tenant, geo-distributed, high availability design of Azure AD means that you can rely on it for your most critical business needs. Running out of 28 data centers around the world with automated failover, you'll have the comfort of knowing that Azure AD is highly reliable and that even if a data center goes down, copies of your

directory data are live in at least two more regionally dispersed data centers and available for instant access.

For more details, see Service Level Agreements.

What are the benefits of Azure AD?

Your organization can use Azure AD to improve employee productivity, streamline IT processes, improve security and cut costs in many ways:

- Quickly adopt cloud services, providing employees and partners with an easy single-sign on experience powered by Azure AD's fully automated SaaS app access management and provisioning services capabilities.
- Empower employees with access to world class cloud apps and self-service capabilities from wherever they need to work on the devices they love to use.
- Easily and securely manage employee and vendor access to your corporate social media accounts.
- Improve application security with Azure AD multifactor authentication and conditional access.
- Implement consistent, self-service application access management, empowering business owners to move quickly while cutting IT costs and overheads.
- Monitor application usage and protect your business from advanced threats with security reporting and monitoring.
- Secure mobile (remote) access to on-premises applications.

How does Azure AD compare to on-premises Active Directory Domain Services (AD DS)?

Both Azure Active Directory (Azure AD) and on-premises Active Directory (Active Directory Domain Services or AD DS) are systems that store directory data and manage communication between users and resources, including user logon processes, authentication, and directory searches.

AD DS is a server role on Windows Server, which means that it can be deployed on physical or virtual machines. It has a hierarchical structure based on X.500. It uses DNS for locating objects, can be interacted with using LDAP, and it primarily uses Kerberos for authentication. Active Directory enables organizational units (OUs) and Group Policy Objects (GPOs) in addition to joining machines to the domain, and trusts are created between domains.

Azure AD is a multi-customer public directory service, which means that within Azure AD you can create a tenant for your cloud servers and applications such as Office 365. Users and groups are created in a flat structure without OUs or GPOs. Authentication is performed through protocols such as SAML, WS-Federation, and OAuth. It's possible to query Azure AD, but instead of using LDAP you must use a REST API called AD Graph API. These all work over HTTP and HTTPS.

You can use Azure AD Connect to sync your on-premises identities with Azure AD.

Authentication and authorization details

AZURE AD	ON-PREMISES AD DS
• SAML • WS-Federation • Interactive with supported credentials • OAuth 2.0 • OpenID Connect	• SAML • WS-Federation • NTLM • Kerberos • MD5 • Basic

Object repository details

AZURE AD	ON-PREMISES AD DS
Access via Azure AD Graph and Microsoft Graph	X.500 LDAP

Programmatic access details

AZURE AD	ON-PREMISES AD DS
MS/Azure AD Graph REST APIs	LDAP

SSO to applications details

AZURE AD	ON-PREMISES AD DS
• OpenID Connect • SAML	• SAML • WS-Fed • Open-ID connect

Access management details

AZURE AD	ON-PREMISES AD DS

• Resource-defined scope and role based access control
• Client-define delegated and application permissions
• Consent Framework (enforces proper user/admin consent, as defined/requested by resource/client)
• Via app role, can be applied individually or through groups, supports:
  • Admin managed
  • Self-service application access
  • User consent
	• Via ACLs, can be applied individually or through groups, supports:
	  • Admin managed

Group management details

AZURE AD	ON-PREMISES AD DS
• Admin managed • Rule/dynamic managed • Self-service group management	• Admin managed • External system (FIM, or other) required for: • Rule/dynamic managed

Supported credentials details

AZURE AD	ON-PREMISES AD DS
• Username + password • Smartcard	• Username + password • Smartcard

How can I get started?

- If you are an IT admin:
  - Try it out! - you can sign up for a free 30 trial today and deploy your first cloud solution in under 5 minutes using this link
  - Read “Getting started with Azure AD” for tips and tricks on getting an Azure AD tenant up and running fast

- If you are a developer:
  - Check out our Developers Guide to Azure Active Directory
  - Start a trial – sign up for a free 30 day trial today and start integrating your apps with Azure AD

Where can I learn more?

We have a ton of great resources online to help you learn all about Azure AD. Here's a list of great articles to get you started:

- Enabling your directory for hybrid management with Azure AD Connect
- Additional security for an ever connected world
- Automate User Provisioning and Deprovisioning to SaaS Applications with Azure Active Directory
- Getting started with Azure AD Reporting
- Manage your passwords from anywhere
- What is application access and single sign-on with Azure Active Directory?
- Automate User Provisioning and Deprovisioning to SaaS Applications with Azure Active Directory
- How to provide secure remote access to on-premises applications
- Managing access to resources with Azure Active Directory groups
- What is Microsoft Azure Active Directory licensing?
- How can I discover unsanctioned cloud apps that are used within my organization

Azure Security MVP Program

11/15/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes

---

Microsoft Most Valuable Professionals (MVPs) are community leaders who've demonstrated an exemplary commitment to helping others get the most out of their experience with Microsoft technologies. They share their exceptional passion, real-world knowledge, and technical expertise with the community and with Microsoft.

We are happy to announce that 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	• 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)	• Windows App Development • Classic Windows Development • Windows Bridges • Windows On Devices (IoT /Embedded) • Windows Hardware Engineering • Emerging Experiences (More Personal Computing)	• 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 Angular.js	• 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	• 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 we evaluate include 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

11/15/2016 • 1 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes • Barclay Neira • TerryLanfear

---

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 which can help our 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.

Azure security courses from Microsoft Virtual Academy

11/22/2016 • 3 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • TerryLanfear • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil • Yuri Diogenes

---

Microsoft Virtual Academy provides free, online training to help Developers, IT and Data Professionals, and students learn the latest technology, build their skills, and advance their careers.

On this page, you find a curated collection of Azure security-related courses. Visit the Microsoft Virtual Academy to see all the courses they have available.

Dev/Test in the Cloud

Are you a developer who needs to deliver faster and better applications? Moving your development and testing environments to the cloud can help you achieve exactly that! Learn how to get it done, and find out the benefits of making the move. Plus, see demonstrations and presentations that show you how Microsoft Azure can support your development and testing needs. Included are lesson on security development and deployment practices.

Common Tasks for Linux on Azure

If you have questions about using Linux on the Microsoft Azure platform, this detailed course has answers for you. Explore some common tasks with the experts from Linux Academy. Learn about creating a Linux virtual machine (VM) in Azure, accessing the Linux VM using remote desktop software, and running virtual hosts. Many security technologies and configurations are addressed in this course.

Secure the Cloud

In this session, learn how Microsoft can help you meet global compliance requirements, such as ISO 27001 / 27018, FedRAMP, PCI, and HIPAA, with new security controls. These controls range from at-rest data encryption, key management, VM protection, logging and monitoring, to anti-malware services, identity management, access controls, and more.

Design and Implement Cloud Data Platform Solutions

Learn the features and capabilities of Microsoft cloud data platform solutions. Get a platform overview and hear about security features, options for high availability, techniques for monitoring and managing cloud data, and more. Plus, get guidance on how to identify tradeoffs and make decisions for designing public and hybrid cloud solutions using Microsoft cloud data platform features.

Manage and Secure Identities in a Cloud and Mobile World

In this session, learn how Azure Active Directory and Microsoft Advanced Threat Analytics helps you secure and manage user identity, identify security breaches before they cause damage, and provide your users a single identity for accessing all corporate resources. Explore the technologies used to discover Shadow IT, manage application access, and monitor suspicious activity through advanced security reporting, user behavioral analytics, auditing, and alerting.

Security in a Cloud-Enabled World

Experts lead you through the customer responsibility roadmap in the Microsoft Cloud Security for Enterprise Architects poster. The experts also provide recommendations for modernizing each part of your security posture,

including governance, containment strategies, security operations, high-value asset protection, information protection, and user and device security, with a particular emphasis on protecting administrative control. Learn from the same framework that the Microsoft cybersecurity team uses to assess customers' cloud security and to build them a security roadmap.

Microsoft Azure IaaS Deep Dive

Learn how to use Microsoft Azure infrastructure capabilities. If you are an IT Pro, no need to have previous experience with Azure. This course walks you through creating and configuring Azure VMs, Azure Virtual Networks, and cross-premises connectivity to get things up and running on the cloud. Security features and considerations are included throughout the course.

Getting Started with Azure Security for the IT Professional

In this demo-filled course, a team of security experts and Azure engineers takes you beyond the basic certifications and explores what's possible inside Azure. See how to design and use various technologies to ensure that you have the security and architecture you need to successfully launch your projects in the cloud. Dive into datacenter operations, VM configuration, network architecture, and storage infrastructure.

Deep Dive into Azure Resource Manager Scenarios and Patterns

Explore Azure Resource Manager with a team of experts, who show you scripts and tools that make it easy to spin up or spin down elements of your application infrastructure. Explore the use of role-based access control (RBAC) to implement security with Azure Resource Manager.

Azure Rights Management Services Core Skills

Find out why information protection is a "must have" requirement in your organization and how rights management protects your organization's intellectual property, wherever it travels across devices and the cloud. Get hands-on experience and technical know-how from Microsoft experts.

Azure security videos on Channel 9

11/22/2016 • 3 min to read • Edit on GitHub

Contributors

Thomas W. Shinder, M.D • TerryLanfear • Andy Pasic • Kim Whitlatch (Beyondsoft Corporation) • Tyson Nevil

---

Channel 9 is a community that brings forward the people behind our products and connects them with customers.

They think there is a great future in software and they're excited about it. Channel 9 is a community to participate in the ongoing conversation.

The following is a curated list of Azure security presentations on Channel 9. Make sure to check this page monthly for new videos.

Accelerating Azure Consumption with Barracuda Security

See how you can use Barracuda security to secure your Azure deployments.

Azure Security Center - Threat Detection

With Azure Security Center, you get a central view of the security state of all your Azure resources. At a glance, verify that the appropriate security controls are in place and configured correctly. Scott talks to Sarah Fender who explains how Security Center integrates Threat Detection.

Azure Security Center Overview

With Azure Security Center, you get a central view of the security state of all your Azure resources. At a glance, verify that the appropriate security controls are in place and configured correctly. Scott talks to Sara Fender who explains it all!

Live Demo: Protecting against, Detecting and Responding to Threats

Join this session to see the Microsoft security platform in action. General Manager for Cloud & Enterprise, Julia White, demonstrates the security features of Windows 10, Azure, and Office 365 that can help you keep your organization secure.

Encryption in SQL Server Virtual Machines in Azure for better security

Jack Richins teaches Scott how to easily encrypt his SQL Server databases on Virtual Machine Azure instances. It's easier than you'd think!

Areas covered in this video:

- Understanding encryption and SQL Server
- Understanding the Data Protection API, master keys, and certificates
- Using SQL commands to create the master key and certificates, and encrypt the database

How to set security in DevTest Labs

As an owner of your lab, you can secure lab access by via two lab roles: Owner and DevTest Labs User. A person in the Owner role has complete access in the lab whereas a person in the DevTest Labs User role has limited access. In this video, we show you how to add a person in either of these roles to a lab.

Managing Secrets for Azure Apps

Every serious app you deploy on Azure has critical secrets – connection strings, certificates, keys. Silly mistakes in managing these secrets leads to fatal consequences – leaks, outages, compliance violations. As multiple recent surveys point out, silly mistakes cause four times more data breaches than adversaries. In this session, we go over some best practices to manage your important app secrets. These best practices may seem like common sense, yet many developers neglect them. We also go over how to use Azure Key Vault to implement those best practices. As an added benefit, following these practices helps you demonstrate compliance with standards such as SOC. The first 10 minutes of the session are level 100 and they apply to any cloud app you develop on any platform. The remainder is level 200-300 and focuses on apps you build on the Azure platform.

Securing your Azure Virtual Network using Network Security Groups with Narayan Annamalai

Senior Program Manager Narayan Annamalai teaches Scott how to use Network Security Groups within an Azure Virtual Network. You can create control access to objects within Azure by subnet and network! You learn how to control access and create groups within Azure using PowerShell.

Azure AD Privileged Identity Management: Security Wizard, Alerts, Reviews

Azure Active Directory (AD) Privileged Identity Management is a premium functionality that allows you to discover, restrict, and monitor privileged identities and their access to resources. It also enforces on-demand, just in time administrative access when needed. Learn about:

- Managing protection for Office 365 workload-specific administrative roles
- Configuring Azure Multi-Factor Authentication(MFA) for privileged role activations
- Measuring and improving your tenant security posture
- Monitoring and fixing security findings
- Reviewing who needs to remain in privileged roles for periodic recertification workflows

Azure Key Vault with Amit Bapat

Amit Bapat introduces Scott to Azure Key Vault. With Azure Key Vault, you can encrypt keys and small secrets like passwords using keys stored in hardware security modules (HSMs). It's cloud-based, hardware-based secret management for Microsoft Azure!