Microsoft

Microsoft Official Course

AZ-300T01

Deploying and Configuring Infrastructure

AZ-300T01

Deploying and Configuring Infrastructure

Contents

Module 0 Start Here . . . 1
Welcome to Deploying and Configuring Infrastructure . . . 1

Module 1 Module Managing Azure Subscriptions and Resources . . . 5
Exploring Monitoring Capabilities in Azure . . . 5
Azure Alerts . . . 14
Azure Activity Logs and Log Analytics . . . 22
Network Watcher . . . 34
Subscriptions and Accounts . . . 41
Online Lab - Exploring Monitoring Capabilities in Azure . . . 49
Review Questions . . . 54

Module 2 Module Implementing and Managing Storage . . . 57
Azure Storage Accounts . . . 57
Data Replication . . . 69
Storing and Accessing Data . . . 74
Monitoring Storage . . . 89
Review Questions . . . 102

Module 3 Module Deploying and Managing Virtual Machines (VMs) . . . 105
Creating Virtual Machines . . . 105
Backup and Restore . . . 117
Virtual Machine Extensions . . . 126
Monitoring Virtual Machines . . . 130
Online Lab - Implementing custom Azure VM images . . . 133
Review Questions . . . 138

Module 4 Module Configuring and Managing Virtual Networks . . . 141
Azure Virtual Networks . . . 141
Review of IP Addressing . . . 149
Network Routing . . . 155
Intersite Connectivity . . . 161
Virtual Network Peering . . . 171
Review Questions . . . 176

Module 5 Module Managing Identities . . . 179
Managing Azure Active Directory (AAD) . . . 179
Implementing and Managing Hybrid Identities . . . 190

Online Lab - Implementing User-Assigned Managed Identities for Azure Resources 199
Review Questions 204

Module 0 Start Here

Welcome to Deploying and Configuring Infrastructure

Welcome to Deploying and Configuring Infrastructure

About This Course: Deploying and Configuring Infrastructure

Welcome to *Deploying and Configuring Infrastructure*. This course is part of a series of five courses to help students prepare for Microsoft's Azure Solutions Architect technical certification exam AZ-300: Microsoft Azure Architect Technologies. These courses are designed for IT professionals and developers with experience and knowledge across various aspects of IT operations, including networking, virtualization, identity, security, business continuity, disaster recovery, data management, budgeting, and governance.

This course explains how to manage Azure resources, including deployment and configuration of virtual machines, virtual networks, storage accounts, and Azure AD including implementing and managing hybrid identities. You will learn how cloud resources are managed in Azure using group and user accounts as well as how to grant access to Azure AD users, groups, and services using Role-based access control (RBAC).

You will study the assorted storage accounts and services in addition to data replication concepts and replication schemes. You will be introduced to Storage Explorer as the convenient way to manage Azure storage data. For example, Azure blob storage is how Azure stores unstructured data in the cloud, and you will work with blobs and blob containers. Besides blob storage, this course covers table and queue storage as options for storing structured data.

You will learn how to create and deploy virtual machines in Azure using the Azure portal, PowerShell, and ARM templates including deploying custom images and Linux virtual machines. And, you will see how deploying highly available virtual machines is critical for planned and unplanned events, and how to use availability sets to ensure that virtual machine resources are available during downtime.

You will become skilled with the monitoring tools and capabilities provided by Azure, including Azure Alerts and Activity Logs. In addition to alerts and logs, you will be introduced to Log Analytics as an effective data analytics solution for understanding system status and health. And perhaps the most exciting thing you will discover is how to utilize the Azure Resource Manager deployment model to create and manage resources, resource groups, and ARM templates.

Because this course is the first course in the series for the Azure Solutions Architect exams, there is a sizeable amount of introductory content presented to prepare students for the remaining courses in the curriculum. Students are provided with a lesson that covers tips and tricks for working in the Azure portal, as well as an introduction to key tools used in the Azure environment, such as the Cloud Shell and Resource Explorer. Emphasis is focused on PowerShell and the command line interface (CLI) as important skills to acquire not only in preparation for the exam but for the job role itself.

The course outline is as follows:

Module 1 - Managing Azure Subscriptions and Resources

In this module you will explore Azure monitoring capabilities using Azure alerts, Azure activity logs, and Log Analytics. You will learn to query, analyze, and interpret the data viewed in Log Analytics.

Module 1 online lab:

- This module contains the lab Exploring Monitoring Capabilities in Azure.

Module 2 - Implementing and Managing Storage

In this module you will learn about Azure storage accounts, data replication, how to use Azure Storage Explorer, and monitor storage.

Module 3 - Deploying and Managing Virtual Machines (VMs)

In this module you will learn how to do the following:

- Create Virtual Machines (VM)s within the Azure Portal
- Create Virtual Machines (VM)s using Azure PowerShell
- Create Virtual Machines (VM)s using ARM templates
- Deploy Linux Virtual Machines (VM)s
- Monitor Virtual Machines (VM)s

Additionally, you will learn how to protect data using backups at regular intervals, using snapshots, Azure Backup, or Azure Site Recovery.

Module 3 online lab:

- This module contains the lab Implementing Custom Azure VM Images.

Module 4 - Configuring and Managing Virtual Networks

In this module you will create and implement virtual networks using the Azure Portal as well as Azure PowerShell and CLI. You will receive an overview on how to assign IP addresses to Azure resources to communicate with other Azure resources, an on-premises network, and the Internet. Additionally, you will gain a better understanding of the following:

- Network routing using routing tables and algorithms
- Inter-site connectivity using VNet-to-VNet connections and VPNs
- Virtual network peering for regional and global considerations
- Gateway transit to allow gateway transit for the virtual network to communicate with resources outside the peering

Module 5 - Managing Identities

This module covers Azure Active Directory (Azure AD) for IT Admins and Developers with a focus on the Azure AD multi-tenant cloud-based directory and identity management service.

Topics include: Role-Based Access Control (RBAC), built-in roles, Self-Service Password Reset (SSPR), and authentication methods for password reset.

Module 5 online lab:

- This module contains the lab Implementing User-Assigned Managed Identities.

What You'll Learn:

- Managing Azure Subscriptions and Resources
- Implementing and Managing Storage
- Deploying and Managing VMs
- Configuring and Managing Virtual Networks
- Managing Identities using Azure Active Directory

Prerequisites

Successful Cloud Solutions Architects begin this role with practical experience with operating systems, virtualization, cloud infrastructure, storage structures, billing, and networking.

Module 1 Module Managing Azure Subscriptions and Resources

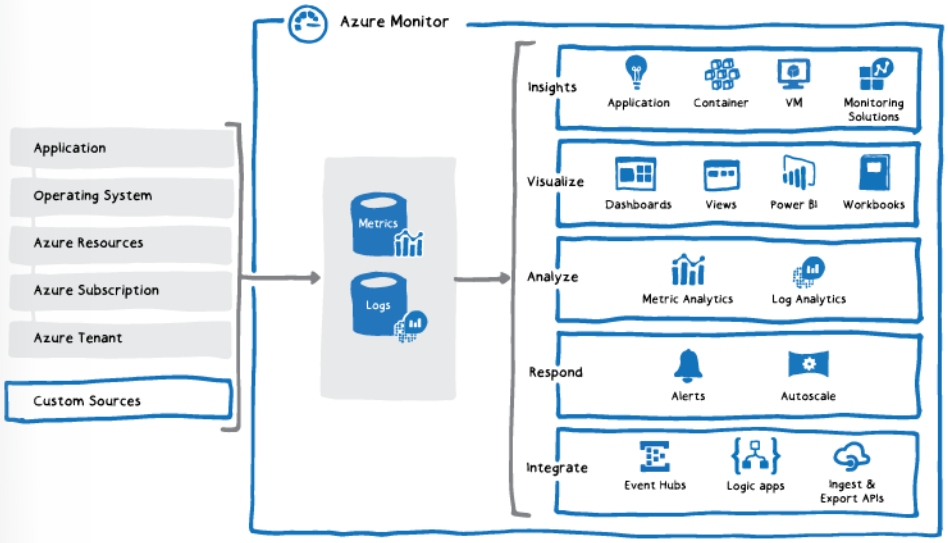
Exploring Monitoring Capabilities in Azure

Introducing Azure Monitoring Services

Monitoring is the act of collecting and analyzing data to determine the performance, health, and availability of your business application and the resources that it depends on. An effective monitoring strategy helps you understand the detailed operation of the components of your application. It also helps you increase your uptime by proactively notifying you of critical issues so that you can resolve them before they become problems.

Azure includes multiple services that individually perform a specific role or task in the monitoring space. Together, these services deliver a comprehensive solution for collecting, analyzing, and acting on telemetry from your application and the Azure resources that support them. They can also work to monitor critical on-premises resources to provide a hybrid monitoring environment. Understanding the tools and data that are available is the first step in developing a complete monitoring strategy for your application.

The next diagram gives a high-level view of Azure Monitor. At the center of the diagram are the data stores for metrics and logs, which are the two fundamental types of data use by Azure Monitor. On the left are the sources of monitoring data that populate these data stores. On the right are the different functions that Azure Monitor performs with this collected data such as analysis, alerting, and streaming to external systems.



For more information, you can see:

Azure Monitor Documentation- **https://docs.microsoft.com/en-us/azure/azure-monitor/**

Overview of Azure Monitor

The following diagram gives a high-level view of Azure Monitor. At the center of the diagram are the data stores for metrics and logs, which are the two fundamental types of data use by Azure Monitor. On the left are the sources of monitoring data that populate these data stores. On the right are the different functions that Azure Monitor performs with this collected data such as analysis, alerting, and streaming to external systems.

For more information, you can see: Monitoring Azure applications and resources - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview**

Monitoring data platform

All data collected by Azure Monitor fits into one of two fundamental types, metrics and logs. Metrics are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios. Logs contain different kinds of data organized into records with different sets of properties for each type. Telemetry such as events and traces are stored as logs in addition to performance data so that it can all be combined for analysis.

For many Azure resources, you'll see data collected by Azure Monitor right in their Overview page in the Azure portal. Have a look at any virtual machine for example, and you'll see several charts displaying performance metrics. Click on any of the graphs to open the data in Metric explorer in the Azure portal, which allows you to chart the values of multiple metrics over time. You can view the charts interactively or pin them to a dashboard to view them with other visualizations.

Log data collected by Azure Monitor is stored in Log Analytics which includes a rich query language to quickly retrieve, consolidate, and analyze collected data. You can create and test queries using the Log

Analytics page in the Azure portal and then either directly analyze the data using these tools or save queries for use with visualizations or alert rules.

Azure Monitor uses a version of the Data Explorer query language that is suitable for simple log queries but also includes advanced functionality such as aggregations, joins, and smart analytics. You can quickly learn the query language using multiple lessons. Particular guidance is provided to users who are already familiar with SQL and Splunk.

What data does Azure Monitor collect?

Azure Monitor can collect data from a variety of sources. You can think of monitoring data for your applications in tiers ranging from your application, any operating system and services it relies on, down to the platform itself. Azure Monitor collects data from each of the following tiers:

- Application monitoring data: Data about the performance and functionality of the code you have written, regardless of its platform.
- Guest OS monitoring data: Data about the operating system on which your application is running. This could be running in Azure, another cloud, or on-premises.
- Azure resource monitoring data: Data about the operation of an Azure resource.
- Azure subscription monitoring data: Data about the operation and management of an Azure subscription, as well as data about the health and operation of Azure itself.
- Azure tenant monitoring data: Data about the operation of tenant-level Azure services, such as Azure Active Directory.

As soon as you create an Azure subscription and start adding resources such as virtual machines and web apps, Azure Monitor starts collecting data. Activity Logs record when resources are created or modified. Metrics tell you how the resource is performing and the resources that it's consuming.

Extend the data you're collecting into the actual operation of the resources by enabling diagnostics and adding an agent to compute resources. This will collect telemetry for the internal operation of the resource and allow you to configure different data sources to collect logs and metrics from Windows and Linux guest operating system.

Add an instrumentation package to your application, to enable Application Insights to collect detailed information about your application including page views, application requests, and exceptions. Further verify the availability of your application by configuring an availability test to simulate user traffic.

Custom sources

Azure Monitor can collect log data from any REST client using the Data Collector API. This allows you to create custom monitoring scenarios and extend monitoring to resources that don't expose telemetry through other sources.

Insights

Monitoring data is only useful if it can increase your visibility into the operation of your computing environment. Azure Monitor includes several features and tools that provide valuable insights into your applications and other resources that they depend on. Monitoring solutions and features such as Application Insights and Container Insights provide deep insights into different aspects of your application and specific Azure services.

Application Insights

Application Insights monitors the availability, performance, and usage of your web applications whether they're hosted in the cloud or on-premises. It leverages the powerful data analysis platform in Azure Monitor to provide you with deep insights into your application's operations and diagnose errors without waiting for a user to report them. Application Insights includes connection points to a variety of development tools and integrates with Visual Studio to support your DevOps processes.

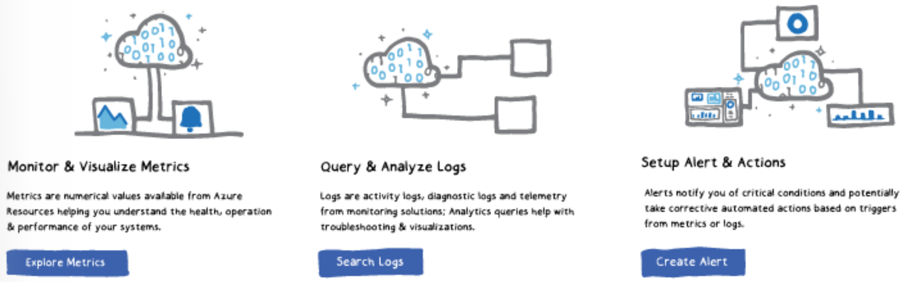
Azure Monitor: Key Capabilities

Azure Monitor enables core monitoring for Azure services by allowing the collection of metrics, activity logs, and diagnostic logs. For example, the activity log tells you when new resources are created or modified.

Metrics are available that provide performance statistics for different resources and even the operating system inside a virtual machine. You can view this data with one of the explorers in the Azure portal and create alerts based on these metrics. Azure Monitor provides the fastest metrics pipeline (5 minute down to 1 minute), so you should use it for time critical alerts and notifications.

Azure Monitor provides three main capabilities.

- **Monitor and visualize metrics**. Metrics are numerical values available from Azure resources helping you understand the health, operation and performance of your system.
- **Query and analyze logs**. Logs are activity logs, diagnostic logs, and telemetry from monitoring solutions; analytics queries help with troubleshooting and visualizations.
- **Setup alerts and actions**. Alerts notify you of critical conditions and potentially take automated corrective actions based on triggers from metrics or logs.



Monitor & Visualize Metrics

Metrics are numerical values available from Azure Resources helping you understand the health, operation & performance of your systems.

Query & Analyze Logs

Logs are activity logs, diagnostic logs and telemetry from monitoring solutions; Analytics queries help with troubleshooting & visualizations.

Setup Alert & Actions

Alerts notify you of critical conditions and potentially take corrective automated actions based on triggers from metrics or logs.

You can also send these metrics and logs to Azure Log Analytics for trending and detailed analysis, or create additional alert rules to proactively notify you of critical issues as a result of that analysis.

For more information, see: Get started with Azure Monitor – **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-get-started**

Monitoring Data Platform

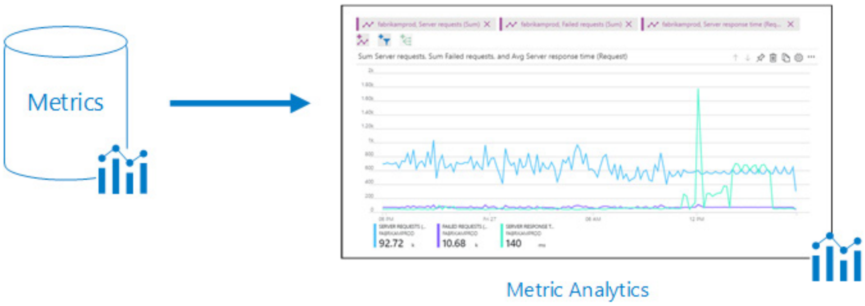
All data collected by Azure Monitor fits into one of two fundamental types, **metrics and logs**[1].

- **Metrics** are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios.

[1] https://docs.microsoft.com/en-us/azure/azure-monitor/platform/data-collection

- **Logs** contain different kinds of data organized into records with different sets of properties for each type. Telemetry such as events and traces are stored as logs in addition to performance data so that it can all be combined for analysis.

For many Azure resources, you'll see data collected by Azure Monitor right in their Overview page in the Azure portal. Have a look at any virtual machine for example, and you'll see several charts displaying performance metrics. Click on any of the graphs to open the data in Metric explorer in the Azure portal, which allows you to chart the values of multiple metrics over time. You can view the charts interactively or pin them to a dashboard to view them with other visualizations.

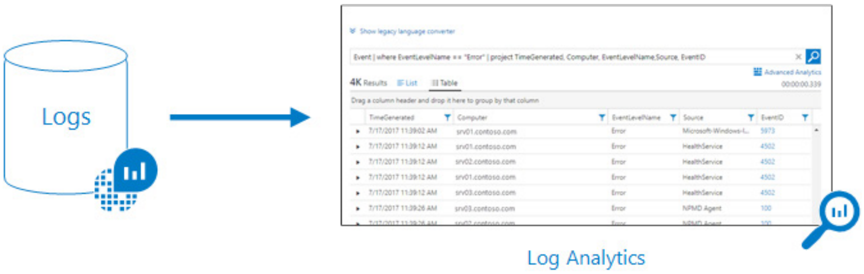

Metrics
fabrikamprod, Server requests (Sum)
fabrikamprod, Failed requests (Sum)
fabrikamprod, Server response time (Req...
Sum Server requests, Sum Failed requests, and Avg Server response time (Request)
2k
1.80k
1.60k
1.40k
1.20k
1k
800
600
400
200
0
06 PM
Fri 27
06 AM
12 PM
SERVER REQUESTS (...
FABRIKAMPROD
92.72 k
FAILED REQUESTS (...
FABRIKAMPROD
10.68 k
SERVER RESPONSE T...
FABRIKAMPROD
140 ms

Metric Analytics

Log Data

Log data collected by Azure Monitor is stored in Log Analytics which includes a **rich query language**[2] to quickly retrieve, consolidate, and analyze collected data. You can create and test queries using the Log Analytics page in the Azure portal and then either directly analyze the data using these tools or save queries for use with visualizations or alert rules.

Azure Monitor uses a version of the **Data Explorer**[3] query language that is suitable for simple log queries but also includes advanced functionality such as aggregations, joins, and smart analytics. You can quickly learn the query language using multiple lessons. Particular guidance is provided to users who are already familiar with SQL and Splunk.


Logs
Show legacy language converter
Advanced Analytics
4K Results
List
Table
00:00:00.339
Drag a column header and drop it here to group by that column

```
Event | where EventLevelName == "Error" | project TimeGenerated, Computer, EventLevelName,Source, EventID
```

TimeGenerated	Computer	EventLevelName	Source	EventID
7/17/2017 11:39:02 AM	srv01.contoso.com	Error	Microsoft-Windows-I...	5973
7/17/2017 11:39:12 AM	srv01.contoso.com	Error	HealthService	4502
7/17/2017 11:39:12 AM	srv02.contoso.com	Error	HealthService	4502
7/17/2017 11:39:12 AM	srv01.contoso.com	Error	HealthService	4502
7/17/2017 11:39:12 AM	srv03.contoso.com	Error	HealthService	4502
7/17/2017 11:39:26 AM	srv03.contoso.com	Error	NPMD Agent	100
7/17/2017 11:39:26 AM	srv02.contoso.com	Error	NPMD Agent	100

Log Analytics

Data Types

Azure Monitor can collect data from a variety of sources. You can think of monitoring data for your applications in tiers ranging from your application, any operating system and services it relies on, down to the platform itself. Azure Monitor collects data from each of the following tiers:

- **Application monitoring data**: Data about the performance and functionality of the code you have written, regardless of its platform.

[2] https://docs.microsoft.com/en-us/azure/azure-monitor/log-query/log-query-overview
[3] https://docs.microsoft.com/en-us/azure/kusto/query/

- **Guest OS monitoring data**: Data about the operating system on which your application is running. This could be running in Azure, another cloud, or on-premises.
- **Azure resource monitoring data**: Data about the operation of an Azure resource.
- **Azure subscription monitoring data**: Data about the operation and management of an Azure subscription, as well as data about the health and operation of Azure itself.
- **Azure tenant monitoring data**: Data about the operation of tenant-level Azure services, such as Azure Active Directory.

As soon as you create an Azure subscription and start adding resources such as virtual machines and web apps, Azure Monitor starts collecting data. Activity Logs record when resources are created or modified. Metrics tell you how the resource is performing and the resources that it's consuming.

Extend the data you're collecting into the actual operation of the resources by enabling diagnostics and adding an agent to compute resources. This will collect telemetry for the internal operation of the resource and allow you to configure different data sources to collect logs and metrics from Windows and Linux guest operating systems.

✓ Azure Monitor can collect log data from any REST client using the Data Collector API. This allows you to create custom monitoring scenarios and extend monitoring to resources that don't expose telemetry through other sources.

For more information, you can see:

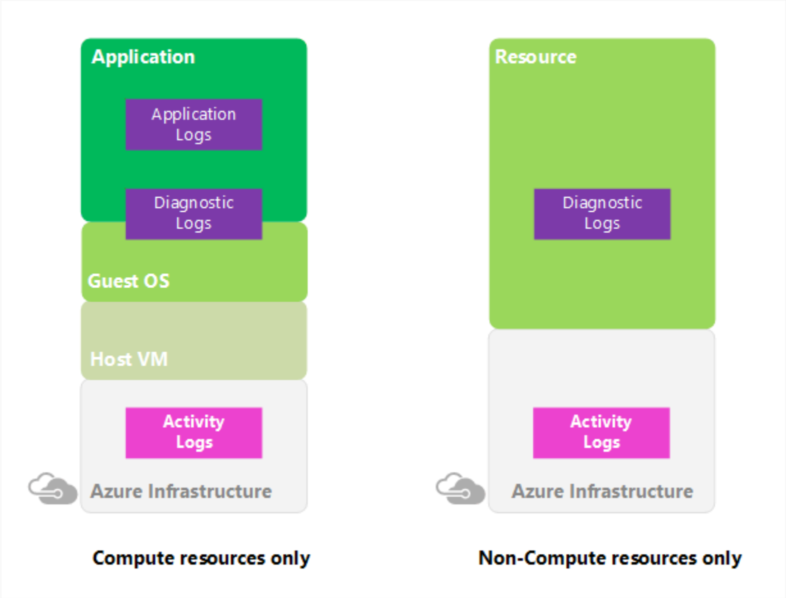
Azure Fridays, Azure Monitor - **https://channel9.msdn.com/Shows/Azure-Friday/Azure-Monitor/player**

Activity Log

The Azure Activity Log is a subscription log that provides insight into subscription-level events that have occurred in Azure. This includes a range of data, from Azure Resource Manager operational data to updates on Service Health events.

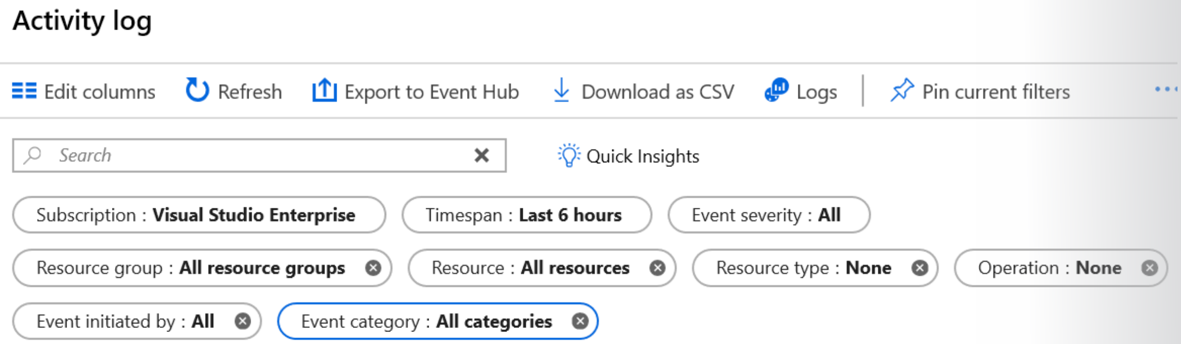
With the Activity Log, you can determine the 'what, who, and when' for any write operations (PUT, POST, DELETE) taken on the resources in your subscription. You can also understand the status of the operation and other relevant properties. Through activity logs, you can determine:

- What operations were taken on the resources in your subscription.
- Who started the operation.
- When the operation occurred.
- The status of the operation.
- The values of other properties that might help you research the operation.



✓ Activity logs are kept for 90 days. You can query for any range of dates, as long as the starting date isn't more than 90 days in the past. You can retrieve events from your Activity Log using the Azure portal, CLI, PowerShell cmdlets, and Azure Monitor REST API.

Query the Activity Log



Subscription : **Visual Studio Enterprise**
Timespan : **Last 6 hours**
Event severity : **All**
Resource group : **All resource groups**
Resource : **All resources**
Resource type : **None**
Operation : **None**
Event initiated by : **All**
Event category : **All categories**

In the Azure portal, you can filter your Activity Log by these fields:

- **Subscription**. One or more Azure subscription names.
- **Timespan**. The start and end time for events.
- **Event Severity**. The severity level of the event (Informational, Warning, Error, Critical).
- **Resource group**. One or more resource groups within those subscriptions.
- **Resource (name)**. The name of a specific resource.
- **Resource type**. The type of resource, for example, Microsoft.Compute/virtualmachines.
- **Operation name**. The name of an Azure Resource Manager operation, for example, Microsoft.SQL/servers/Write.
- **Event initiated by**. The 'caller,' or user who performed the operation.

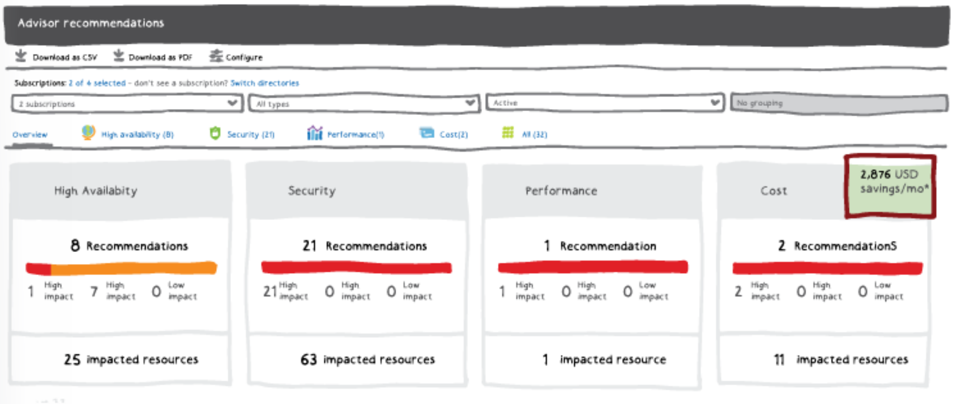
- **Event Category**. The event category is described in the next topic.
- **Search**. This is an open text search box that searches for that string across all fields in all events.

✓ Once you have defined a set of filters, you can pin the filtered state to the dashboard or download the search results as a CSV file.

Azure Advisor

Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, high availability, and security of your Azure resources.

The Advisor cost recommendations page helps you optimize and reduce your overall Azure spend by identifying idle and underutilized resources.



Advisor recommendations

Download as CSV | Download as PDF | Configure

Subscriptions: 2 of 4 selected - don't see a subscription? Switch directories

2 subscriptions | All types | Active | No grouping

Overview | High availability (8) | Security (21) | Performance(1) | Cost(2) | All (32)

High Availabity	Security	Performance	Cost (2,876 USD savings/mo*)
8 Recommendations	21 Recommendations	1 Recommendation	2 RecommendationS
1 High impact, 7 High impact, 0 Low impact	21 High impact, 0 High impact, 0 Low impact	1 High impact, 0 High impact, 0 Low impact	2 High impact, 0 High impact, 0 Low impact
25 impacted resources	63 impacted resources	1 impacted resource	11 impacted resources

Select the recommended action for a recommendation to implement the recommendation. A simple interface will open that enables you to implement the recommendation or refer you to documentation that assists you with implementation.

✓ Advisor provides recommendations for virtual machines, availability sets, application gateways, App Services, SQL servers, and Redis Cache.

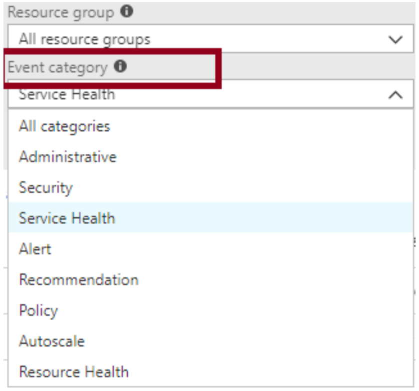
For more information, you can see:

Introduction to Azure Advisor - **https://docs.microsoft.com/en-us/azure/advisor/advisor-overview**

Advisor Cost recommendations - **https://docs.microsoft.com/en-us/azure/advisor/advisor-cost-recommendations**

Event Categories

The Activity Log provides several event categories. You may select one or more.



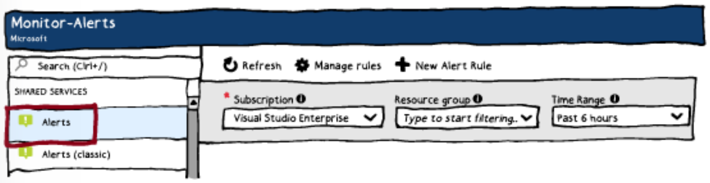
Resource group
All resource groups
Event category
Service Health

- **Administrative**. This category contains the record of all create, update, delete, and action operations performed through Resource Manager. Examples of the types of events you would see in this category include “create virtual machine” and "delete network security group". The Administrative category also includes any changes to role-based access control in a subscription.
- **Service Health**. This category contains the record of any service health incidents that have occurred in Azure. An example of the type of event you would see in this category is “SQL Azure in East US is experiencing downtime.” Service health events come in five varieties: Action Required, Assisted Recovery, Incident, Maintenance, Information, or Security.
- **Alert**. This category contains the record of all activations of Azure alerts. An example of the type of event you would see in this category is “CPU % on myVM has been over 80 for the past 5 minutes.”
- **Autoscale**. This category contains the record of any events related to the operation of the autoscale engine based on any autoscale settings you have defined in your subscription. An example of the type of event you would see in this category is “Autoscale scale up action failed.”
- **Recommendation**. This category contains recommendation events from certain resource types, such as web sites and SQL servers. These events offer recommendations for how to better utilize your resources.
- **Security**. This category contains the record of any alerts generated by Azure Security Center. An example of the type of event you would see in this category is “Suspicious double extension file executed.”
- **Policy and Resource Health**. These categories do not contain any events; they are reserved for future use.

Azure Alerts

Azure Monitor Alerts

Alerting is now available with Azure Monitor.



* Subscription: Visual Studio Enterprise

Resource group: Type to start filtering..

Time Range: Past 6 hours

The Monitor Alerts experience has many benefits.

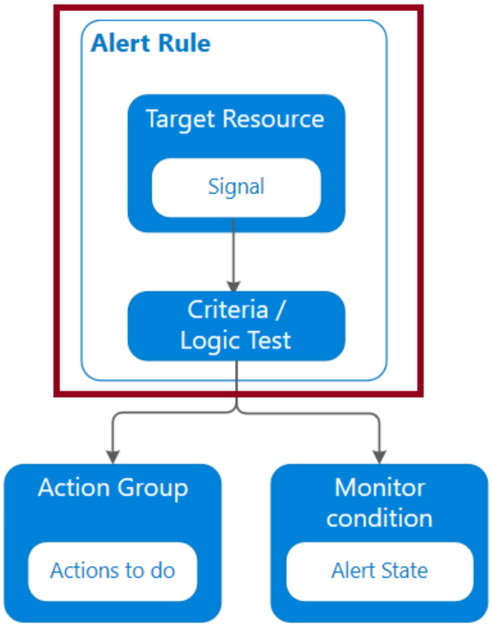
- **Better notification system**. All newer alerts use action groups, which are named groups of notifications and actions that can be reused in multiple alerts.
- **A unified authoring experience**. All alert creation for metrics, logs and activity log across Azure Monitor, Log Analytics, and Application Insights is in one place.
- **View Log Analytics alerts in Azure portal**. You can now also see Log Analytics alerts in your subscription. Previously these were in a separate portal.
- **Separation of Fired Alerts and Alert Rules**. Alert Rules (the definition of the condition that triggers an alert), and Fired Alerts (an instance of the alert rule firing) are differentiated, so the operational and configuration views are separated.
- **Better workflow**. The new alerts authoring experience guides the user along the process of configuring an alert rule, which makes it simpler to discover the right things to get alerted on.

For more information, you can see:

The new alerts experience in Azure Monitor - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-unified-alerts**

Creating Alert Rules

Alerts proactively notify you when important conditions are found in your monitoring data. They allow you to identify and address issues before the users of your system notice them. Alerts consists of alert rules, action groups, and monitor conditions.

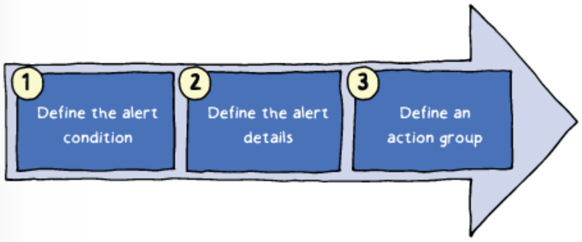


Alert rules are separated from alerts and the actions that are taken when an alert fires. The alert rule captures the target and criteria for alerting. The alert rule can be in an enabled or a disabled state. Alerts only fire when enabled. The key attributes of an alert rule are:

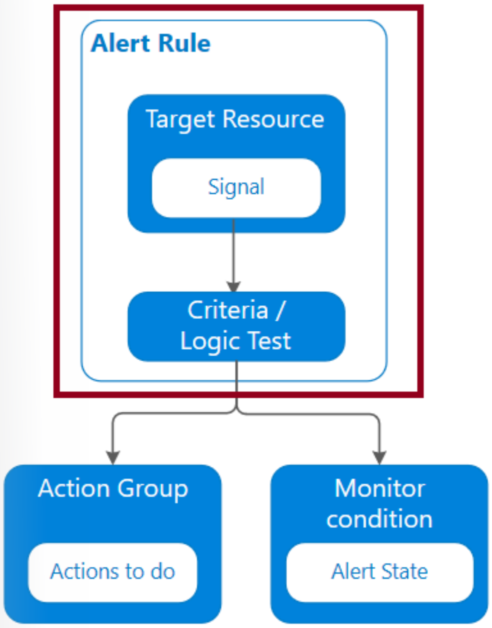
- **Target Resource** – Defines the scope and signals available for alerting. A target can be any Azure resource. Example targets: a virtual machine, a storage account, a virtual machine scale set, a Log Analytics workspace, or an Application Insights resource. For certain resources (like Virtual Machines), you can specify multiple resources as the target of the alert rule.
- **Signal** – Signals are emitted by the target resource and can be of several types. Metric, Activity log, Application Insights, and Log.
- **Criteria** – Criteria is a combination of Signal and Logic applied on a Target resource. Examples: + Percentage CPU > 70%; Server Response Time > 4 ms; and Result count of a log query > 100.
- **Alert Name** – A specific name for the alert rule configured by the user.
- **Alert Description** – A description for the alert rule configured by the user.
- **Severity** – The severity of the alert once the criteria specified in the alert rule is met. Severity can range from 0 to 4.
- **Action** – A specific action taken when the alert is fired. See the Action Groups topic coming up.

Alert Rules

Creating an alert is a three-step task: define the alert condition, define alert details, and define an action group.



Alerts proactively notify you when important conditions are found in your monitoring data. They allow you to identify and address issues before the users of your system notice them. Alerts consists of alert rules, action groups, and monitor conditions.



Alert rules are separated from alerts and the actions that are taken when an alert fires. The alert rule captures the target and criteria for alerting. The alert rule can be in an enabled or a disabled state. Alerts only fire when enabled. The key attributes of an alert rule are:

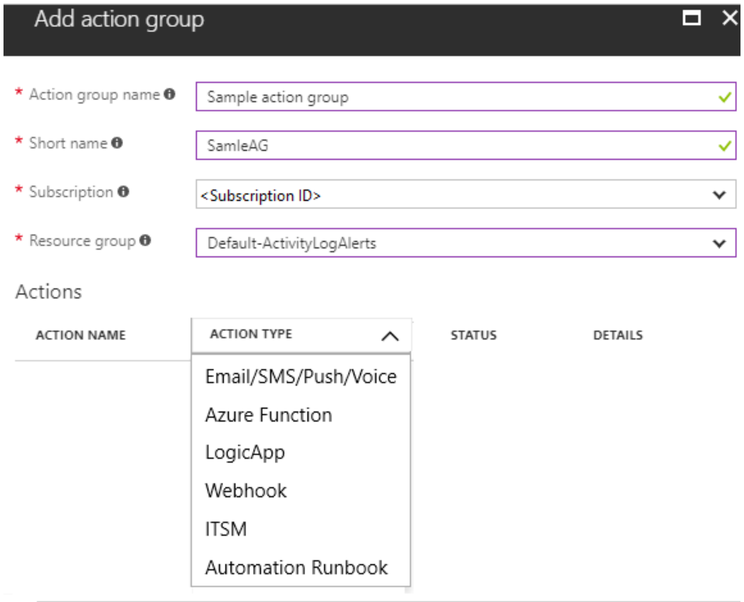
- **Target Resource** – Defines the scope and signals available for alerting. A target can be any Azure resource. Example targets: a virtual machine, a storage account, a virtual machine scale set, a Log Analytics workspace, or an Application Insights resource. For certain resources (like Virtual Machines), you can specify multiple resources as the target of the alert rule.
- **Signal** – Signals are emitted by the target resource and can be of several types. Metric, Activity log, Application Insights, and Log.
- **Criteria** – Criteria is a combination of Signal and Logic applied on a Target resource. Examples: + Percentage CPU > 70%; Server Response Time > 4 ms; and Result count of a log query > 100.
- **Alert Name** – A specific name for the alert rule configured by the user.
- **Alert Description** – A description for the alert rule configured by the user.
- **Severity** – The severity of the alert once the criteria specified in the alert rule is met. Severity can range from 0 to 4.

- **Action** – A specific action taken when the alert is fired. See the Action Groups topic coming up.

Action Groups

An action group is a collection of notification preferences defined by the owner of an Azure subscription. Azure Monitor and Service Health alerts use action groups to notify users that an alert has been triggered. Various alerts may use the same action group or different action groups depending on the user's requirements.

When an action is configured to notify a person by email or SMS the person will receive a confirmation indicating he / she has been added to the action group.



Add action group

* Action group name: Sample action group

* Short name: SamleAG

* Subscription: <Subscription ID>

* Resource group: Default-ActivityLogAlerts

Actions

ACTION NAME	ACTION TYPE	STATUS	DETAILS

- **Email** – Emails will be sent to the email addresses. Ensure that your email filtering is configured appropriately. You may have up to 1000 email actions in an Action Group.
- **ITSM** – You may have up to 10 ITSM actions in an Action Group ITSM Action requires an ITSM Connection.
- **Logic App** – You may have up to 10 Logic App actions in an Action Group.
- **Function App** – The function keys for Function Apps configured as actions are read through the Functions API.
- **Runbook** – You may have up to 10 Runbook actions in an Action Group.
- **SMS** – You may have up to 10 SMS actions in an Action Group.
- **Voice** – You may have up to 10 Voice actions in an Action Group.
- **Webhook** – You may have up to 10 Webhook actions in an Action Group. Retry logic - The timeout period for a response is 10 seconds. The webhook call will be retried a maximum of 2 times when the following HTTP status codes are returned: 408, 429, 503, 504 or the HTTP endpoint does not respond. The first retry happens after 10 seconds. The second and last retry happens after 100 seconds.

✓ You may have up to 10 Azure app actions in an Action Group. At this time the Azure app action only supports ServiceHealth alerts.

Managing Alerts

You can alert on metrics and logs as described in monitoring data sources. These include but are not limited to:

- Metric values
- Log search queries
- Activity Log events
- Health of the underlying Azure platform
- Tests for web site availability

Alert states

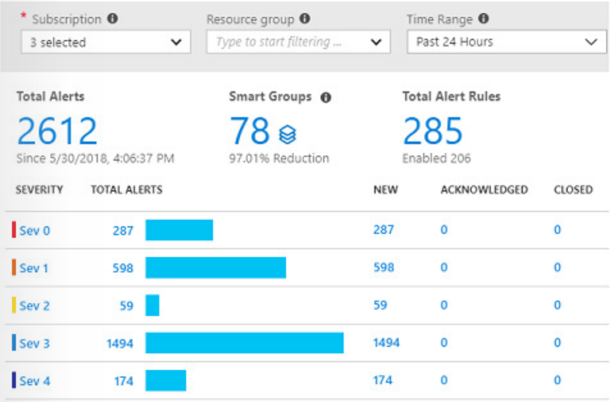
You can set the state of an alert to specify where it is in the resolution process. When the criteria specified in the alert rule is met, an alert is created or fired, it has a status of **New**. You can change the status when you acknowledge an alert and when you close it. All state changes are stored in the history of the alert. The following alert states are supported.

State	Description
New	The issue has just been detected and has not yet been reviewed.
Acknowledged	An administrator has reviewed the alert and started working on it.
Closed	The issue has been resolved. After an alert has been closed, you can reopen it by changing it to another state.

✓ Alert state is different and independent of the monitor condition. Alert state is set by the user. Monitor condition is set by the system. When an alert fires, the alert's monitor condition is set to fired. When the underlying condition that caused the alert to fire clears, the monitor condition is set to resolved. The alert state isn't changed until the user changes it.

Alerts Experience

The default Alerts page provides a summary of alerts that are created within a particular time window. It displays the total alerts for each severity with columns that identify the total number of alerts in each state for each severity.



* Subscription: 3 selected

Resource group: Type to start filtering ...

Time Range: Past 24 Hours

Total Alerts: 2612 — Since 5/30/2018, 4:06:37 PM

Smart Groups: 78 — 97.01% Reduction

Total Alert Rules: 285 — Enabled 206

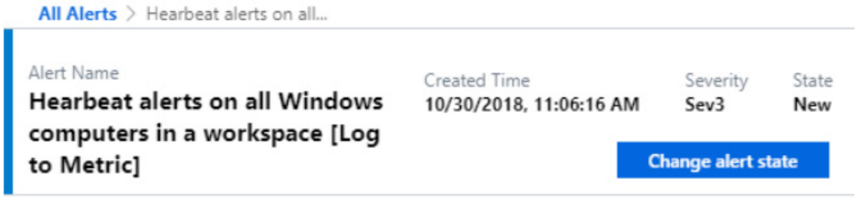
SEVERITY	TOTAL ALERTS	NEW	ACKNOWLEDGED	CLOSED
Sev 0	287	287	0	0
Sev 1	598	598	0	0
Sev 2	59	59	0	0
Sev 3	1494	1494	0	0
Sev 4	174	174	0	0

Column	Description
Subscription	Select up to five Azure subscriptions. Only alerts in the selected subscriptions are included in the view.
Resource group	Select a single resource group. Only alerts with targets in the selected resource group are included in the view.
Time range	Only alerts fired within the selected time window are included in the view. Supported values are the past hour, the past 24 hours, the past 7 days, and the past 30 days.

✓ You can select Total Alerts, Smart Groups, and Total Alert Rules to open a new page.

Alert Detail Page

The Alert detail page is displayed when you select an alert. It provides details of the alert and enables you to change its state.



All Alerts > Hearbeat alerts on all...

Alert Name: **Hearbeat alerts on all Windows computers in a workspace [Log to Metric]**

Created Time: 10/30/2018, 11:06:16 AM

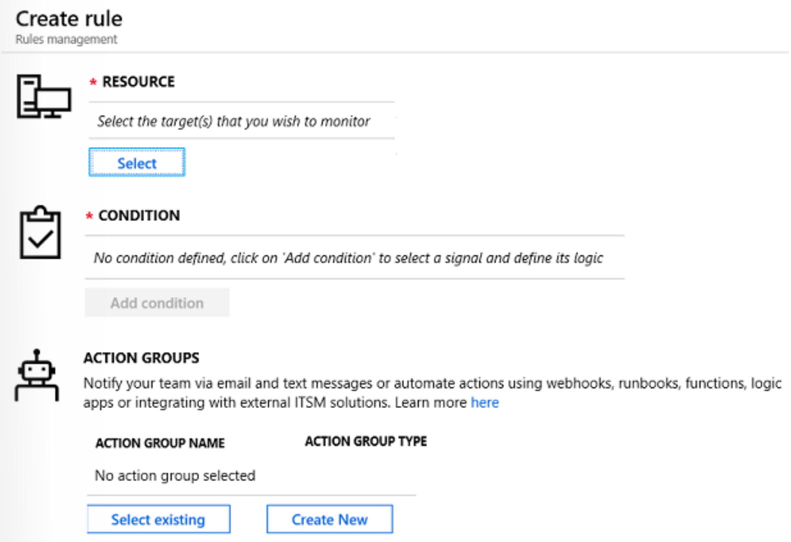
Severity: Sev3

State: New

Section	Description
Essentials	Displays the properties and other significant information about the alert.
History	Lists each action taken by the alert and any changes made to the alert. Currently limited to state changes.
Smart group	Information about the smart group the alert is included in. The alert count refers to the number of alerts that are included in the smart group. Includes other alerts in the same smart group that were created in the past 30 days regardless of the time filter in the alerts list page. Select an alert to view its detail.
More details	Displays further contextual information for the alert, which is typically specific to the type of source that created the alert.

Create an Alert

Alerts can be authored in a consistent manner regardless of the monitoring service or signal type. All fired alerts and related details are available in single page. You create a new alert rule with the following three steps:



Create rule
Rules management

* RESOURCE

Select the target(s) that you wish to monitor

Select

* CONDITION

No condition defined, click on 'Add condition' to select a signal and define its logic

Add condition

ACTION GROUPS

Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions. Learn more here

ACTION GROUP NAME	ACTION GROUP TYPE
No action group selected	

Select existing   Create New

- **Resource**. Select the resource you want to monitor. For example, resource group, virtual machine, or storage account.
- **Condition**. Select the signal and define its logic. The signal could be All, Metrics, or Activity log.
- **Action Group**. Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions.
- **Alert rule name**. Specify a name to identify your alert.
- **Description**. Provide a description for your alert rule.
- **Enable rule upon creation**. You can enable and disable your alert rules.

✓ We currently support configuring only two metrics signals or one log search signal or one activity log signal per alert rule. An alert will be triggered when the conditions for all the above configured criteria are met.

Demonstration - Alerts

In this demonstration, we will create an alert rule.

Create an alert rule

1. In Azure portal, click on **Monitor**. The Monitor blade consolidates all your monitoring settings and data in one view.
2. Click **Alerts** then click **+ New alert rule**. As most resource blades also have Alerts in their resource menu under Monitoring, you could create alerts from there as well.

Explore alert targets

1. Click **Select** under Target, to select a target resource that you want to alert on. Use **Subscription** and **Resource type** drop-downs to find the resource you want to monitor. You can also use the search bar to find your resource.
2. If the selected resource has metrics you can create alerts on, Available signals on the bottom right will include metrics. You can view the full list of resource types supported for metric alerts in this article.
3. Click **Done** when you have made your selection.

Explore alert conditions

1. Once you have selected a target resource, click on **Add condition**.
2. You will see a list of signals supported for the resource, select the metric you want to create an alert on.
3. Optionally, refine the metric by adjusting Period and Aggregation. If the metric has dimensions, you will see the Dimensions table presented.
4. You will see a chart for the metric for the last 6 hours. Adjust the **Show history** drop-down.
5. Define the **Alert logic**. This will determine the logic which the metric alert rule will evaluate.
6. If you are using a static threshold, the metric chart can help determine what might be a reasonable threshold. If you are using a Dynamic Thresholds, the metric chart will display the calculated thresholds based on recent data.
7. Click **Done**.
8. Optionally, add another criteria if you want to monitor a complex alert rule.

Explore alert details

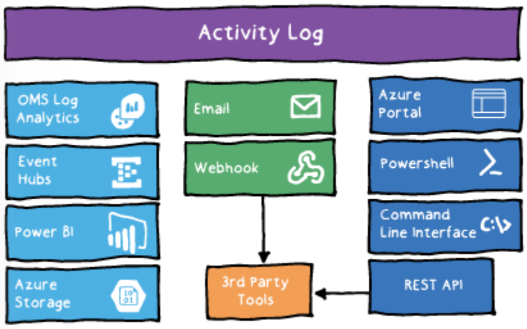
1. Fill in Alert details like **Alert Rule Name**, **Description** and **Severity**.
2. Add an action group to the alert either by selecting an existing action group or creating a new action group.
3. Click **Done** to save the metric alert rule.

Azure Activity Logs and Log Analytics

Overview of Activity Log

The Azure Activity Log is a subscription log that provides insight into subscription-level events that have occurred in Azure. This includes a range of data, from Azure Resource Manager operational data to updates on Service Health events. The Activity Log was previously known as "Audit Logs" or "Operational Logs".

Using the Activity Log, you can determine the 'what, who, and when' for any write operation taken on the resources in your subscription. For example, who stopped a service. It provides an audit trail of the activities or operations performed on your resources by someone working on the Azure platform. You can also understand the status of the operation and other relevant properties.



This diagram shows many of the things you can do with the activity log including:

- Send data to Log Analytics for advanced search and alerts
- Query or manage events in the Portal, PowerShell, CLI, and REST API
- Stream information to Event Hub
- Archive data to a storage account
- Analyze data with Power BI

✓ The Activity Log differs from **Diagnostic Logs**[4]. Activity Logs provide data about the operations on a resource from the outside (the "control plane"). Diagnostics Logs are emitted by a resource and provide information about the operation of that resource (the "data plane").

For more information, you can see:

Monitor Subscription Activity with the Azure Activity Log - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs**

[4] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-of-diagnostic-logs

Categories in the Activity Log[5]

The Activity Log contains several categories of data. For full details on the schemata of these categories, **see this article**[6]. These include:

- **Administrative** - This category contains the record of all create, update, delete, and action operations performed through Resource Manager. Examples of the types of events you would see in this category include "create virtual machine" and "delete network security group" Every action taken by a user or application using Resource Manager is modeled as an operation on a particular resource type. If the operation type is Write, Delete, or Action, the records of both the start and success or fail of that operation are recorded in the Administrative category. The Administrative category also includes any changes to role-based access control in a subscription.
- **Service Health** - This category contains the record of any service health incidents that have occurred in Azure. An example of the type of event you would see in this category is "SQL Azure in East US is experiencing downtime." Service health events come in five varieties: Action Required, Assisted Recovery, Incident, Maintenance, Information, or Security, and only appear if you have a resource in the subscription that would be impacted by the event.
- **Resource Health** - This category contains the record of any resource health events that have occurred to your Azure resources. An example of the type of event you would see in this category is "Virtual Machine health status changed to unavailable." Resource health events can represent one of four health statuses: Available, Unavailable, Degraded, and Unknown. Additionally, resource health events can be categorized as being Platform Initiated or User Initiated.
- **Alert** - This category contains the record of all activations of Azure alerts. An example of the type of event you would see in this category is "CPU % on myVM has been over 80 for the past 5 minutes." A variety of Azure systems have an alerting concept -- you can define a rule of some sort and receive a notification when conditions match that rule. Each time a supported Azure alert type 'activates,' or the conditions are met to generate a notification, a record of the activation is also pushed to this category of the Activity Log.
- **Autoscale** - This category contains the record of any events related to the operation of the autoscale engine based on any autoscale settings you have defined in your subscription. An example of the type of event you would see in this category is "Autoscale scale up action failed." Using autoscale, you can automatically scale out or scale in the number of instances in a supported resource type based on time of day and/or load (metric) data using an autoscale setting. When the conditions are met to scale up or down, the start and succeeded or failed events are recorded in this category.
- **Recommendation** - This category contains recommendation events from Azure Advisor.
- **Security** - This category contains the record of any alerts generated by Azure Security Center. An example of the type of event you would see in this category is "Suspicious double extension file executed."
- **Policy** - This category does not contain any events; it is reserved for future use.

[5] https://docs.microsoft.com/en-us/azure/azure-monitor/platform/activity-logs-overview#categories-in-the-activity-log
[6] https://docs.microsoft.com/en-us/azure/azure-monitor/platform/activity-log-schema

Query the Activity Log in Azure

Query the Activity Log in the Azure portal[7]

Within the Azure portal, you can view your Activity Log in several places:

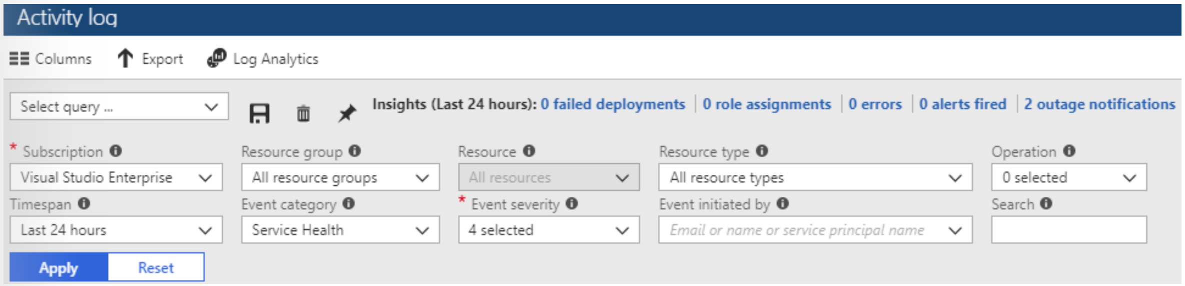
- The **Activity Log** that you can access by searching for the Activity Log under **All services** in the left-hand navigation pane.
- **Monitor** appears by default in the left-hand navigation pane. The Activity Log is one section of Azure Monitor.
- Most **resources**, for example, the configuration blade for a Virtual Machine. The Activity Log is a section on most resource blades, and clicking on it automatically filters the events to those related to that specific resource.

In the Azure portal, you can filter your Activity Log by these fields:

- Timespan - The start and end time for events.
- Category - The event category as described above.
- Subscription - One or more Azure subscription names.
- Resource group - One or more resource groups within those subscriptions.
- Resource (name) - The name of a specific resource.
- Resource type - The type of resource, for example, Microsoft.Compute/virtualmachines.
- Operation name - The name of an Azure Resource Manager operation, for example, Microsoft.SQL/servers/Write.
- Severity - The severity level of the event (Informational, Warning, Error, Critical).
- Event initiated by - The 'caller,' or user who performed the operation.
- Open search - This is an open text search box that searches for that string across all fields in all events.

✓ Once you have defined a set of filters, you can save it as a query that is persisted across sessions if you ever need to perform the same query with those filters applied again in the future. You can also pin a query to your Azure dashboard to always keep an eye on specific events.

For even more power, you can click the **Logs** icon, which displays your Activity Log data in the **Log Analytics Activity Log Analytics solution**[8]. The Activity Log blade offers a basic filter/browse experience on logs, but Log Analytics enables you to pivot, query, and visualize your data in more powerful ways.



Insights (Last 24 hours): 0 failed deployments | 0 role assignments | 0 errors | 0 alerts fired | 2 outage notifications

* Subscription	Resource group	Resource	Resource type	Operation
Visual Studio Enterprise	All resource groups	All resources	All resource types	0 selected
Timespan	Event category	* Event severity	Event initiated by	Search
Last 24 hours	Service Health	4 selected	Email or name or service principal name	

[7] https://docs.microsoft.com/en-us/azure/azure-monitor/platform/activity-logs-overview#query-the-activity-log-in-the-azure-portal
[8] https://docs.microsoft.com/en-us/azure/azure-monitor/platform/collect-activity-logs

Export the Activity Log with a Log Profile[9]

A **Log Profile** controls how your Activity Log is exported. Using a Log Profile, you can configure:

- Where the Activity Log should be sent (Storage Account or Event Hubs)
- Which event categories (Write, Delete, Action) should be sent. *The meaning of "category" in Log Profiles and Activity Log events is different. In the Log Profile, "Category" represents the operation type (Write, Delete, Action). In an Activity Log event, the "category" property represents the source or type of event (for example, Administration, ServiceHealth, Alert, and more).*
- Which regions (locations) should be exported. Make sure to include "global," as many events in the Activity Log are global events.
- How long the Activity Log should be retained in a Storage Account.
- A retention of zero days means logs are kept forever. Otherwise, the value can be any number of days between 1 and 2147483647.
  If retention policies are set but storing logs in a Storage Account is disabled (for example, if only Event Hubs or Log Analytics options are selected), the retention policies have no effect.
- Retention policies are applied per-day, so at the end of a day (UTC), logs from the day that is now beyond the retention policy are deleted. For example, if you had a retention policy of one day, at the beginning of the day today the logs from the day before yesterday would be deleted. The delete process begins at midnight UTC, but note that it can take up to 24 hours for the logs to be deleted from your storage account.

You can use a storage account or event hub namespace that is not in the same subscription as the one emitting logs. The user who configures the setting must have the appropriate RBAC access to both subscriptions.

For more information, you can see: **Query the Activity Log in the Azure portal** - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs#query-the-activity-log-in-the-azure-portal**[10]

Log Analytics Scenarios

One of the challenges with any broad data analytics solution is figuring out where you're going to see value for your organization. Out of all the things that are possible, what does your business need? What we hear from customers is that the following areas all have the potential to deliver significant business value:

Example 1 - Assessing updates

An important part of the daily routine for any IT administrator is assessing systems update requirements and planning patches. Accurate scheduling is critical, as it directly relates to SLAs to the business and can seriously impact business functions. In the past, you had to schedule an update with only limited knowledge of how long the patching would take. Operations Management Suite collects data from all customers performing patches and uses that data to provide an average patching time for specific missing updates. This use of "crowd-sourced" data is unique to cloud systems, and is a great example of how Log Analytics can help meet strict SLAs.

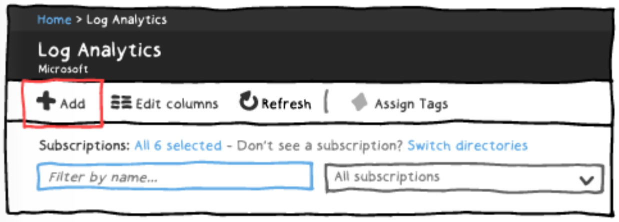
Example 2 - Change tracking

[9] https://docs.microsoft.com/en-us/azure/azure-monitor/platform/activity-logs-overview#export-the-activity-log-with-a-log-profile
[10] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs

Troubleshooting an operational incident is a complex process, requiring access to multiple data streams. With Operations Management Suite, you can easily perform analysis from multiple angles, using data from a wide variety of sources through a single interface for correlation of information. By tracking changes throughout the environment, Log Analytics helps to easily identify things like abnormal behavior from a specific account, users installing unapproved software, unexpected system reboots or shutdowns, evidence of security breaches, or specific problems in loosely coupled applications.

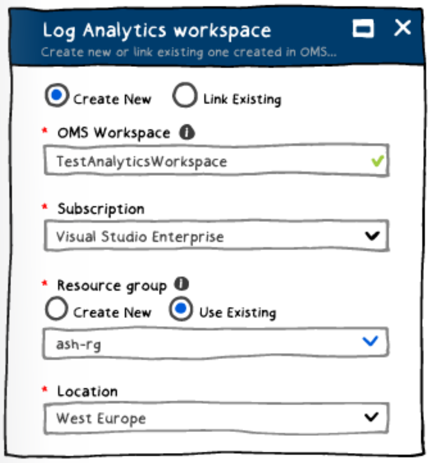
Create a Workspace

To get started with Log Analytics you need to add a workspace. In the Azure portal, click All services. In the list of resources, type Log Analytics. As you begin typing, the list filters based on your input. Select Log Analytics.



Subscriptions: All 6 selected - Don't see a subscription? Switch directories

You can them click Create and select your choices for the new workspace.



● Create New ○ Link Existing

* OMS Workspace: TestAnalyticsWorkspace

* Subscription: Visual Studio Enterprise

* Resource group: ○ Create New ● Use Existing

ash-rg

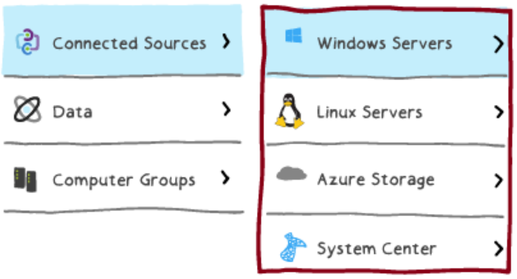
* Location: West Europe

- Provide a name for the new Log Analyics workspace, such as DefaultLAWorkspace.
- Select a Subscription from the drop-down list.
- For Resource Group, select an existing resource group that contains one or more Azure virtual machines.
- Select the Location your VMs are deployed to. See which regions **Log Analytics is available in**[11].
- The workspace will automatically use the Per GB pricing plan.

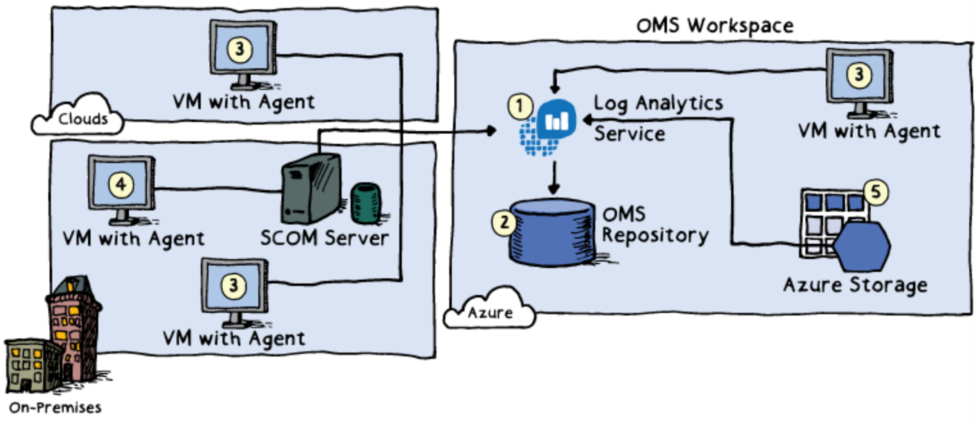
[11] https://azure.microsoft.com/regions/services/

Connected Sources

Connected sources are the computers and other resources that generate data collected by Log Analytics. This can include agents installed on **Windows**[12] and **Linux**[13] computers that connect directly or agents in a connected **System Center Operations Manager management group**[14] . Log Analytics can also collect data from **Azure storage**[15].



This following diagram shows how Connected Sources flow data to the Log Analytics service.



Ensure you can locate each of the following.

- The Log Analytics service (1) collects data and stores it in the OMS repository (2). The OMS Repository is hosted in Azure. Connected Sources provide information to the Log Analytics service.
- Computer agents (3) generate data to the Log Analytics service. These agents can run on Windows or Linux computers, virtual or physical computers, on-premises or cloud computers, and Azure or other cloud providers.
- A System Center Operations Manager (SCOM) management group can be connected to Log Analytics. SCOM agents (4) communicate with management servers which forward events and performance data to Log Analytics.
- An Azure storage account (5) can also collect Azure Diagnostics data from a worker role, web role, or virtual machine in Azure. This information can be sent to the Log Analytics service.

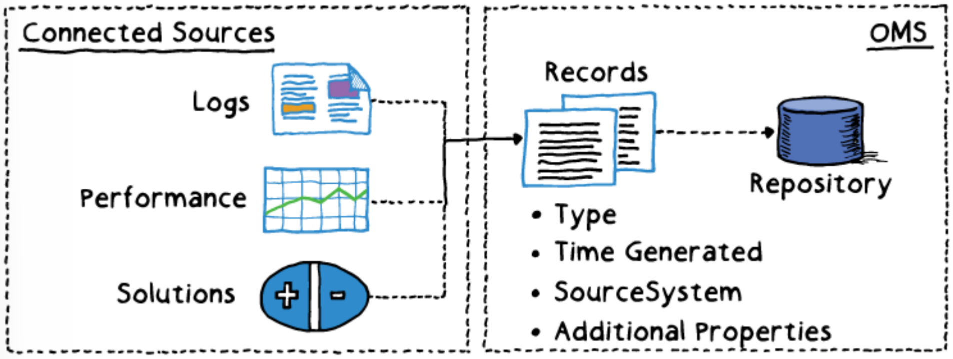
[12] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-windows-agents
[13] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-linux-agents
[14] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-om-agents
[15] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-azure-storage

For more information, you can see:

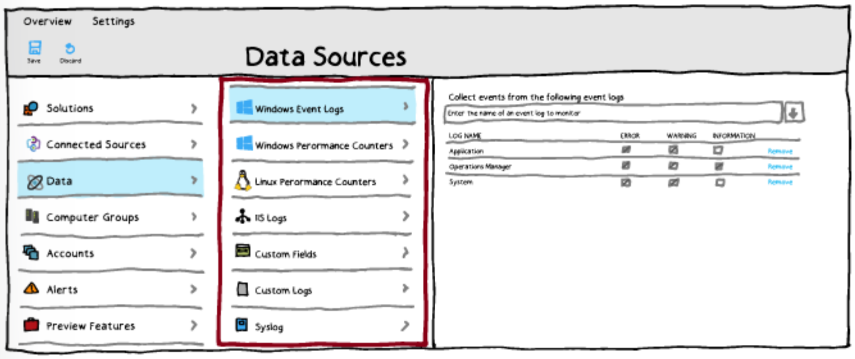
Connecting Computers to the Log Analytics Service - **https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-windows-agents#system-requirements-and-required-configuration**[16]

Data Sources

Data sources are the different kinds of data collected from each connected source. These can include events and performance data from Windows and Linux agents, in addition to sources such as IIS logs and custom text logs. You configure each data source that you want to collect, and the configuration is automatically delivered to each connected source.



When you configure the Log Analytics settings you can see the data sources that are available. Data sources include: Windows Event Logs, Windows Performance Counters, Linux Performance Counters, IIS Logs, Custom Fields, Custom Logs, and Syslog. Each data source has additional configuration options. For example, the Windows Event Log can be configured to forward Error, Warning, or Informational messages.



Collect events from the following event logs

Enter the name of an event log to monitor

LOG NAME	ERROR	WARNING	INFORMATION	
Application	☑	☑	☐	Remove
Operations Manager	☑	☑	☑	Remove
System	☑	☑	☐	Remove

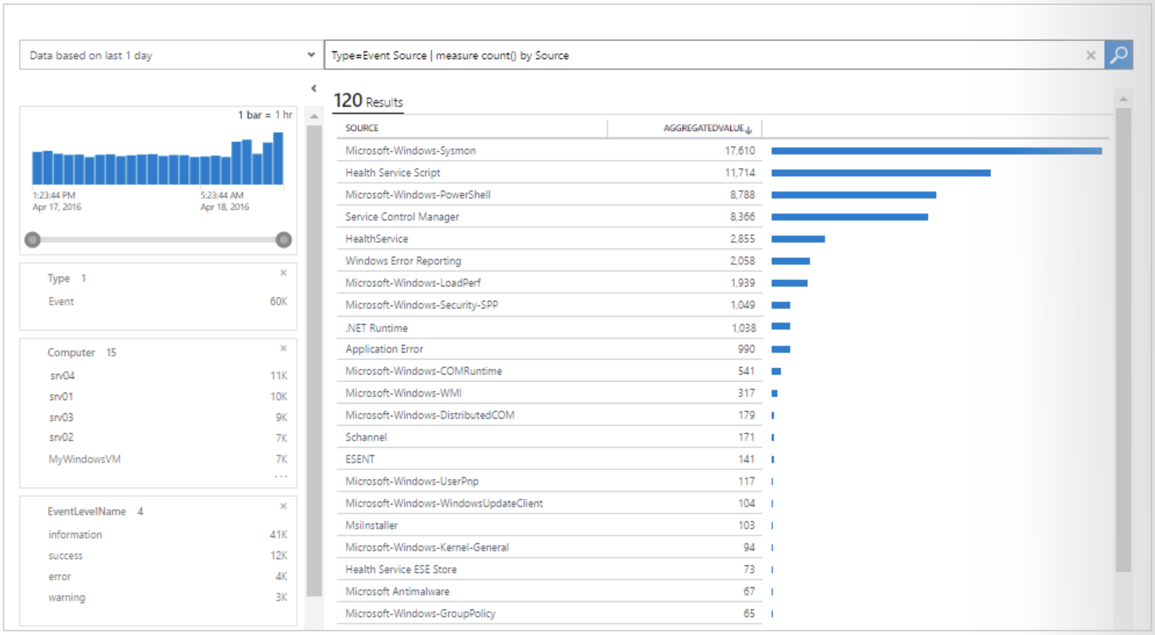
For more information, you can see:

Data Sources in Log Analytics - **https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-data-sources**

[16] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-windows-agents

Log Analytics Querying

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.



Type=Event Source | measure count() by Source

Type 1	
Event	60K

Computer 15	
srv04	11K
srv01	10K
srv03	9K
srv02	7K
MyWindowsVM	7K

EventLevelName 4	
information	41K
success	12K
error	4K
warning	3K

120 Results

SOURCE	AGGREGATEDVALUE
Microsoft-Windows-Sysmon	17,610
Health Service Script	11,714
Microsoft-Windows-PowerShell	8,788
Service Control Manager	8,366
HealthService	2,855
Windows Error Reporting	2,058
Microsoft-Windows-LoadPerf	1,939
Microsoft-Windows-Security-SPP	1,049
.NET Runtime	1,038
Application Error	990
Microsoft-Windows-COMRuntime	541
Microsoft-Windows-WMI	317
Microsoft-Windows-DistributedCOM	179
Schannel	171
ESENT	141
Microsoft-Windows-UserPnp	117
Microsoft-Windows-WindowsUpdateClient	104
MsiInstaller	103
Microsoft-Windows-Kernel-General	94
Health Service ESE Store	73
Microsoft Antimalware	67
Microsoft-Windows-GroupPolicy	65

To give a quick graphical view of the health of your overall environment, you can add visualizations for saved log searches to your dashboard. To analyze data outside of Log Analytics, you can export the data from the 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.

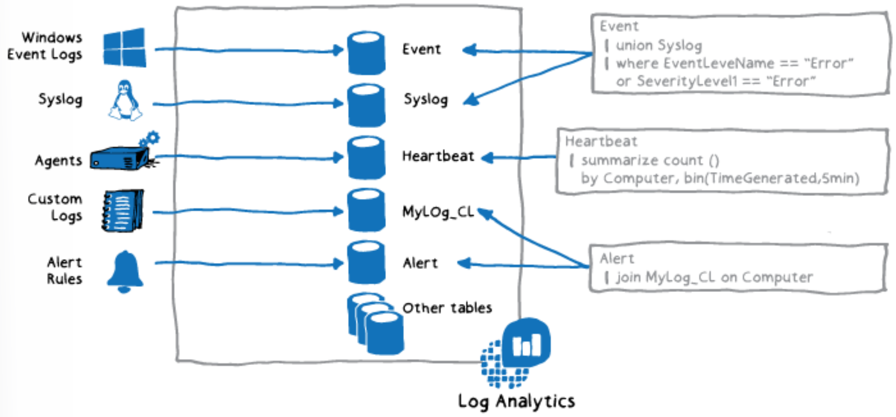
For more information, you can see:

Azure Log Analytics – meet our new query language - **https://azure.microsoft.com/en-us/blog/azure-log-analytics-meet-our-new-query-language-2/**

Querying Language Syntax

When you build a query, you start by determining which tables have the data that you're looking for. Each data source and solution stores its data in dedicated tables in the Log Analytics workspace. Documentation for each data source and solution includes the name of the data type that it creates and a description of each of its properties. Many queries will only require data from a single table, but others may use a variety of options to include data from multiple tables.

The main query tables are: Event, Syslog, Heartbeat, and Alert.



```
Event
| union Syslog
| where EventLeveName == "Error"
  or SeverityLevel1 == "Error"
```

```
Heartbeat
| summarize count ()
  by Computer, bin(TimeGenerated,5min)
```

```
Alert
| join MyLog_CL on Computer
```

The basic structure of a query is a source table followed by a series of operators separated by a pipe character |. You can chain together multiple operators to refine the data and perform advanced functions. For example, this query returns a count of the top 10 errors in the Event log during the last day. The results are in descending order.

```
Event
| where (EventLevelName == "Error")
| where (TimeGenerated > ago(1days))
| summarize ErrorCount = count() by Computer
| top 10 by ErrorCount desc
```

Some common operators are:

- **count** - Returns the number of records in the input record set.

  ```
  StormEvents | count
  ```

- **limit** - Return up to the specified number of rows.

  ```
  T | limit 5
  ```

- **summarize** - Produces a table that aggregates the content of the input table.

  ```
  T | summarize count(), avg(price) by fruit, supplier
  ```

- **top** - Returns the first N records sorted by the specified columns.

  ```
  T | top 5 by Name desc nulls last
  ```

- **where** - Filters a table to the subset of rows that satisfy a predicate.

  ```
  T | where fruit=="apple"
  ```

For more information, you can see:

Azure Monitor log queries - **https://docs.microsoft.com/en-us/azure/azure-monitor/log-query/query-language**

Demonstration - Log Analytics

In this demonstration, you will work with the Log Analytics query language.

Access the demonstration environment

1. Access the **Log Analytics Querying Demonstration**[17] page.
2. This page provides a live demonstration workspace where you can run and test queries.

Use the Query Explorer

1. Select **Query Explorer** (top right).
2. Expand **Favorites** and then select **All Syslog records with errors**.
3. Notice the query is added to the editing pane. Notice the structure of the query.
4. **Run** the query. Explore the records returned.
5. As you have time experiment with other **Favorites** and also **Saved Queries**.

✓ Is there a particular query you are interested in?

Practice: Visualize Data

Log Analytics dashboards can visualize all your saved log searches, giving you the ability to find, correlate, and share IT operational data in the organization. This practice covers creating a log search that will be used to support a shared dashboard that will be accessed by your IT operations support team.

Take a few minutes to try the **Create and share dashboards of Log Analytics data**[18] tutorial. You learn how to:

- Create a shared dashboard in the Azure portal.
- Visualize a performance log search.
- Add a log search to a shared dashboard.
- Customize a tile in a shared dashboard.

✓ In this tutorial, you learned how to create a dashboard in the Azure portal and add a log search to it. In the next tutorial you will learn the different responses you can implement based on log search results.

Practice: Alert on Data

[17] https://portal.loganalytics.io/demo

[18] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-tutorial-dashboards

Azure Alerts automatically runs specified log queries at regular intervals. If the results of the log query match your criteria, then an alert record is created.

Take a few minutes to try the **Respond to events with Azure Monitor Alerts**[19] tutorial. You learn how to:

- Create an alert rule.
- Configure an Action Group to send an e-mail notification.

✓ In this tutorial, you learned to create an alert based on your Log Analytics workspace and then defined a custom log search. You then activated your alert and created an Action Group to send an email notification each time the alert is triggered.

Practice: Collect and Analyze Data

This is a two part practice. In the first practice, you will collect performance data from virtual machines. In the second practice, you will create and edit queries to analyze the data.

Part 1

Take a few minutes to try the **Collect data about Azure Virtual Machines**[20] QuickStart. In this QuickStart, you learn how to:

- Create a workspace.
- Enable Log Analytics on virtual machines.
- Collect event and performance data.
- View the data collected.

Part 2

Take a few minutes to try the **View or analyze data collected with Log Analytics log search**[21] tutorial. In this tutorial, you learn how to:

- Perform a simple search of event data and use features to modify and filter the results.
- Learn how to work with performance data.

✓ In the next tutorial you will learn how to visualize the data by creating a dashboard.

For more information, you can see:

Writing a query - **https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-log-search#writing-a-query**[22]

[19] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-tutorial-response
[20] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-quick-collect-azurevm
[21] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-tutorial-viewdata
[22] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-log-search

Practice: Log Analytics Queries

Take a few minutes to access the **Log Analytics Querying Demonstration**[23] page. This page provides a live demonstration workspace where you can run and test queries. Some of the testing queries are:

- See the volume of data collected in the last 24 hours in intervals of 30 minutes.
- Chart the distribution of billable data by type, over the last 24 hours.
- Find out which computers were alive in the past 2 days but haven't sent any data in the last 6 hours.

✓ The reference link has additional queries you can try. Is there a specific query that will help with your day to day tasks?

For more information, you can see:

Getting Started with the Analytics Portal - **https://portal.loganalytics.io/demo#/discover/home**[24]

[23] https://portal.loganalytics.io/demo

[24] https://portal.loganalytics.io/demo

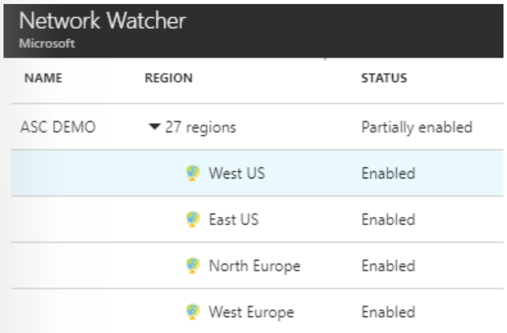
Network Watcher

Network Watcher

Azure Network Watcher provides tools to **monitor**, **diagnose**, view **metrics**, and enable or disable **logs** for resources in an Azure virtual network.

- **Automate remote network monitoring with packet capture.** Monitor and diagnose networking issues without logging in to your virtual machines (VMs) using Network Watcher. Trigger packet capture by setting alerts, and gain access to real-time performance information at the packet level. When you see an issue, you can investigate in detail for better diagnoses.
- **Gain insight into your network traffic using flow logs**. Build a deeper understanding of your network traffic pattern using Network Security Group flow logs. Information provided by flow logs helps you gather data for compliance, auditing and monitoring your network security profile.
- **Diagnose VPN connectivity issues**. Network Watcher provides you the ability to diagnose your most common VPN Gateway and Connections issues. Allowing you, not only, to identify the issue but also to use the detailed logs created to help further investigate.

Network Watcher is a regional service that enables you to monitor and diagnose conditions at a network scenario level.



Network Watcher
Microsoft

NAME	REGION	STATUS
ASC DEMO	▾ 27 regions	Partially enabled
	West US	Enabled
	East US	Enabled
	North Europe	Enabled
	West Europe	Enabled

For more information, you can see:

Network Watcher - **https://azure.microsoft.com/en-us/services/network-watcher/**

Monitoring and Visualization

Connection monitor

Connection monitor is a feature of Network Watcher that can monitor communication between a virtual machine and an endpoint. The connection monitor capability monitors communication at a regular interval and informs you of reachability, latency, and network topology changes between the VM and the endpoint.

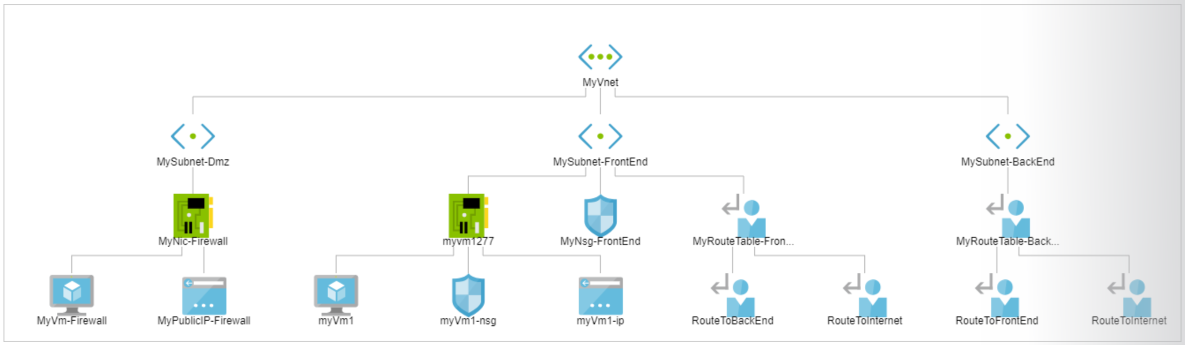
For example, you might have a web server VM that communicates with a database server VM. Someone in your organization may, unknown to you, apply a custom route or network security rule to the web server or database server VM or subnet.
If an endpoint becomes unreachable, connection troubleshoot informs you of the reason. Potential reasons might be DNS name resolution problem, the CPU, memory, or firewall within the operating system of a VM, or the hop type of a custom route, or security rule for the VM or subnet of the outbound connection. Connection monitor also provides the minimum, average, and maximum latency observed over time.

Network performance monitor

Network performance monitor is a cloud-based hybrid network monitoring solution that helps you monitor network performance between various points in your network infrastructure. It also helps you monitor network connectivity to service and application endpoints and monitor the performance of Azure ExpressRoute. Network performance monitor detects network issues like traffic blackholing, routing errors, and issues that conventional network monitoring methods aren't able to detect. The solution generates alerts and notifies you when a threshold is breached for a network link. It also ensures timely detection of network performance issues and localizes the source of the problem to a particular network segment or device.

Topology

Network Watcher's Topology capability enables you to generate a visual diagram of the resources in a virtual network, and the relationships between the resources. The following picture shows an example topology diagram for a virtual network that has three subnets, two VMs, network interfaces, public IP addresses, network security groups, route tables, and the relationships between the resources:



✓ To use Network Watcher capabilities, the account you log into Azure with, must be assigned to the Owner, Contributor, or Network contributor built-in roles, or assigned to a custom role. A custom role can be given permissions to read, write, and delete the Network Watcher.

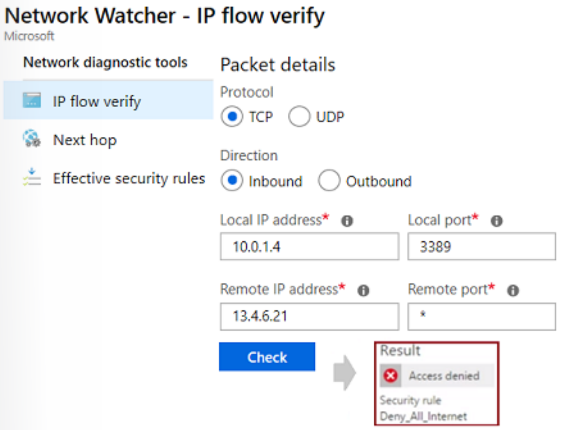
Diagnostics - IP Flow Verify

Verify IP Flow Purpose: Quickly diagnose connectivity issues from or to the internet and from or to the on-premises environment. For example, confirming if a security rule is blocking ingress or egress traffic to or from a virtual machine.

Example

When you deploy a VM, Azure applies several default security rules to the VM that allow or deny traffic to or from the VM. You might override Azure's default rules or create additional rules. At some point, a VM may become unable to communicate with other resources, because of a security rule.

The IP flow verify capability enables you to specify a source and destination IPv4 address, port, protocol (TCP or UDP), and traffic direction (inbound or outbound). IP flow verify then tests the communication and informs you if the connection succeeds or fails. If the connection fails, IP flow verify tells you which security rule allowed or denied the communication, so that you can resolve the problem.


Network Watcher - IP flow verify

Microsoft

Network diagnostic tools

- IP flow verify
- Next hop
- Effective security rules

Packet details

Protocol

● TCP ○ UDP

Direction

● Inbound ○ Outbound

Local IP address*: 10.0.1.4

Local port*: 3389

Remote IP address*: 13.4.6.21

Remote port*: *

Check

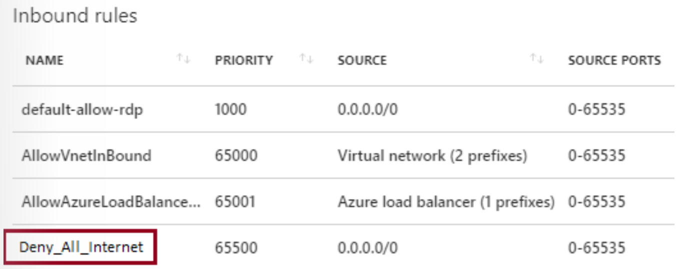
Result

Access denied

Security rule

Deny_All_Internet

If IP flow verify does not return the expected behavior you can investigate the security rule that was involved to determine what is going wrong and make an adjustment.


Inbound rules

NAME	PRIORITY	SOURCE	SOURCE PORTS
default-allow-rdp	1000	0.0.0.0/0	0-65535
AllowVnetInBound	65000	Virtual network (2 prefixes)	0-65535
AllowAzureLoadBalance...	65001	Azure load balancer (1 prefixes)	0-65535
Deny_All_Internet	65500	0.0.0.0/0	0-65535

✓ IP flow verify is ideal for making sure security rules are being correctly applied. When used for troubleshooting, if IP flow verify doesn't show a problem, you will need to explore other areas such as firewall restrictions.

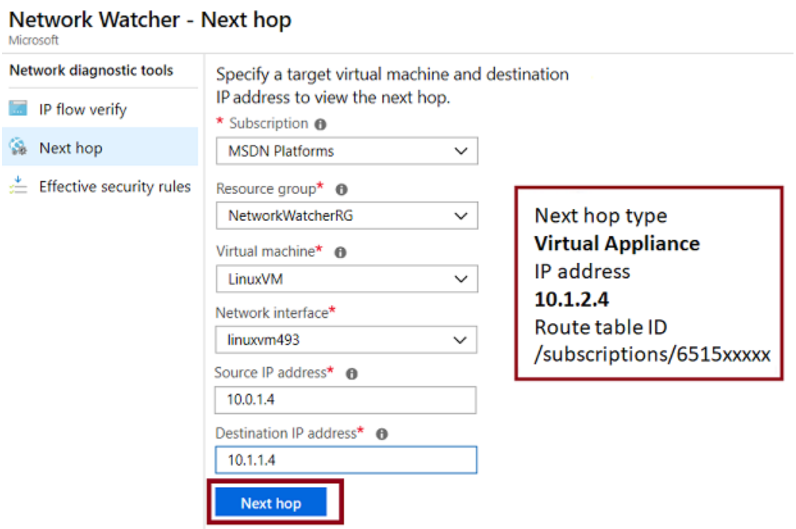
Diagnostics - Next Hop

Next Hop Purpose: To determine if traffic is being directed to the intended destination by showing the next hop. This will help determine if networking routing is correctly configured.

When you create a virtual network, Azure creates several default outbound routes for network traffic. The outbound traffic from all resources, such as VMs, deployed in a virtual network, are routed based on Azure's default routes. You might override Azure's default routes or create additional routes.

Example

You may find that a VM can no longer communicate with other resources because of a specific route. The next hop capability enables you to specify a source and destination IPv4 address. Next hop then tests the communication and informs you what type of next hop is used to route the traffic. You can then remove, change, or add a route, to resolve a routing problem.



Network Watcher - Next hop
Microsoft

Specify a target virtual machine and destination IP address to view the next hop.

- * Subscription: MSDN Platforms
- Resource group*: NetworkWatcherRG
- Virtual machine*: LinuxVM
- Network interface*: linuxvm493
- Source IP address*: 10.0.1.4
- Destination IP address*: 10.1.1.4

Next hop type
Virtual Appliance
IP address
10.1.2.4
Route table ID
/subscriptions/6515xxxxx

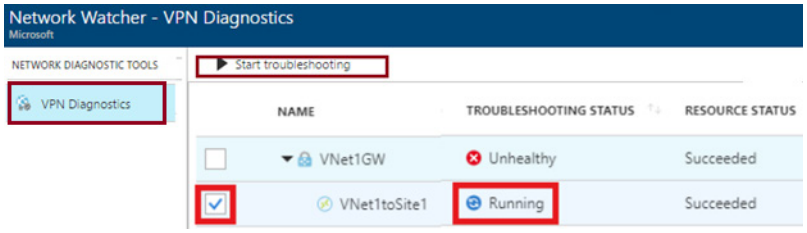
Next hop also returns the route table associated with the next hop. If the route is defined as a user-defined route, that route is returned. Otherwise, next hop returns System Route. Depending on your situation the next hop could be Internet, Virtual Appliance, Virtual Network Gateway, VNet Local, VNet Peering, or None. None lets you know that while there may be a valid system route to the destination, there is no next hop to route the traffic to the destination.

Diagnostics - VPN Diagnostics

VPN Diagnostics Purpose: Troubleshoot gateways and connections.

Example

Virtual Network Gateways provide connectivity between on-premises resources and other virtual networks within Azure. Monitoring gateways and their connections are critical to ensuring communication is working as expected. VPN diagnostics can troubleshoot the health of the gateway, or connection, and provide detailed logging. The request is a long running transaction and results are returned once the diagnosis is complete.



Network Watcher - VPN Diagnostics
Microsoft

	NAME	TROUBLESHOOTING STATUS	RESOURCE STATUS
□	▾ VNet1GW	✖ Unhealthy	Succeeded
☑	VNet1toSite1	Running	Succeeded

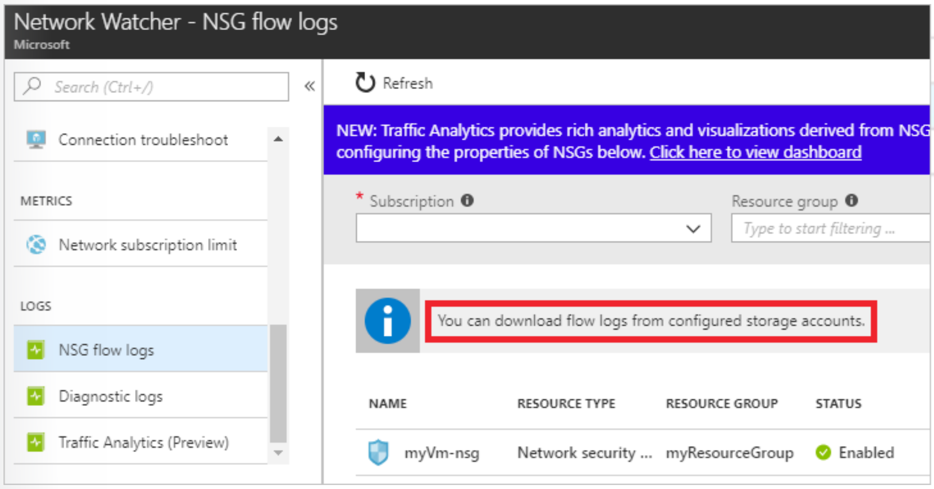
VPN Diagnostics returns a wealth of information. Summary information is available in the portal and more detailed information is provided in log files. The log files are stored in a storage account and include things like connection statistics, CPU and memory information, IKE security errors, packet drops, and buffers and events.

✓ You can select multiple gateways or connections to troubleshoot simultaneously or you can focus on an individual component.

NSG Flow Logs

NSG flow logs allows you to view information about ingress and egress IP traffic through an NSG. Flow logs are written in JSON format and show outbound and inbound flows on a per rule basis. The JSON format can be visually displayed in Power BI or third-party tools like Kibana.

You can download flow logs from configured storage accounts. Navigate to the storage container and look for the PT1H.JSON file.



NEW: Traffic Analytics provides rich analytics and visualizations derived from NSG configuring the properties of NSGs below. Click here to view dashboard

You can download flow logs from configured storage accounts.

NAME	RESOURCE TYPE	RESOURCE GROUP	STATUS
myVm-nsg	Network security ...	myResourceGroup	✓ Enabled

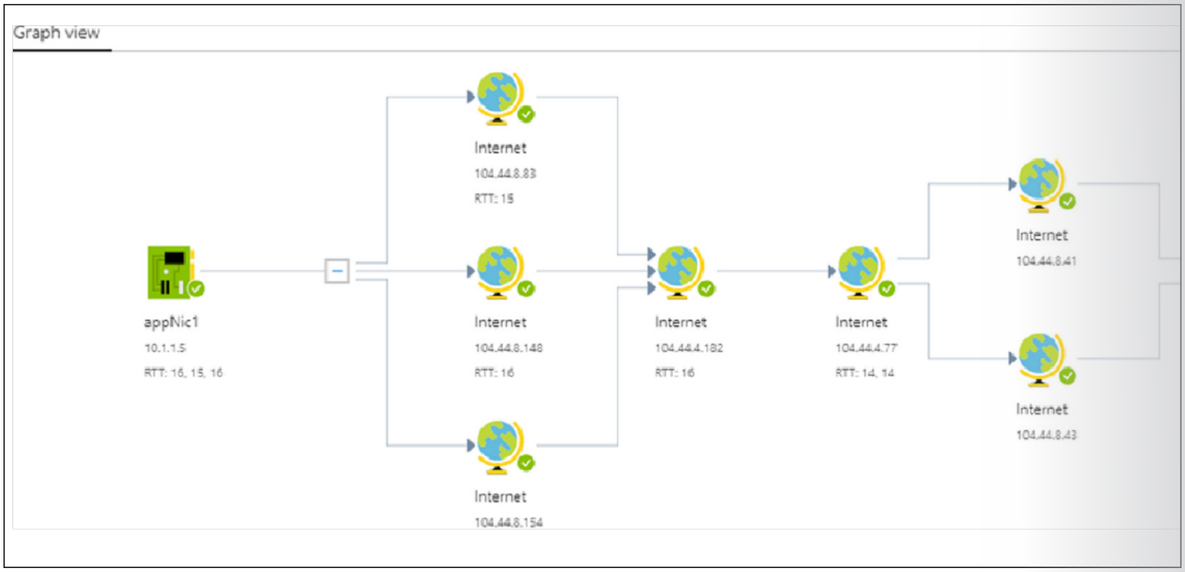
✓ These capabilities can be used in security compliance and auditing. You can define a prescriptive set of security rules as a model for security governance in your organization. A periodic compliance audit can be implemented in a programmatic way by comparing the prescriptive rules with the effective rules for each of the VMs in your network. Explore this feature with NSG Auditing practice.

Connection Troubleshoot

Azure Network Watcher Connection Troubleshoot is a more recent addition to the Network Watcher suite of networking tools and capabilities. Connection Troubleshoot enables you to troubleshoot network performance and connectivity issues in Azure.

This adds to the current capabilities of Network Watcher in providing even more ways for you troubleshoot networking operations. You can use Connection Troubleshoot to:

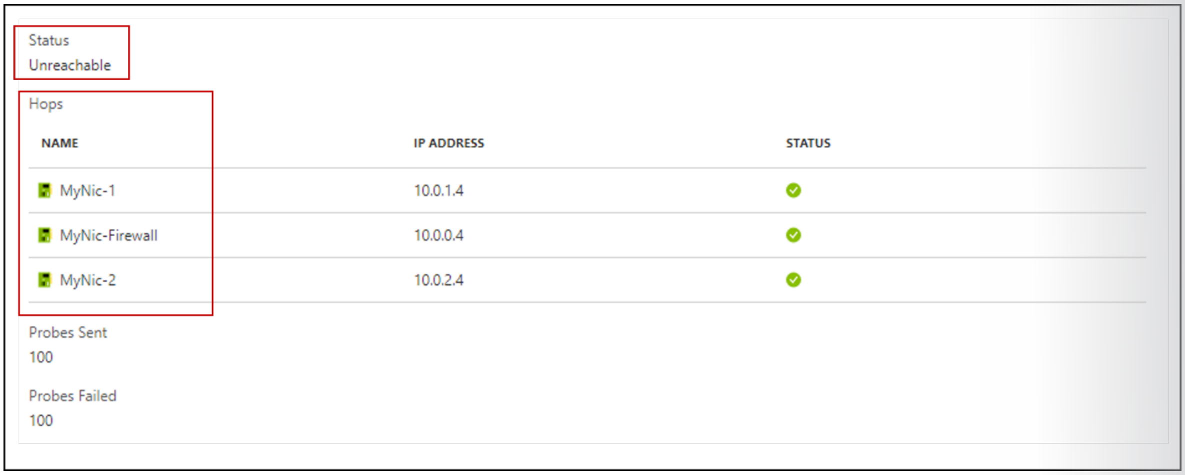
- Check connectivity between source (VM) and destination (VM, URI, FQDN, IP Address).
- Identify configuration issues that are impacting reachability.
- Provide all possible hop by hop paths from the source to destination.
- Hop by hop latency.
- Latency - min, max, and average between source and destination.
- View the number of packets dropped during the connection troubleshoot check.
- Connection Troubleshoot can also provide a topology (graphical) view from your source to destination, as shown in the following illustration.



Example Scenario

Connection Troubleshoot supports all networking scenarios where the source and destination is an Azure VM, FQDN, URI or an IPv4 Address.

In this example, an instance of Network Watcher is configured to check connectivity to a destination VM over port 80. When you open Connection Troubleshoot and select the VM and port to test, once you click Check, connectivity between the VMs on the port specified is checked. In this case, the destination VM is unreachable, and a listing of hops is shown.



Status
Unreachable

Hops

NAME	IP ADDRESS	STATUS
MyNic-1	10.0.1.4	✓
MyNic-Firewall	10.0.0.4	✓
MyNic-2	10.0.2.4	✓

Probes Sent
100

Probes Failed
100

Further examples of different supported network troubleshooting scenarios include:

- Checking the connectivity and latency to a remote endpoint, such as for websites and storage endpoints.
- Connectivity between an Azure VM and an Azure resource like Azure SQL server, where all Azure traffic is tunneled through an on-premises network.
- Connectivity between VMs in different VNets connected using VNet peering.

For more information, see:

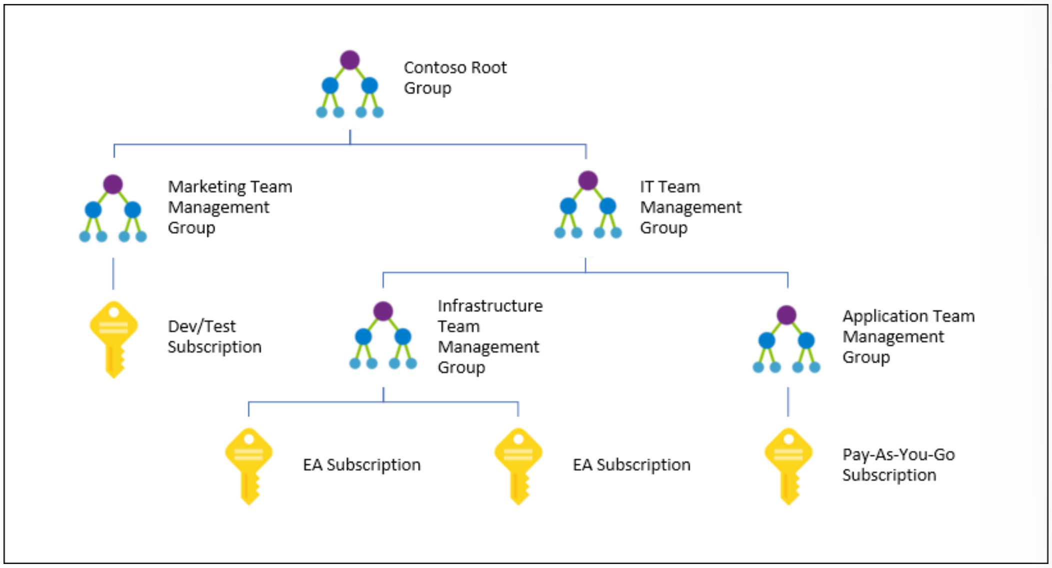
Troubleshoot connections with Azure Network Watcher using the Azure portal - **https://docs.microsoft.com/en-us/azure/network-watcher/network-watcher-connectivity-portal**

Subscriptions and Accounts

Management Groups

If your organization has several subscriptions, you may need a way to efficiently manage access, policies, and compliance for those subscriptions. Azure management groups provide a level of scope above subscriptions. You organize subscriptions into containers called "management groups" and apply your governance conditions to the management groups. Management group enable:

- Organizational alignment for your Azure subscriptions through custom hierarchies and grouping.
- Targeting of policies and spend budgets across subscriptions and inheritance down the hierarchies.
- Compliance and cost reporting by organization (business/teams).


Contoso Root Group
Marketing Team Management Group
IT Team Management Group
Dev/Test Subscription
Infrastructure Team Management Group
Application Team Management Group
EA Subscription
EA Subscription
Pay-As-You-Go Subscription

All subscriptions within a management group automatically inherit the conditions applied to the management group. For example, you can apply policies to a management group that limits the regions available for virtual machine (VM) creation. This policy would be applied to all management groups, subscriptions, and resources under that management group by only allowing VMs to be created in that region.

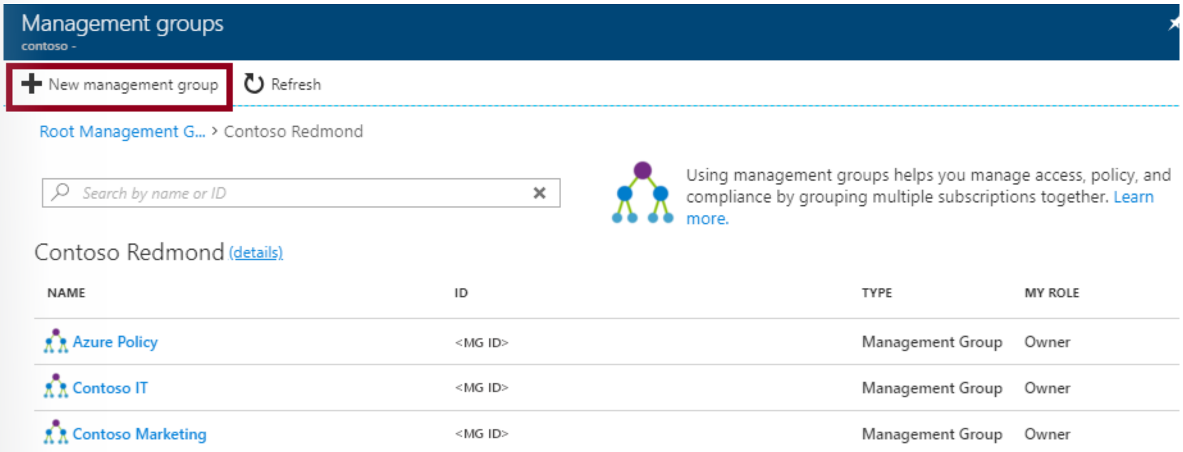
✓ Management groups is a relatively new concept in Azure.

For more information, you can see:

Organize your resources with Azure management groups - **https://docs.microsoft.com/en-us/azure/azure-resource-manager/management-groups-overview**

Creating Management Groups

You can create the management group by using the portal, PowerShell, or Azure CLI. Currently, you can't use Resource Manager templates to create management groups.



Using management groups helps you manage access, policy, and compliance by grouping multiple subscriptions together. Learn more.

Contoso Redmond (details)

NAME	ID	TYPE	MY ROLE
Azure Policy	<MG ID>	Management Group	Owner
Contoso IT	<MG ID>	Management Group	Owner
Contoso Marketing	<MG ID>	Management Group	Owner

- The **Management Group ID** is the directory unique identifier that is used to submit commands on this management group. This identifier is not editable after creation as it is used throughout the Azure system to identify this group.
- **The Display Name** field is the name that is displayed within the Azure portal. A separate display name is an optional field when creating the management group and can be changed at any time.

Within PowerShell, use the **New-AzManagementGroup** cmdlet:

```
New-AzManagementGroup -GroupName 'Contoso'
```

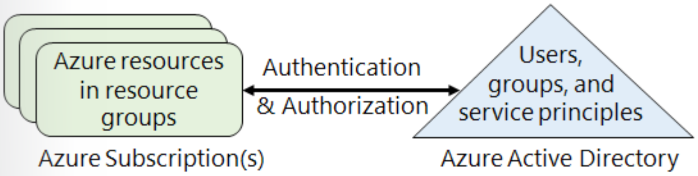
To show a different name for the management group within the Azure portal, add the DisplayName parameter with the desired string:

```
New-AzManagementGroup -GroupName 'Contoso' -DisplayName 'Contoso Develop-
ment'
```

Azure Subscriptions

An Azure subscription is a logical unit of Azure services that is linked to an Azure account. Billing for Azure services is done on a per-subscription basis. If your account is the only account associated with a subscription, then you are responsible for billing.

Subscriptions help you organize access to cloud service resources. They also help you control how resource usage is reported, billed, and paid for. Each subscription can have a different billing and payment setup, so you can have different subscriptions and different plans by department, project, regional office, and so on. Every cloud service belongs to a subscription, and the subscription ID may be required for programmatic operations.


Azure resources in resource groups
Authentication & Authorization
Users, groups, and service principles
Azure Subscription(s)
Azure Active Directory

Azure accounts

Subscriptions have accounts. An Azure account is simply an identity in Azure Active Directory (Azure AD) or in a directory that is trusted by Azure AD, such as a work or school organization. If you don't belong to one of these organizations, you can sign up for an Azure account by using your Microsoft Account, which is also trusted by Azure AD.

Getting access to resources

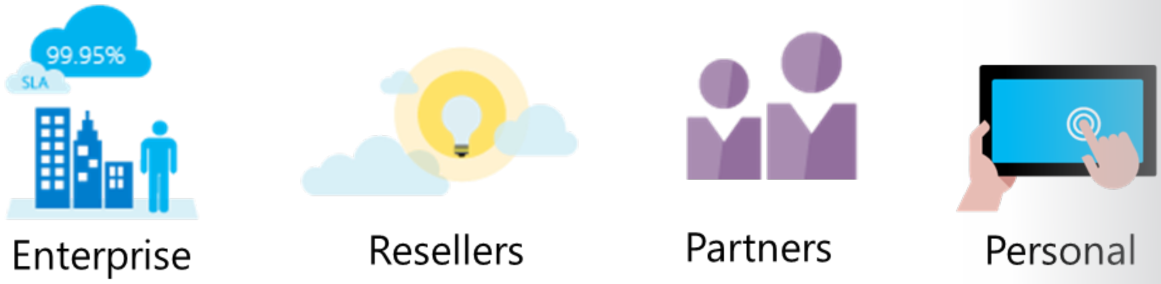
Every Azure subscription is associated with an Azure Active Directory. Users and services that access resources of the subscription first need to authenticate with Azure Active Directory.

Typically to grant a user access to your Azure resources, you would add them to the Azure AD directory associated with your subscription. The user will now have access to all the resources in your subscription. This is an all-or-nothing operation that may give that user access to more resources than you anticipated.

✓ Do you know how many subscriptions your organization has? Do you know how resources are organized into resource groups?

Getting a Subscription

There are several ways to get an Azure subscription: Enterprise agreements, Microsoft resellers, Microsoft partners, and a personal free account.



Enterprise agreements

Any **Enterprise Agreement**[25] customer can add Azure to their agreement by making an upfront monetary commitment to Azure. That commitment is consumed throughout the year by using any combination of the wide variety of cloud services Azure offers from its global datacenters. Enterprise agreements have a 99.95% monthly SLA.

Reseller

Buy Azure through the **Open Licensing program**[26], which provides a simple, flexible way to purchase cloud services from your Microsoft reseller. If you already purchased an Azure in Open license key, **activate a new subscription or add more credits now**[27].

Partners

Find a **Microsoft partner**[28] who can design and implement your Azure cloud solution. These partners have the business and technology expertise to recommend solutions that meet the unique needs of your business.

Personal free account

[25] https://azure.microsoft.com/en-us/pricing/enterprise-agreement/
[26] https://www.microsoft.com/en-us/licensing/licensing-programs/open-license.aspx
[27] https://azure.microsoft.com/en-us/offers/ms-azr-0111p/
[28] https://azure.microsoft.com/en-us/partners/directory/

With a **free trial account**[29] you can get started using Azure right away and you won't be charged until you choose to upgrade.

✓ Which subscription model are you most interested in?

For more information, you can see:

Solution providers - **https://www.microsoft.com/en-us/solution-providers/home**

Subscription Usage

Azure offers free and paid subscription options to suit different needs and requirements. The most commonly used subscriptions are:

- Free
- Pay-As-You-Go
- Enterprise Agreement
- Student

Azure free subscription

An Azure free subscription includes a $200 credit to spend on any service for the first 30 days, free access to the most popular Azure products for 12 months, and access to more than 25 products that are always free. This is an excellent way for new users to get started. To set up a free subscription, you need a phone number, a credit card, and a Microsoft account.

Note: Credit card information is used for identity verification only. You won't be charged for any services until you upgrade.

Azure Pay-As-You-Go subscription

A Pay-As-You-Go (PAYG) subscription charges you monthly for the services you used in that billing period. This subscription type is appropriate for a wide range of users, from individuals to small businesses, and many large organizations as well.

Azure Enterprise Agreement

An Enterprise Agreement provides flexibility to buy cloud services and software licenses under one agreement, with discounts for new licenses and Software Assurance. It's targeted at enterprise-scale organizations.

Azure for Students subscription

An Azure for Students subscription includes $100 in Azure credits to be used within the first 12 months plus select free services without requiring a credit card at sign-up. You must verify your student status through your organizational email address.

Subscription User Types

An Azure account determines how Azure usage is reported and who the Account Administrator is. Accounts and subscriptions are created in the Azure Account Center. The person who creates the account is the Account Administrator for all subscriptions created in that account. That person is also the default Service Administrator for the subscription.

Subscription User Types

[29] https://azure.microsoft.com/en-us/free/

There are three roles related to Azure accounts and subscriptions:

Administrative role	Limit	Summary
Account Administrator	1 per Azure account	Authorized to access the Account Center (create subscriptions, cancel subscriptions, change billing for a subscription, change Service Administrator). This role has full control over the subscription and is the account that is responsible for billing.
Service Administrator	1 per Azure subscription	Authorized to access Azure Management Portal for all subscriptions in the account. By default, same as the Account Administrator when a subscription is created. This role has control over all the services in the subscription.
Co-Administrator	200 per subscription (in addition to Service Administrator)	Same as Service Administrator but can't change the association of subscriptions to Azure directories.

Account administrator

The Account Administrator for a subscription is the only person with access to the Account Center. The Account Administrator does not have any other access to services in that subscription; they need to also be the Service Administrator or a Co-Administrator for that. For security reasons, the Account Administrator for a subscription can only be changed with a call to Azure support. The Account Administrator can easily reassign the Service Administrator for a subscription at the Account Center at any time.

Service Administrator and Co-Administrator

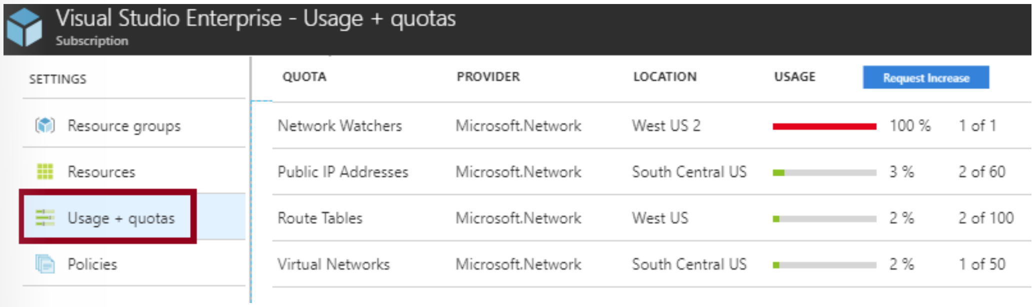
The Service Administrator is the first Co-Administrator for a subscription. Like other Co-Administrators, the Service Administrator has management access to cloud resources using the Azure Management Portal, as well as tools like Visual Studio, other SDKs, and command line tools like PowerShell. The Service Administrator can also add and remove other Co-Administrators.

Additionally, Co-Administrators can't delete the Service Administrator from the Azure Management Portal. Only the Account Administrator can change this assignment at the Account Center. The Service Administrator is the only user authorized to change a subscription's association with a directory in the Azure Management Portal.

✓ Account Administrators using a Microsoft account must log in every 2 years (or more frequently) to keep the account active. Inactive accounts are cancelled, and the related subscriptions removed. There are no login requirements if using a work or school account.

Check Resource Limits

Azure provides the ability to see the number of each network resource type that you've deployed in your subscription and what your subscription limits are. The ability to view resource usage against limits is helpful to track current usage, and plan for future use. In this example, there are two Public IP Addresses in South Central US and the limit is 60.



Visual Studio Enterprise - Usage + quotas
Subscription

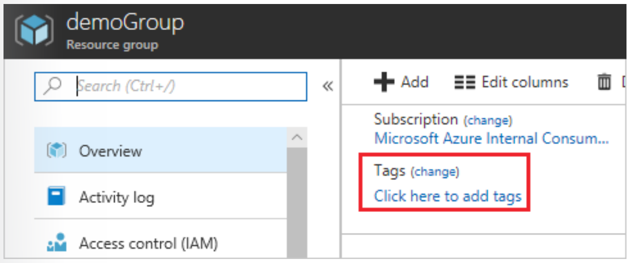
QUOTA	PROVIDER	LOCATION	USAGE		
Network Watchers	Microsoft.Network	West US 2	100 %	1 of 1
Public IP Addresses	Microsoft.Network	South Central US	3 %	2 of 60
Route Tables	Microsoft.Network	West US	2 %	2 of 100
Virtual Networks	Microsoft.Network	South Central US	2 %	1 of 50

The limits shown are the limits for your subscription. If you need to increase a default limit, there is a Request Increase link. You will complete and submit the support request. All resources have a maximum limit listed in Azure **limits**[30]. If your current limit is already at the maximum number, the limit can't be increased.

Resource Tags

You can apply tags to your Azure resources to logically organize them by categories. Each tag consists of a name and a value. For example, you can apply the name “Environment” and the value "Production" or “Development” to your resources. After creating your tags, you associate them with the appropriate resources.

With tags in place, you can retrieve all the resources in your subscription with that tag name and value. This means, you can retrieve related resources from different resource groups.

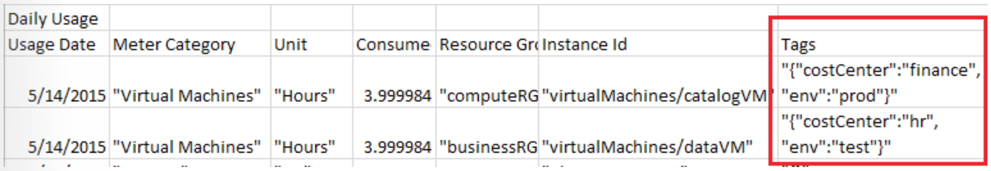


demoGroup
Resource group

Subscription (change)
Microsoft Azure Internal Consum...

Tags (change)
Click here to add tags

Perhaps one of the best uses of tags is to group billing data. When you download the usage CSV for services, the tags appear in the Tags column. For example, you could group virtual machines by cost center and production environment.



Daily Usage						
Usage Date	Meter Category	Unit	Consume	Resource Gr	Instance Id	Tags
5/14/2015	"Virtual Machines"	"Hours"	3.999984	"computeRG	"virtualMachines/catalogVM"	"{"costCenter":"finance", "env":"prod"}"
5/14/2015	"Virtual Machines"	"Hours"	3.999984	"businessRG	"virtualMachines/dataVM"	"{"costCenter":"hr", "env":"test"}"

There are a few things to consider about tagging:

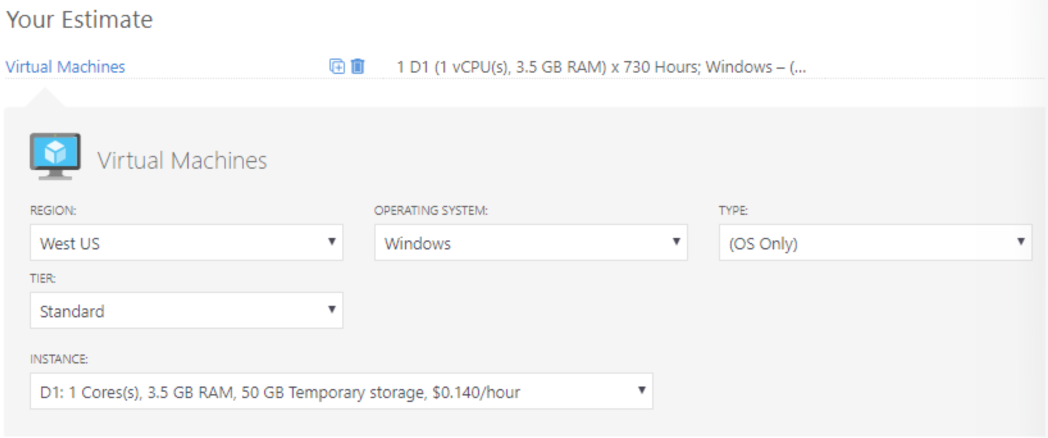
- Each resource or resource group can have a maximum of 15 tag name/value pairs.
- Tags applied to the resource group are not inherited by the resources in that resource group.

✓ If you need to create a lot of tags you will want to do that programmatically. You can use PowerShell or the CLI.

[30] https://docs.microsoft.com/en-us/azure/azure-subscription-service-limits?toc=%2fazure%2fnetworking%2ftoc.json

Billing

The **Pricing Calculator** provides estimates in all areas of Azure including compute, networking, storage, web, and databases.



Your Estimate

Virtual Machines — 1 D1 (1 vCPU(s), 3.5 GB RAM) x 730 Hours; Windows – (...

Virtual Machines

REGION: West US

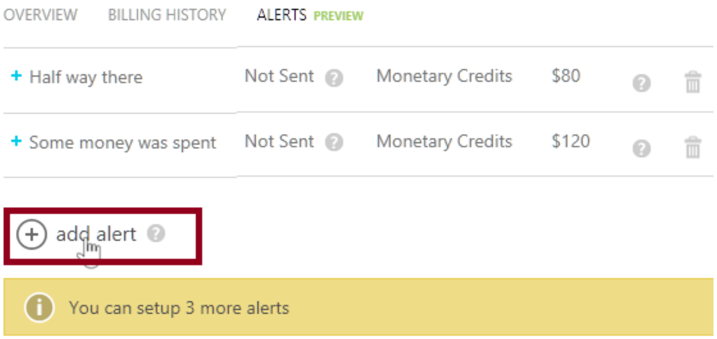
OPERATING SYSTEM: Windows

TYPE: (OS Only)

TIER: Standard

INSTANCE: D1: 1 Cores(s), 3.5 GB RAM, 50 GB Temporary storage, $0.140/hour

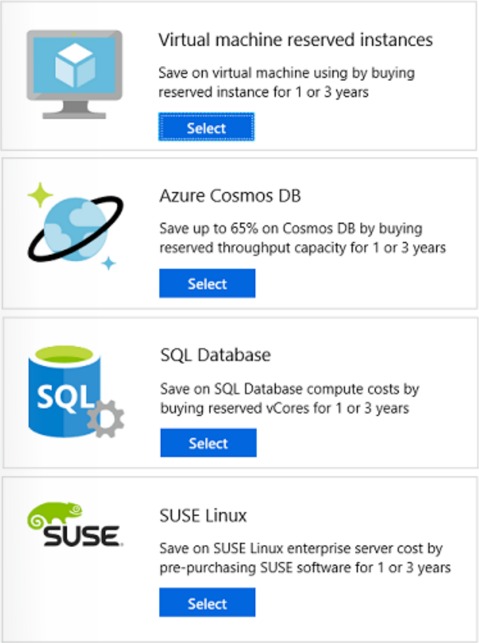
Billing Alerts help you monitor and manage billing activity for your Azure accounts. Billing alerts is available from the Account portal. You can set up a total of five billing alerts per subscription, with a different threshold and up to two email recipients for each alert. Monthly budgets are evaluated against spending every four hours. Budgets reset automatically at the end of a period.



+ Half way there	Not Sent	Monetary Credits	$80
+ Some money was spent	Not Sent	Monetary Credits	$120

You can setup 3 more alerts

Reservations helps you save money by pre-paying for one-year or three-years of virtual machine, SQL Database compute capacity, Azure Cosmos DB throughput, or other Azure resources. Pre-paying allows you to get a discount on the resources you use. Reservations can significantly reduce your virtual machine, SQL database compute, Azure Cosmos DB, or other resource costs up to 72% on pay-as-you-go prices. Reservations provide a billing discount and don't affect the runtime state of your resources.



Virtual machine reserved instances

Save on virtual machine using by buying reserved instance for 1 or 3 years

Azure Cosmos DB

Save up to 65% on Cosmos DB by buying reserved throughput capacity for 1 or 3 years

SQL Database

Save on SQL Database compute costs by buying reserved vCores for 1 or 3 years

SUSE Linux

Save on SUSE Linux enterprise server cost by pre-purchasing SUSE software for 1 or 3 years

Budgets help you plan for and drive organizational accountability. With budgets, you can account for the Azure services you consume or subscribe to during a specific period. They help you inform others about their spending to proactively manage costs, and to monitor how spending progresses over time. When the budget thresholds you've created are exceeded, only notifications are triggered. None of your resources are affected and your consumption isn't stopped. You can use budgets to compare and track spending as you analyze costs.

For more information, you can see:

Pricing Calculator - **https://azure.microsoft.com/en-us/pricing/calculator/**

Online Lab - Exploring Monitoring Capabilities in Azure

Lab Steps

Online Lab: Exploring Monitoring Capabilities in Azure

NOTE: For the most recent version of this online lab, see: **https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies**

Scenario

Adatum Corporation wants to explore monitoring capabilities in Azure

Objectives

After completing this lab, you will be able to:

- Deploy Azure VM scale sets
- Implement monitoring and alerting by using Azure Monitor

Lab Setup

Estimated Time: 45 minutes

User Name: **Student**

Password: **Pa55w.rd**

Exercise 1: Deploy Azure VM scale sets

The main tasks for this exercise are as follows:

1. Deploy an Azure VM scale set by using an Azure QuickStart template
2. Review autoscaling settings of the Azure VM scale set

Task 1: Deploy an Azure VM scale set by using an Azure QuickStart template

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. In the Azure portal, in the Microsoft Edge window, start a **PowerShell** session within the Cloud Shell.
3. If you are presented with the **You have no storage mounted** message, configure storage using the following settings:
   - Subsciption: the name of the target Azure subscription
   - Cloud Shell region: the name of the Azure region that is available in your subscription and which is closest to the lab location

- Resource group: the name of a new resource group **az3000100-LabRG**
- Storage account: a name of a new storage account
- File share: a name of a new file share

4. From the Cloud Shell pane, run the following to identify a unique DNS domain name (substitute the placeholder `<custom-label>` with any alphanumeric string starting with a letter and no longer than 9 characters, which is likely to be unique and the placeholder `<location>` with the name of the Azure region into which you intend to deploy resources in this lab):

```
Test-AzDnsAvailability -DomainNameLabel <custom-label> -Location <location>
```

5. Verify that the command returned **True**. If not, rerun the same command with a different value of the until the command returns **True**.
6. Note the value of the that resulted in the successful outcome. You will need it in the next task.
7. From the lab virtual machine, start Microsoft Edge and browse to the Azure QuickStart template that deploys autoscale demo app on Ubuntu 16.04 at **https://github.com/Azure/azure-quickstart-templates/tree/master/201-vmss-bottle-autoscale**.
8. Click **Deploy to Azure** and, when prompted, sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
9. In the Azure Portal, on the **Deploy VM Scale Set with Python Bottle server & AutoScale** blade, specify the following settings and initiate the deployment:
    - Subscription: the name of the target Azure subscription
    - Resource group: the name of a new resource group **az3000101-LabRG**
    - Location: the name of the Azure region that you referenced when running `Test-AzDnsAvailability` earlier in this task
    - Vm Sku: **Standard_D1_v2**
    - Vmss Name: the custom label you identified when running `Test-AzDnsAvailability` earlier in this task
    - Instance count: **1**
    - Admin Username: **student**
    - Admin Password: **Pa55w.rd1234**
10. Wait for the deployment to complete. This will take about 5 minutes.

Task 2: Review autoscaling settings of the Azure VM scale set

1. In Azure Portal, navigate to the blade representing the newly deployed **Azure VM scale set**.
2. From the VM scale set blade, navigate to the its **Scaling** blade.
3. Note that the Azure VM scale set is configured to scale dynamically based on a metric using the following criteria:
    - Scale out: increase instance count by 1 when average percentage of CPU > 60
    - Scale in: decrease instance count by 1 when average percentage of CPU < 30
    - Minimum number of instances: 1

- Maximum number of instances: 10

4. Modify the maximum number of instances to 3 and **save** your changes.

Result: After you completed this exercise, you have deployed an Azure VM scale set and reviewed its autoscaling settings.

Exercise 2: Implementing monitoring and alerting by using Azure Monitor

The main tasks for this exercise are as follows:

1. Create Azure VM scale set metrics-based alerts
2. Configure Azure VM scale set autoscaling-based notifications
3. Test Azure VM scale set monitoring and alerting.

Task 1: Create Azure VM scale set metrics-based alerts

1. In the Azure portal, navigate to the blade representing the newly deployed Azure VM scale set and, from there, switch to the **Monitoring - Metrics** blade.
2. On the **Monitoring - Metrics** blade, use the filter to display **Avg Percentage CPU** metric of the VM scale set resource you provisioned in the previous exercise of this lab.
3. Review the resulting chart and note the average percentage CPU within the last few minutes.
4. Navigate to the **Monitoring - Alerts** blade.
5. From the **Monitoring - Alerts** blade, navigate to the **New alert rule** blade.
6. In the **Resource** section, select the VM scale set you provisioned in the previous exercise of this lab.
7. In the **Condition** section, click **Add condition**, select the **Percentage CPU** metric, leave the dimension settings and condition type with their default values, set the condition to **Greater than**, set the time aggregation to **Average**, set the threshold to **60**, set the period (grain) to **Over the last 1 minutes**, set the frequency to **Every 1 minute** and click **done**.
8. In the **Action Groups** section, click **Create new**, set the action group name to **az30001 action group**, set short name: **az30001**, select the Azure subscription you used in the previous exercise, accept the default name of the resource group of **Default-ActivityLogAlerts (to be created)**, set the action name: **az30001-email**, and set the action type to **Email/SMS/Push/Voice**.
9. On the **Email/SMS/Push/Voice** blade, set an email address, a mobile phone number, or a phone number that you want to use to receive alerts generated by this rule.
10. In the **Alert Details** section, set the alert rule name to **Percentage CPU of the VM scale set is greater than 60 percent**, its description to **Percentage CPU of the VM scale set is greater than 60 percent**, its severity to **Sev 3**, and set enable rule upon creation to **Yes**.
11. **Note**: It can take up to 10 minutes for a metric alert rule to become active

Task 2: Configure Azure VM scale set autoscaling-based notifications

1. In the Azure portal, navigate to the blade representing the newly deployed Azure VM scale set and, from there, switch to the **Scaling** blade.

2. On the **Scaling setting** blade, click the **Notify** tab heading, configure the following settings, and save your changes:
   - Email administrators: enabled
   - Email co-administrators: disabled
   - Additional administrator emails(s): add an email address that you want to use to receive notifications about autoscaling events

Task 3: Test Azure VM scale set monitoring and alerting.

1. In the Azure portal, navigate to the blade representing the **Load balancer** set you deployed in the previous exercise of this lab.
2. Identify the value of the **Public IP address** assigned to the front end of the load balancer associated with the VM scale set.
3. From the lab computer, start Microsoft Edge and browse to *http://Public IP address*:9000 (where ***Public IP address*** is the IP address you identified in the previous step)
4. On the **Worker interface** page, click the **Start work** link.
5. Use the **CPU (average)** chart on the VM scale set blade to monitor changes to the CPU utilization.
6. **Note**: Alternatively, you can navigate back to the **Monitoring - Metrics** blade and use the filter to display **Avg Percentage CPU** metric of the VM scale set resource.
7. **Note**: You should receive an alert regarding increased CPU utilization within a couple of minutes
8. Switch to the **Instances** blade of the VM scale set in order to identify the number of its instances.
9. **Note**: Alternatively, you can navigate back to the **Scaling** blade, in the list of resources capable of autoscaling, click the name of the VM scale set, on the **Autoscale settings** blade, click **Run history**, and then review the list of autoscale events.
10. **Note**: Autoscaling should be triggered within a couple of minutes.
11. Switch to the Microsoft Edge window displaying worker instances page and click the **Stop work** link.
12. Monitor decrease in CPU utilization and scaling in events using the same methods that you used when scaling out the VM scale set.

Result: After you completed this exercise, you have implemented and tested monitoring and alerting by using Azure Monitor.

Exercise 3: Remove lab resources

Task 1: Delete resource group

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

```
az group list --query "[?starts_with(name,'az300010')].[name]" --output tsv
```

3. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

Task 2: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

   ```
   az group list --query "[?starts_with(name,'az300010')].[name]" --output tsv
   | %{az group delete -n $_ --no-wait -y}
   ```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

Result: After you completed this exercise, you removed the resources used in this lab.

Review Questions

Module 1 Review Questions

Azure Log Analytics Data Collector Connector for Logic Apps

You are an architect for a managed service provider. Your company monitors Azure environments for client companies

You need design a solution that allows your company to monitor all of the client's environments.

How can you perform Log Analytics across multiple Azure subscriptions?

Suggested Answer ↓

The basic strategy for using Log Analytics monitor multiple subscriptions is to configure Azure Activity Log to send events to an Event Hub in the client's subscription. A Logic App sends log data to your Log Analytics workspace.

Advantages of this approach include:

- Low latency since the Azure Activity Log is streamed into the Event Hub. The Logic App is triggered. It posts data to Log Analytics.
- Minimal code is required.
- No infrastructure is needed to implement the solution.

Troubleshooting Operational Incidents

You are an architect for a managed service provider. Your company monitors Azure environments for client companies.

A client reports that specific services are unavailable. The client suspects that a former employee may have turned off services before his accounts were shut down.

How can you analyze the incident? What tools should you consider using?

Suggested Answer ↓

Troubleshooting an operational incident is a complex process. It often requires access to multiple data streams. With Operations Management Suite, you can easily perform analysis from multiple angles, using data from a wide variety of sources through a single interface for correlation of information.

By tracking changes throughout the environment, Log Analytics helps you identify things like abnormal behavior from a specific account, users installing unapproved software, unexpected system reboots or shutdowns, evidence of security breaches, or specific problems in loosely coupled applications.

Azure Log Analytics and Azure Advisor

You are an architect for a managed service provider. Your company monitors Azure environments for client companies.

A client reports that specific Azure services are slow and unresponsive during periods of high activity.

You need to help the client understand the issues and recommend changes and additions to Azure resources.

What tools should you use?

Suggested Answer ↓

You should use Azure Monitor and Azure Advisor.

Azure Monitor provides the fastest metrics pipeline (5 minute down to 1 minute), so you should use it for time critical alerts and notifications. These metrics can be sent to Azure Log Analytics for trending and detailed analysis.

Once data is collected, Azure Advisor analyzes your resource configuration and usage telemetry and recommends solutions that can help you improve the cost effectiveness, performance, high availability, and security of your Azure resources. The Advisor cost recommendations page helps you optimize and reduce your overall Azure spend by identifying idle and underutilized resources.

Module 2 Module Implementing and Managing Storage

Azure Storage Accounts

Azure Storage

Azure Storage is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers a massively scalable object store for data objects, a file system service for the cloud, a messaging store for reliable messaging, and a NoSQL store. Azure Storage is:

- **Durable and highly available**. Redundancy ensures that your data is safe in the event of transient hardware failures. You can also opt to replicate data across datacenters or geographical regions for additional protection from local catastrophe or natural disaster. Data replicated in this way remains highly available in the event of an unexpected outage.
- **Secure**. All data written to Azure Storage is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
- **Scalable**. Azure Storage is designed to be massively scalable to meet the data storage and performance needs of today's applications.
- **Managed**. Microsoft Azure handles hardware maintenance, updates, and critical issues for you.
- **Accessible**. Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS. Microsoft provides SDKs for Azure Storage in a variety of languages – .NET, Java, Node.js, Python, PHP, Ruby, Go, and others – as well as a mature REST API. Azure Storage supports scripting in Azure PowerShell or Azure CLI. And the Azure portal and Azure Storage Explorer offer easy visual solutions for working with your data.

Azure Storage is a service that you can use to store files, messages, tables, and other types of information. You can use Azure storage on its own—for example as a file share—but it is often used by developers as a store for working data. Such stores can be used by websites, mobile apps, desktop applications, and many other types of custom solutions. Azure storage is also used by IaaS virtual machines, and PaaS cloud services. You can generally think of Azure storage in three categories.

- **Storage for Virtual Machines**. This includes disks and files. Disks are persistent block storage for Azure IaaS virtual machines. Files are fully managed file shares in the cloud.
- **Unstructured Data**. This includes Blobs and Data Lake Store. Blobs are highly scaleable, REST based cloud object store. Data Lake Store is Hadoop Distributed File System (HDFS) as a service.
- **Structured Data**. This includes Tables, Cosmos DB, and Azure SQL DB. Tables are a key/value, auto-scaling NoSQL store. Cosmos DB is a globally distributed database service. Azure SQL DB is a fully managed database-as-a-service built on SQL.

For more information, you can see:

Azure Storage - **https://azure.microsoft.com/en-us/services/storage/**

Azure Storage Services

Azure Storage services

Azure Storage includes these data services, each of which is accessed through a storage account.

- **Azure Blobs**: A massively scalable object store for text and binary data.
- **Azure Files**: Managed file shares for cloud or on-premises deployments.
- **Azure Queues**: A messaging store for reliable messaging between application components.
- **Azure Tables**: A NoSQL store for schemaless storage of structured data.

Blob storage

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data, such as text or binary data. Blob storage is ideal for:

- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Storing data for backup and restore, disaster recovery, and archiving.
- Storing data for analysis by an on-premises or Azure-hosted service.
  Objects in Blob storage can be accessed from anywhere in the world via HTTP or HTTPS. Users or client applications can access blobs via URLs, the Azure Storage REST API, Azure PowerShell, Azure CLI, or an Azure Storage client library. The storage client libraries are available for multiple languages, including .NET, Java, Node.js, Python, PHP, and Ruby.

Azure Files

Azure Files enables you to set up highly available network file shares that can be accessed by using the standard Server Message Block (SMB) protocol. That means that multiple VMs can share the same files with both read and write access. You can also read the files using the REST interface or the storage client libraries.

One thing that distinguishes Azure Files from files on a corporate file share is that you can access the files from anywhere in the world using a URL that points to the file and includes a shared access signature (SAS) token. You can generate SAS tokens; they allow specific access to a private asset for a specific amount of time.

File shares can be used for many common scenarios:

- Many on-premises applications use file shares. This feature makes it easier to migrate those applications that share data to Azure. If you mount the file share to the same drive letter that the on-premises application uses, the part of your application that accesses the file share should work with minimal, if any, changes.
- Configuration files can be stored on a file share and accessed from multiple VMs. Tools and utilities used by multiple developers in a group can be stored on a file share, ensuring that everybody can find them, and that they use the same version.
- Diagnostic logs, metrics, and crash dumps are just three examples of data that can be written to a file share and processed or analyzed later.

At this time, Active Directory-based authentication and access control lists (ACLs) are not supported, but they will be at some time in the future. The storage account credentials are used to provide authentication for access to the file share. This means anybody with the share mounted will have full read/write access to the share.

Queue storage

The Azure Queue service is used to store and retrieve messages. Queue messages can be up to 64 KB in size, and a queue can contain millions of messages. Queues are generally used to store lists of messages to be processed asynchronously.

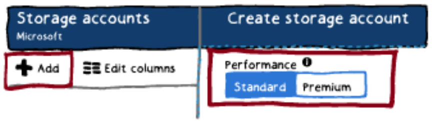
For example, say you want your customers to be able to upload pictures, and you want to create thumbnails for each picture. You could have your customer wait for you to create the thumbnails while uploading the pictures. An alternative would be to use a queue. When the customer finishes his upload, write a message to the queue. Then have an Azure Function retrieve the message from the queue and create the thumbnails. Each of the parts of this processing can be scaled separately, giving you more control when tuning it for your usage.

Table storage

Azure Table storage is now part of Azure Cosmos DB. To see Azure Table storage documentation, see the Azure Table Storage Overview. In addition to the existing Azure Table storage service, there is a new Azure Cosmos DB Table API offering that provides throughput-optimized tables, global distribution, and automatic secondary indexes. To learn more and try out the new premium experience, please check out Azure Cosmos DB Table API.

Standard and Premium Storage Accounts

As discussed previously, general purpose storage accounts have two tiers: Standard and Premium.



Standard storage accounts are backed by magnetic drives (HDD) and provide the lowest cost per GB. They are best for applications that require bulk storage or where data is accessed infrequently.

Premium storage accounts are backed by solid state drives (SSD) and offer consistent low-latency performance. They can only be used with Azure virtual machine disks and are best for I/O-intensive applications, like databases. Additionally, virtual machines that use Premium storage for all disks qualify for a 99.99% SLA, even when running outside an availability set.

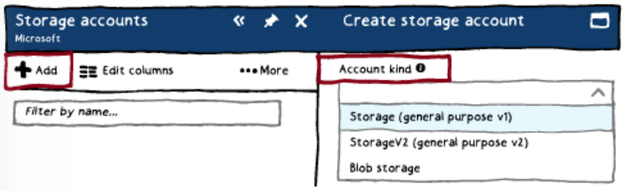
✓ It is not possible to convert a Standard storage account to Premium storage account or vice versa. You must create a new storage account with the desired type and copy data, if applicable, to a new storage account.

Storage Types

An Azure storage account provides a unique namespace in the cloud to store and access your data objects in Azure Storage. A storage account contains any blobs, files, queues, tables, and disks that you create under that account.

Storage Account Types (Kinds)

When you create a storage account you can choose from: Storage (general purpose v1), Storage V2 (general purpose v2), and Blob storage.



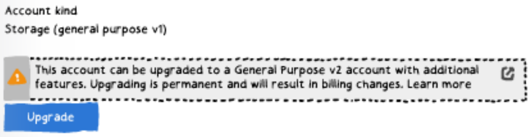
A **general-purpose storage account** gives you access to Azure Storage services such as tables, queues, files, blobs and Azure virtual machine disks under a single account. This type of storage account has two performance tiers:

- A **standard storage performance tier** which allows you to store tables, queues, files, blobs, and Azure virtual machine disks.
- A **premium storage performance tier** which currently only supports Azure virtual machine disks.

A **Blob storage account** is a specialized storage account for storing your unstructured data as blobs (objects) in Azure Storage. Blob storage has different tiers based on frequency of use:

- A **Hot** access tier which indicates that the objects in the storage account will be more frequently accessed.
- A **Cool** access tier which indicates that the objects in the storage account will be less frequently accessed.
- An **Archive** access tier which only applies to blob level storage in the general purpose v2 accounts.

✓ To take advantage of the new archive access tier and for the lowest price per gigabyte, it's recommended that you create new storage accounts as **general-purpose v2 accounts**. You can upgrade your GPv1 account to a GPv2 account using PowerShell or Azure CLI.



Account kind
Storage (general purpose v1)

This account can be upgraded to a General Purpose v2 account with additional features. Upgrading is permanent and will result in billing changes. Learn more

For more information, you can see:

Storage Account Overview - **https://docs.microsoft.com/en-us/azure/storage/common/storage-account-options#overview**[1]

Upgrade a storage account to GPv2 - **https://docs.microsoft.com/en-us/azure/storage/common/storage-account-options#upgrade-a-storage-account-to-gpv2**[2]

Storage Account Endpoints

Every object that you store in Azure Storage has a unique URL address. The storage account name forms the subdomain of that address. The combination of subdomain and domain name, which is specific to each service, forms an endpoint for your storage account.

For example, if your storage account is named *mystorageaccount*, then the default endpoints for your storage account are (NOTE: The links below are for sample purposes and not actual destinations):

- Blob service: http://mystorageaccount.blob.core.windows.net
- Table service: http://mystorageaccount.table.core.windows.net
- Queue service: http://mystorageaccount.queue.core.windows.net
- File service: http://mystorageaccount.file.core.windows.net

The URL for accessing an object in a storage account is built by appending the object's location in the storage account to the endpoint. For example, to access *myblob* in the *mycontainer*, use this format: http://mystorageaccount.blob.core.windows.net/mycontainer/myblob.

✓ A Blob storage account only exposes the Blob service endpoint. And, you can also configure a custom domain name to use with your storage account.

For more information, you can see:

Storage Account Endpoints - **https://docs.microsoft.com/en-us/azure/storage/common/storage-create-storage-account?toc=%2fazure%2fstorage%2fblobs%2ftoc.json#storage-account-endpoints**[3]

Configuring Custom Domain Names

You can specify a custom domain for accessing blob content instead of using the Azure URLs. There are two ways to configure this service: Direct CNAME mapping and an intermediary domain.

Direct CNAME mapping for example, to enable a custom domain for the blobs.contoso.com sub domain to an Azure storage account, create a CNAME record that points from blobs.contoso.com to the Azure storage account [storage account].blob.core.windows.net. The following example maps a domain to an Azure storage account in DNS:

CNAME record	Target
blobs.contoso.com	contosoblobs.blob.core.windows.net

Intermediary mapping with *asverify* Mapping a domain that is already in use within Azure may result in minor downtime as the domain is updated. If you have an application with an SLA, by using the domain you can avoid the downtime by using a second option, the asverify subdomain, to validate the domain. By prepending asverify to your own subdomain, you permit Azure to recognize your custom domain without modifying the DNS record for the domain. After you modify the DNS record for the domain, it will be mapped to the blob endpoint with no downtime.

[1] https://docs.microsoft.com/en-us/azure/storage/common/storage-account-options
[2] https://docs.microsoft.com/en-us/azure/storage/common/storage-account-options
[3] https://docs.microsoft.com/en-us/azure/storage/common/storage-create-storage-account?toc=%2fazure%2fstorage%2fblobs%2ftoc.json

The following examples maps a domain to the Azure storage account in DNS with the asverify intermediary domain:

CNAME record	Target
asverify.blobs.contoso.com	asverify.contosoblobs.blob.core.windows.net
blobs.contoso.com	contosoblobs.blob.core.windows.net

For more information, you can see:

Configure a custom domain name for your Blob storage endpoint - **https://docs.microsoft.com/en-us/azure/storage/blobs/storage-custom-domain-name**

Pricing and Billing

All storage accounts use a pricing model for blob storage based on the tier of each blob. When using a storage account, the following billing considerations apply:

- **Storage costs**: In addition to, the amount of data stored, the cost of storing data varies depending on the storage tier. The per-gigabyte cost decreases as the tier gets cooler.
- **Data access costs**: Data access charges increase as the tier gets cooler. For data in the cool and archive storage tier, you are charged a per-gigabyte data access charge for reads.
- **Transaction costs**: There is a per-transaction charge for all tiers that increases as the tier gets cooler.
- **Geo-Replication data transfer costs**: This charge only applies to accounts with geo-replication configured, including GRS and RA-GRS. Geo-replication data transfer incurs a per-gigabyte charge.
- **Outbound data transfer costs**: Outbound data transfers (data that is transferred out of an Azure region) incur billing for bandwidth usage on a per-gigabyte basis, consistent with general-purpose storage accounts.
- **Changing the storage tier**: Changing the account storage tier from cool to hot incurs a charge equal to reading all the data existing in the storage account. However, changing the account storage tier from hot to cool incurs a charge equal to writing all the data into the cool tier (GPv2 accounts only).

For more information, you can see:

Pricing model for Blob storage accounts - **https://azure.microsoft.com/pricing/details/storage/**

Outbound data transfer charges - **https://azure.microsoft.com/pricing/details/data-transfers/**

Creating Storage Accounts

Create a storage account[4]

Every storage account must belong to an Azure resource group. A resource group is a logical container for grouping your Azure services. When you create a storage account, you have the option to either create a new resource group, or use an existing resource group. This quickstart shows how to create a new resource group.

A **general-purpose v2** storage account provides access to all of the Azure Storage services: blobs, files, queues, tables, and disks. The quickstart creates a general-purpose v2 storage account, but the steps to create any type of storage account are similar.

[4] https://docs.microsoft.com/en-us/azure/storage/common/storage-quickstart-create-account?tabs=azure-portal#create-a-storage-account-1

To create a general-purpose v2 storage account in the Azure portal, follow these steps:

1. In the Azure portal, select **All services**. In the list of resources, type **Storage Accounts**. As you begin typing, the list filters based on your input. Select **Storage Accounts**.
2. On the **Storage Accounts** window that appears, choose **Add**.
3. Select the subscription in which to create the storage account.
4. Under the **Resource group** field, select **Create new**. Enter a name for your new resource group, as shown in the following image.
5.
6. Next, enter a name for your storage account. The name you choose must be unique across Azure. The name also must be between 3 and 24 characters in length, and can include numbers and lowercase letters only.
7. Select a location for your storage account, or use the default location.
8. Leave these fields set to their default values:

Field	Value
Deployment model	Resource Manager
Performance	Standard
Account kind	StorageV2 (general-purpose v2)
Replication	Locally redundant storage (LRS)
Access tier	Hot

9. Select **Review + Create** to review your storage account settings and create the account.
10. Select **Create**.

For more information about types of storage accounts and other storage account settings, see **Azure storage account overview**[5]. For more information on resource groups, see **Azure Resource Manager overview**[6].

Demonstration - Creating Storage Accounts

In this demonstration, we will look at creating storage accounts.

Create a storage account in the portal

1. In the Azure portal, select **All services**. In the list of resources, type Storage Accounts. As you begin typing, the list filters based on your input. Select **Storage Accounts**.
2. On the Storage Accounts window that appears, choose **Add**.
3. Select the **subscription** in which to create the storage account.
4. Under the Resource group field, select **Create new**. Enter a name for your new resource group.
5. Enter a **name** for your storage account. The name you choose must be unique across Azure. The name also must be between 3 and 24 characters in length, and can include numbers and lowercase letters only.
6. Select a **location** for your storage account, or use the default location.

[5] https://docs.microsoft.com/azure/storage/common/storage-account-overview

[6] https://docs.microsoft.com/azure/azure-resource-manager/resource-group-overview

7. Leave these fields set to their default values:
   - Deployment model: **Resource Manager**
   - Performance: **Standard**
   - Account kind: **StorageV2 (general-purpose v2)**
   - Replication: **Locally redundant storage (LRS)**
   - Access tier: **Hot**
8. Select **Review + Create** to review your storage account settings and create the account.
9. Select **Create**.

Create a storage account using PowerShell

Use the following code to create a storage account using PowerShell. Swap out the storage types and names to suit your requirements.

```
Get-AzLocation | select Location
$location = "westus"
$resourceGroup = "storage-demo-resource-group"
New-AzResourceGroup -Name $resourceGroup -Location $location
New-AzStorageAccount -ResourceGroupName $resourceGroup -Name "storagedemo"
-Location $location -SkuName Standard_LRS -Kind StorageV2
```

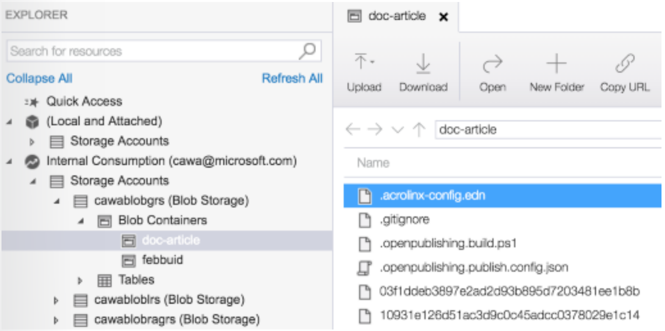
Create a storage account using Azure CLI

Use the following code to create a storage account using Azure CLI. Change the storage types and names to suit your requirements.

```
az group create --name storage-resource-group --location westus
az account list-locations --query "[].{Region:name}" --out table
az storage account create --name storagedemo --resource-group storage-re-
source-group --location westus --sku Standard_LRS --kind StorageV2
```

Azure Storage Explorer

Microsoft Azure Storage Explorer is a standalone app from Microsoft that allows you to easily work with Azure Storage data.


EXPLORER
Search for resources
Collapse All
Refresh All
Quick Access
(Local and Attached)
Storage Accounts
Internal Consumption (cawa@microsoft.com)
Storage Accounts
cawablobgrs (Blob Storage)
Blob Containers
doc-article
febbuid
Tables
cawablobirs (Blob Storage)
cawablobragrs (Blob Storage)
doc-article
Upload
Download
Open
New Folder
Copy URL
doc-article
Name
.acrolinx-config.edn
.gitignore
.openpublishing.build.ps1
.openpublishing.publish.config.json
03f1ddeb3897e2ad2d93b895d7203481ee1b8b
10931e126d51ac3d9c0c45adcc0378029e1c14

Some of the benefits of Azure Storage Explorer include the ability to access multiple accounts and subscriptions across Azure,

Azure Stack, and the sovereign Cloud. Additionally you can use Azure Storage Explorer to create, delete, view and edit Blob, Queue, Table, File, Cosmos DB storage and Data Lake storage. Storage Explorer is available for Windows, Mac, and Linux.

Azure Storage Explorer Features

The following features are present in the latest version of Storage Explorer.

Blob storage

- View, delete, and copy blobs and folders.
- Upload and download blobs while maintaining data integrity.
- Manage snapshots for blobs.

Table storage

- Query entities with OData or query builder.
- Add, edit, and delete entities.
- Import and export tables and query results.

Azure Cosmos DB storage

- Create, manage, and delete databases and collections.
- Generate, edit, delete, and filter documents.
- Manage stored procedure, triggers, and user-defined functions.

Queue storage

- Peek most recent 32 messages.
- View, add, and dequeue messages.
- Clear queue.

File storage

- Navigate files through directories.
- Upload, download, delete, and copy files and directories.
- View and edit file properties.

Azure Data Lake storage

- Navigate ADLS resources across multiple ADL accounts.
- Upload, download files and folders.
- Copy folders or files to the clipboard.
- Delete files and folders.

Azure Storage Explorer has many uses when it comes to managing your storage. See the following articles to learn more.

- **Connect to an Azure subscription**[7]: Manage storage resources that belong to your Azure subscription.
- **Work with local development storage**[8]: Manage local storage by using the Azure Storage Emulator.

[7] https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-manage-with-storage-explorer

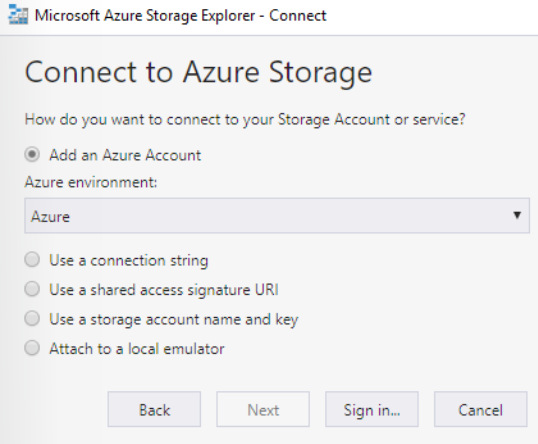
[8] https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-manage-with-storage-explorer

- **Attach to external storage**[9]: Manage storage resources that belong to another Azure subscription or that are under national Azure clouds by using the storage account's name, key, and endpoints.
- **Attach a storage account by using an SAS**[10]: Manage storage resources that belong to another Azure subscription by using a shared access signature (SAS).
- **Attach a service by using an SAS**[11]: Manage a specific storage service (blob container, queue, or table) that belongs to another Azure subscription by using an SAS.
- **Connect to an Azure Cosmos DB account by using a connection string**[12]: Manage Cosmos DB account by using a connection string.

Storage Explorer Connection Options

Storage Explorer provides several ways to connect to storage accounts. For example, you can:

- Connect to storage accounts associated with your Azure subscriptions.
- Connect to storage accounts and services that are shared from other Azure subscriptions.
- Connect to and manage local storage by using the Azure Storage Emulator.



Microsoft Azure Storage Explorer - Connect

Connect to Azure Storage

How do you want to connect to your Storage Account or service?

● Add an Azure Account

Azure environment:

Azure ▾

○ Use a connection string

○ Use a shared access signature URI

○ Use a storage account name and key

○ Attach to a local emulator

In addition, you can work with storage accounts in global and national Azure:

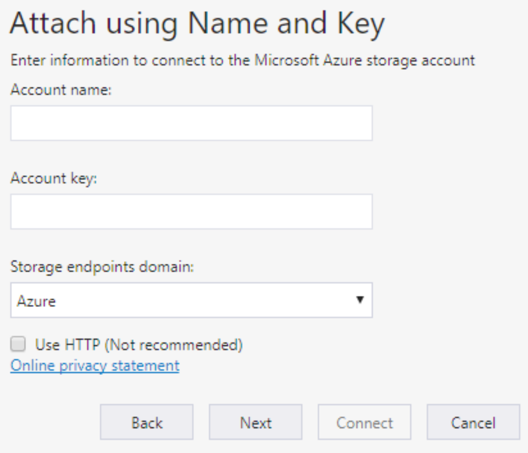
- **Connect to an Azure subscription**. Manage storage resources that belong to your Azure subscription.
- **Work with local development storage**. Manage local storage by using the Azure Storage Emulator.
- **Attach to external storage**. Manage storage resources that belong to another Azure subscription or that are under national Azure clouds by using the storage account's name, key, and endpoints (shown below.)
- **Attach a storage account by using an SAS**. Manage storage resources that belong to another Azure subscription by using a shared access signature (SAS).
- **Attach a service by using an SAS**. Manage a specific storage service (blob container, queue, or table) that belongs to another Azure subscription by using an SAS.

[9] https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-manage-with-storage-explorer
[10] https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-manage-with-storage-explorer
[11] https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-manage-with-storage-explorer
[12] https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-manage-with-storage-explorer

- **Connect to an Azure Cosmos DB account by using a connection string**. Manage Cosmos DB account by using a connection string.

Accessing external storage accounts

As mentioned previously, Storage Explorer lets you attach to external storage accounts so that storage accounts can be easily shared. To create the connection you will need the storage **Account name** and **Account key**. In the portal, the account key is called **key1**.



Attach using Name and Key

Enter information to connect to the Microsoft Azure storage account

Storage endpoints domain: Azure

☐ Use HTTP (Not recommended)

To use a name and key from a national cloud, use the **Storage endpoints domain** drop-down to select **Other** and then enter the custom storage endpoint domain.

✓ Access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.
When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. We will cover access keys in more detail later.

✓ Notice this connection method provide access to the entire storage account.

Demonstration - Storage Explorer

Note: If you have an older version of the Storage Explorer, be sure to upgrade. These steps use version 1.6.2.

Note: For the demonstration we will only do a basic storage account connection.

In this demonstration, we will look at several common Azure Storage Explorer tasks.

Download and install Storage Explorer

1. Download and install Azure Storage Explorer - **https://azure.microsoft.com/en-us/features/storage-explorer/**
2. After the installation, launch the tool.
3. Review the Release Notes and menu options.

Connect to an Azure subscription

1. In Storage Explorer, select **Manage Accounts**, second icon top left. This will take you to the Account Management Panel.

2. The left pane now displays all the Azure accounts you've signed in to. To connect to another account, select **Add an account**.
3. If you want to sign into a national cloud or an Azure Stack, click on the Azure environment dropdown to select which Azure cloud you want to use.
4. Once you have chosen your environment, click the **Sign in…** button.
5. After you successfully sign in with an Azure account, the account and the Azure subscriptions associated with that account are added to the left pane.
6. Select the Azure subscriptions that you want to work with, and then select **Apply**.
7. The left pane displays the storage accounts associated with the selected Azure subscriptions.

Note: This next section requires an Azure storage account.

Attach an Azure storage account

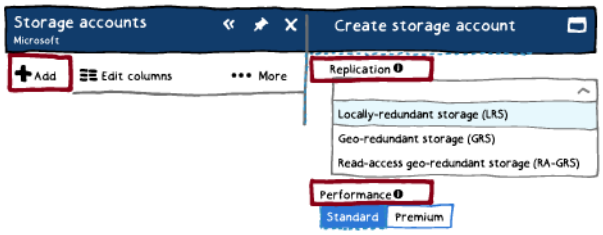
1. Access the Azure portal, and your storage account.
2. Explore the choice for **Storage Explorer**, which is now in preview.
3. Select **Access keys** and read the information about using the keys.
4. To connect in Storage Explorer, you will need the **Storage account name** and **Key1** information.
5. In Storage Explorer, **Add an account**.
6. Paste your account name in the Account name text box, and paste your account key (the key1 value from the Azure portal) into the Account key text box, and then select **Next**.
7. Verify your storage account is available in the navigation pane. You may need to refresh the page.
8. Right-click your storage account and notice the choices including **Open in portal**, **Copy primary key**, and **Add to Quick Access**.

Data Replication

Replication Options

The data in your Azure storage account is always replicated to ensure durability and high availability. Azure Storage replication copies your data so that it is protected from planned and unplanned events ranging from transient hardware failures, network or power outages, massive natural disasters, and so on. You can choose to replicate your data within the same data center, across zonal data centers within the same region, and even across regions. Replication ensures that your storage account meets the Service-Level Agreement (SLA) for Storage even in the face of failures. See the SLA for information about Azure Storage guarantees for durability and availability.

When you create a Standard storage account there are four replications schemes: **Locally-redundant storage (LRS)**, **Geo-redundant storage (GRS)**, **Read-access geo-redundant storage (RA-GRS)**, and **Zone-redundant storage (ZRS)**.



Are there any costs to changing my account's replication strategy?

It depends on your conversion path. Ordering from cheapest to the most expensive redundancy offering we have LRS, ZRS, GRS, and RA-GRS. For example, going from LRS to anything will incur additional charges because you are going to a more sophisticated redundancy level. Going to GRS or RA-GRS will incur an egress bandwidth charge because your data (in your primary region) is being replicated to your remote secondary region. This is a one-time charge at initial setup. After the data is copied, there are no further conversion charges. You will only be charged for replicating any new or updates to existing data. For details on bandwidth charges, see Azure Storage Pricing page.

If you change from GRS to LRS, there is no additional cost, but your replicated data is deleted from the secondary location.

✓ If you select Premium performance only LRS replication will be available.

✓ If you create availability sets for your virtual machines, then Azure uses Zone-redundant Storage (ZRS).

For more information, you can see:

Azure storage replication - **https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy**

Locally -redundant Storage

Replication	Copies	Strategy
Locally redundant storage (LRS)	Maintains three copies of your data.	Data is replicated three time within a single facility in a single region.

Locally redundant storage (LRS) provides at least 99.999999999% (11 nines) durability of objects over a given year. LRS provides this object durability by replicating your data to a storage scale unit. A datacenter, located in the region where you created your storage account, hosts the storage scale unit. A write request to an LRS storage account returns successfully only after the data is written to all replicas. Each replica resides in separate fault domains and upgrade domains within a storage scale unit. A storage scale unit is a collection of racks of storage nodes. A fault domain (FD) is a group of nodes that represent a physical unit of failure. Think of a fault domain as nodes belonging to the same physical rack. An upgrade domain (UD) is a group of nodes that are upgraded together during the process of a service upgrade (rollout). The replicas are spread across UDs and FDs within one storage scale unit. This architecture ensures your data is available if a hardware failure affects a single rack or when nodes are upgraded during a service upgrade.

LRS is the lowest-cost replication option and offers the least durability compared to other options. If a datacenter-level disaster (for example, fire or flooding) occurs, all replicas may be lost or unrecoverable. To mitigate this risk, Microsoft recommends using either zone-redundant storage (ZRS) or geo-redundant storage (GRS).

If your application stores data that can be easily reconstructed if data loss occurs, you may opt for LRS. Some applications are restricted to replicating data only within a country due to data governance requirements. In some cases, the paired regions across which the data is replicated for GRS accounts may be in another country. For more information on paired regions, see Azure regions.

LRS is a low-cost option for protecting your data from local hardware failures. If a datacenter-level disaster (for example, fire or flooding) occurs, all replicas may be lost or unrecoverable. To mitigate this risk, Microsoft recommends using either zone-redundant storage (ZRS) or geo-redundant storage (GRS).

However, LRS may be appropriate in these scenarios:

- If your application stores data that can be easily reconstructed if data loss occurs, you may opt for LRS.
- Some applications are restricted to replicating data only within a country due to data governance requirements.

✓ Do you think LRS is a good choice for your organization?

Zone-redundant Storage

Replication	Copies	Strategy
Zone-redundant storage (ZRS)	Maintains three copies of your data.	Data is replicated three times across two to three facilities, either within a single region or across two regions.

Zone Redundant Storage (ZRS) synchronously replicates your data across three (3) storage clusters in a single region. Each storage cluster is physically separated from the others and resides in its own availability zone. Each availability zone, and the ZRS cluster within it, is autonomous, with separate utilities and networking capabilities.

Storing your data in a ZRS account ensures that you will be able access and manage your data if a zone becomes unavailable. ZRS provides excellent performance and extremely low latency.

Here are a few of more things to know about ZRS:

- ZRS is not yet available in all regions.
- Changing to ZRS from another data replication option requires the physical data movement from a single storage stamp to multiple stamps within a region.
- ZRS may not protect your data against a regional disaster where multiple zones are permanently affected. Instead, ZRS offers resiliency for your data in the case of unavailability.

Support coverage and regional availability
ZRS currently supports standard general-purpose v2 account types. ZRS is available for block blobs, non-disk page blobs, files, tables, and queues.

What happens when a zone becomes unavailable?
Your data is still accessible for both read and write operations even if a zone becomes unavailable. Microsoft recommends that you continue to follow practices for transient fault handling. These practices include implementing retry policies with exponential back-off.

When a zone is unavailable, Azure undertakes networking updates, such as DNS repointing. These updates may affect your application if you are accessing your data before the updates have completed.

ZRS may not protect your data against a regional disaster where multiple zones are permanently affected. Instead, ZRS offers resiliency for your data if it becomes temporarily unavailable. For protection against regional disasters, Microsoft recommends using geo-redundant storage (GRS).

✓ Consider ZRS for scenarios that require strong consistency, strong durability, and high availability even if an outage or natural disaster renders a zonal data center unavailable.

Geo-redundant storage

Replication	Copies	Strategy
Geo-redundant storage (GRS)	Maintains six copies of your data.	Data is replicated three times within the primary region and is also replicated three times in a secondary region hundreds of miles away from the primary region.
Read access geo-redundant storage (RA-GRS)	Maintains six copies of your data.	Data is replicated to a secondary geographic location and provides read access to your data in the secondary location.

Geo-redundant storage (GRS) is the default and recommended replication option and is sometimes called cross-regional replication. GRS replicates your data to a secondary region (hundreds of miles away from the primary location of the source data). GRS costs more than LRS, but GRS provides a higher level of durability for your data, even if there is a regional outage. Geo-redundant storage (GRS) is designed to provide at least 99.99999999999999% (16 9's) durability of objects over a given year by replicating your data to a secondary region that is hundreds of miles away from the primary region. If your storage account has GRS enabled, then your data is durable even in the case of a complete regional outage or a disaster in which the primary region isn't recoverable.

For a storage account with GRS or RA-GRS enabled, all data is first replicated with locally redundant storage (LRS). An update is first committed to the primary location and replicated using LRS. The update is then replicated asynchronously to the secondary region using GRS. When data is written to the secondary location, it's also replicated within that location using LRS. Both the primary and secondary regions manage replicas across separate fault domains and upgrade domains within a storage scale unit. The storage scale unit is the basic replication unit within the datacenter. Replication at this level is provided by LRS; for more information, see Locally redundant storage (LRS): Low-cost data redundancy for Azure Storage.

If you opt for GRS, you have two related options to choose from:

- **GRS** replicates your data to another data center in a secondary region, but that data is available to be read only if Microsoft initiates a failover from the primary to secondary region.
- **Read-access geo-redundant storage** (RA-GRS) is based on GRS. RA-GRS replicates your data to another data center in a secondary region, and also provides you with the option to read from the secondary region. With RA-GRS, you can read from the secondary regardless of whether Microsoft initiates a failover from the primary to the secondary.

What is the RPO and RTO with GRS?

Recovery Point Objective (RPO): In GRS and RA-GRS, the storage service asynchronously geo-replicates the data from the primary to the secondary location. In the event that the primary region becomes unavailable, you can perform an account failover (preview) to the secondary region. When you initiate a failover, recent changes that haven't yet been geo-replicated may be lost. The number of minutes of potential data that's lost is known as the RPO. The RPO indicates the point in time to which data can be recovered. Azure Storage typically has an RPO of less than 15 minutes, although there's currently no SLA on how long geo-replication takes.

Recovery Time Objective (RTO): The RTO is a measure of how long it takes to perform the failover and get the storage account back online. The time to perform the failover includes the following actions:

- The time until the customer initiates the failover of the storage account from the primary to the secondary region.
- The time required by Azure to perform the failover by changing the primary DNS entries to point to the secondary location.

✓ If you enable RA-GRS and your primary endpoint for the Blob service is *myaccount.blob.core.windows.net*, then your secondary endpoint is *myaccount-secondary.blob.core.windows.net*. The access keys for your storage account are the same for both the primary and secondary endpoints.

Replication Option Comparison

The following table provides a quick overview of the scope of durability and availability that each replication strategy will provide you for a given type of event (or event of similar impact).

Replication Option	LRS	ZRS	GRS	RA-GRS
Node unavailability within a data center	Yes	Yes	Yes	Yes

Replication Option	LRS	ZRS	GRS	RA-GRS
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes	Yes
A region-wide outage	No	No	Yes	Yes
Read access to your data (in a remote, geo-replicated region) in the event of region-wide unavailability	No	No	No	Yes
Available in storage account types	GPv1, GPv2, Blob	Standard,GPv2	GPv1, GPv2, Blob	GPv1, GPv2, Blob

For more information, you can see:

Read-access geo-redundant storage - **https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy-grs#read-access-geo-redundant-storage**[13]

Storage Accounts PowerShell Tasks

Below are a few common storage accounts tasks using PowerShell.

Task	Example
Check to see if a storage account name is available.	**Get-AzureRmStorageAccountNameAvailability** -Name ‘mystorageaccount’
Create a storage account.	**New-AzureRmStorageAccount** -ResourceGroupName MyResourceGroup -AccountName mystorageaccount -Location westus -SkuName Standard_GRS
Retrieve a specific storage account or all the storage accounts in a resource group or subscription.	**Get-AzureRmStorageAccount** -ResourceGroupName "RG01" -AccountName “mystorageaccount”
Modify storage account properties, such as type.	**Set-AzureRmStorageAccount** -ResourceGroupName "MyResourceGroup" -AccountName “mystorageaccount” -Type "Standard_RAGRS"

✓ Be sure to try a few commands using the reference link below. You’ll need to create unique names for your own storage accounts and resource groups.

For more information, you can see:

Create a storage account - **https://docs.microsoft.com/en-us/azure/storage/common/storage-powershell-guide-full#create-a-storage-account**[14]

[13] https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy-grs

[14] https://docs.microsoft.com/en-us/azure/storage/common/storage-powershell-guide-full

Storing and Accessing Data

Import and Export Service

When it comes to transferring very large amounts of data to or from the cloud you will want to consider using the Azure Import/Export service. The Azure Import/Export Service allows you to:

- **Import**. Securely transfer large amounts of data to Azure Blob storage (block and page blobs) and Azure Files by shipping disk drives to an Azure data center. In this case, you will be shipping hard drives containing your data.
- **Export**. Transfer data from Azure storage to hard disk drives and ship to your on-premise sites. Currently, you can only export **Block** blobs, **Page** blobs or **Append** blobs from Azure storage using this service. Exporting Azure Files is not currently supported. In this case, you will be shipping empty hard drives.

Consider using Azure Import/Export service when uploading or downloading data over the network is too slow or getting additional network bandwidth is cost-prohibitive. Scenarios where this would be useful include:

- **Migrating data to the cloud**. Move large amounts of data to Azure quickly and cost effectively.
- **Content distribution**. Quickly send data to your customer sites.
- **Backup**. Take backups of your on-premises data to store in Azure blob storage.
- **Data recovery**. Recover large amount of data stored in blob storage and have it delivered to your on-premises location.

✓ Only 2.5” SSD or 2.5" or 3.5" SATA II or III internal HDD are supported for use with the Import/Export service.

For more information, you can see:

Import/Export Prerequisites - **https://docs.microsoft.com/en-us/azure/storage/common/storage-import-export-service#prerequisites**[15]

Azure Import and Export Service - **https://azure.microsoft.com/en-us/documentation/articles/storage-import-export-service/**

Import/Export Pricing - **https://azure.microsoft.com/en-us/pricing/details/storage-import-export/**

Components and Requirements

This topic lists the components that make up Import/Export service and the requirements for using the service.

Import and Export service components

- **Import/Export service**. This service available in Azure portal helps the user create and track data import (upload) and export (download) jobs.
- **WAImportExport tool**. This is a command-line tool that does the following:
  - Prepares your disk drives that are shipped for import.
  - Facilitates copying your data to the drive.
  - Encrypts the data on the drive with BitLocker.

[15] https://docs.microsoft.com/en-us/azure/storage/common/storage-import-export-service

- Generates the drive journal files used during import creation.
- Helps identify numbers of drives needed for export jobs.

Note: The WAImportExport tool is available in two versions, version 1 and 2. We recommend that you use:
Version 1 for import/export into Azure Blob storage.
Version 2 for importing data into Azure files.

- **Disk Drives**. You can ship Solid-state drives (SSDs) or Hard disk drives (HDDs) to the Azure datacenter. When creating an import job, you ship disk drives containing your data. When creating an export job, you ship empty drives to the Azure datacenter.

Requirements

Operating systems

- Windows Server 64-bit OS that supports BitLocker Drive Encryption.
- Windows clients that have .NET Framework 4.5.1 and BitLocker.

Supported storage accounts

- General Purpose v2 storage accounts (recommended for most scenarios)
- Blob Storage accounts
- General Purpose v1 storage accounts (both Classic or Azure Resource Manager deployments)

Supported storage types

- Import jobs can include Azure Blob storage, Azure File storage, Blob blobs, and Page blobs.
- Export jobs can include Azure Blob storage, Block blobs, Page blobs, and Append blobs. Azure Files not supported.

Supported disks

Disk type	Size	Supported	Not supported
SSD	2.5"	All	
HDD	2.5" and 3.5"	SATA II, SATA III	External HDD with built-in USB adaptor and disks inside the casing of an external HDD.

Import and Export Tool

The **Microsoft Azure Import/Export Tool** is the drive preparation and repair tool that you can use with the Microsoft Azure Import/Export service. You can use the tool for the following functions:

- Before creating an import job, you can use this tool to copy data to the hard drives you are going to ship to an Azure datacenter.
- After an import job has completed, you can use this tool to repair any blobs that were corrupted, were missing, or conflicted with other blobs.
- After you receive the drives from a completed export job, you can use this tool to repair any files that were corrupted or missing on the drives.

Import/Export service requires the use of internal SATA II/III HDDs or SSDs. Each disk contains a single NTFS volume that you encrypt with BitLocker when preparing the drive. To prepare a drive, you must connect it to a computer running a 64-bit version of the Windows client or server operating system and run the WAImportExport tool from that computer. The WAImportExport tool handles data copy, volume encryption, and creation of journal files. Journal files are necessary to create an import/export job and help ensure the integrity of the data transfer.

What is a journal file?

Each time you run the WAImportExport tool to copy files to the hard drive, the tool creates a copy session. The state of the copy session is written to the journal file. If a copy session is interrupted (for example, due to a system power loss), it can be resumed by running the tool again and specifying the journal file on the command line.

For each hard drive that you prepare with the Azure Import/Export Tool, the tool will create a single journal file with name DriveID.xml where DriveID is the serial number associated to the drive that the tool reads from the disk. You will need the journal files from all of your drives to create the import job. The journal file can also be used to resume drive preparation if the tool is interrupted.

Simple Import Example

```
WAImportExport.exe PrepImport /j:<JournalFile> /id:<SessionId> /DataSet:<-
dataset.csv>
```

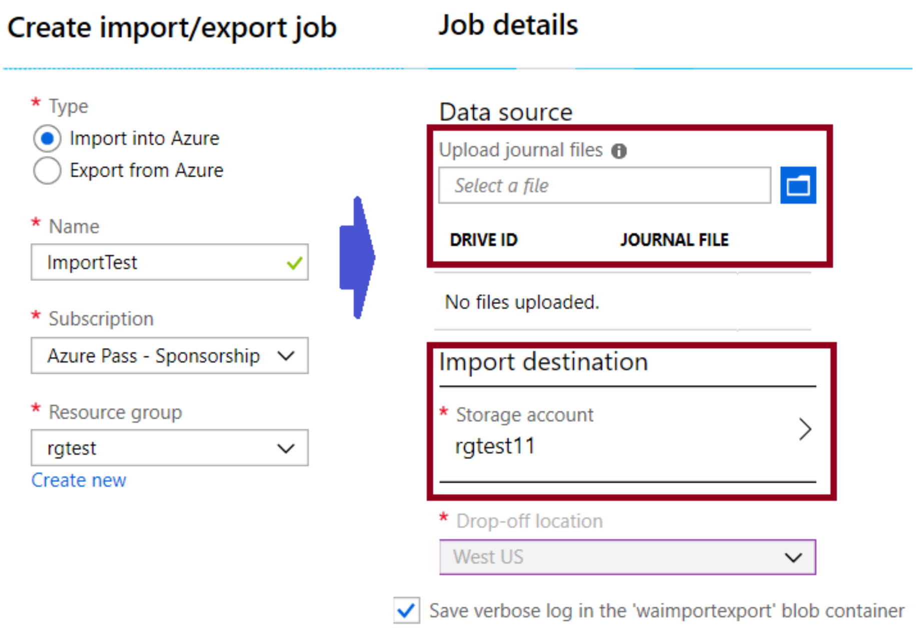
- **PrepImport**. Indicates the tool is preparing drives for an import job.
- **JournalFile**. Path to the journal file that will be created. A journal file tracks a set of drives and records the progress in preparing these drives. The journal file must always be specified.
- **SessionId**. The session Id is used to identify a copy session. It is used to ensure accurate recovery of an interrupted copy session.
- **DataSet**. A CSV file that contains a list of directories and/or a list of files to be copied to target drives.

✓ The WAImportExport tool is available from Microsoft Download site at **https://aka.ms/Welhs7**.

Import Jobs

An Import job securely transfers large amounts of data to Azure Blob storage (block and page blobs) and Azure Files by shipping disk drives to an Azure datacenter. In this case, you will be shipping hard drives containing your data.

Your job will be configured in the Portal. Notice the need for the journal file, created by the Import/Export tool, and a storage account to receive the data. Not shown is the return shipping information.



Create import/export job

* Type
● Import into Azure
○ Export from Azure

* Name
ImportTest ✓

* Subscription
Azure Pass - Sponsorship

* Resource group
rgtest
Create new

Job details

Data source

Upload journal files
Select a file

DRIVE ID	JOURNAL FILE
No files uploaded.	

Import destination

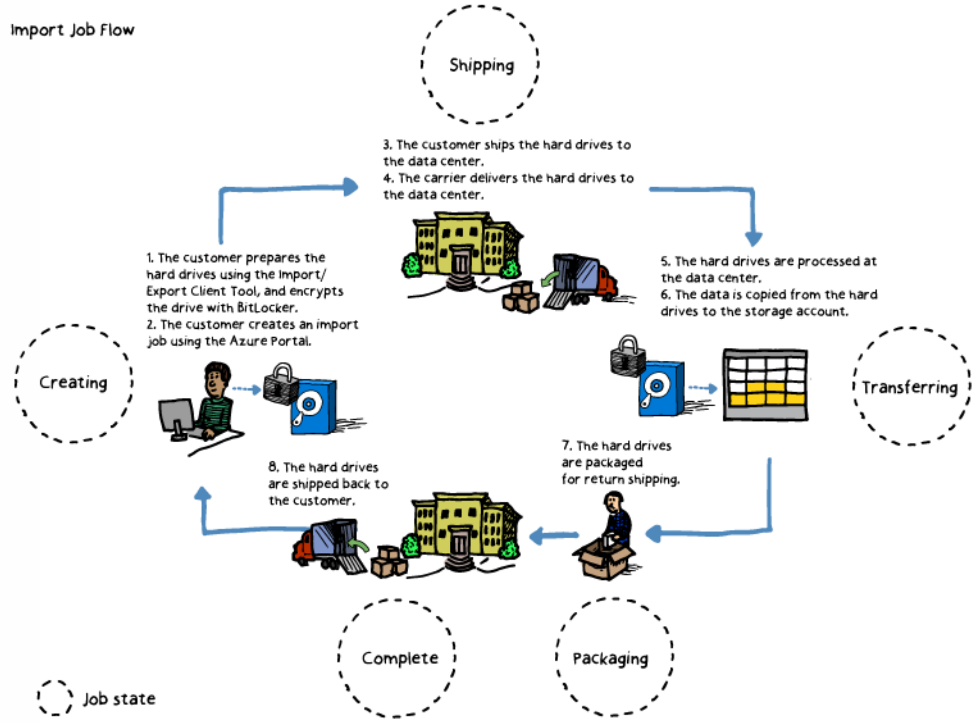
* Storage account
rgtest11

* Drop-off location
West US

☑ Save verbose log in the 'waimportexport' blob container

At a high level, an import job involves the following steps:

- Determine the data to be imported, and the number of drives you need.
- Identify the destination blob or file location for your data in Azure storage.
- Use the WAImportExport Tool to copy your data to one or more hard disk drives and encrypt them with BitLocker.
- Create an import job in your target storage account using the Azure portal or the Import/Export REST API. If using the Azure portal, upload the drive journal files.
- Provide the return address and carrier account number to be used for shipping the drives back to you.
- Ship the hard disk drives to the shipping address provided during job creation.
- Update the delivery tracking number in the import job details and submit the import job.
- Drives are received and processed at the Azure data center.


Import Job Flow
Shipping
3. The customer ships the hard drives to the data center.
4. The carrier delivers the hard drives to the data center.
1. The customer prepares the hard drives using the Import/Export Client Tool, and encrypts the drive with BitLocker.
2. The customer creates an import job using the Azure Portal.
5. The hard drives are processed at the data center.
6. The data is copied from the hard drives to the storage account.
Creating
Transferring
7. The hard drives are packaged for return shipping.
8. The hard drives are shipped back to the customer.
Complete
Packaging
Job state

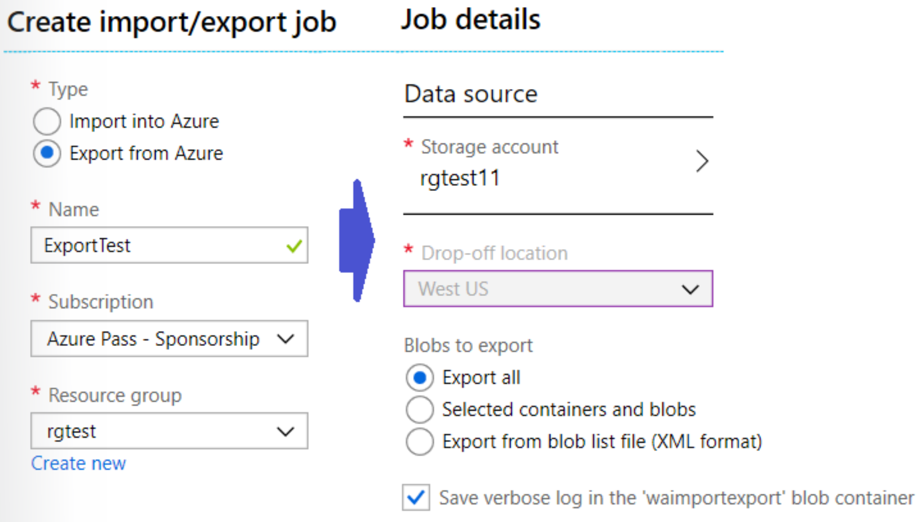
- Drives are shipped using your carrier account to the return address provided in the import job.

For more information, you can see:

Inside an import job - **https://docs.microsoft.com/en-us/azure/storage/common/storage-import-export-service#inside-an-import-job**[16]

Export Jobs

Export jobs transfer data from Azure storage to hard disk drives and ship to your on-premise sites.



Create import/export job

* Type
○ Import into Azure
● Export from Azure

* Name
ExportTest ✓

* Subscription
Azure Pass - Sponsorship

* Resource group
rgtest
Create new

Job details

Data source

* Storage account
rgtest11

* Drop-off location
West US

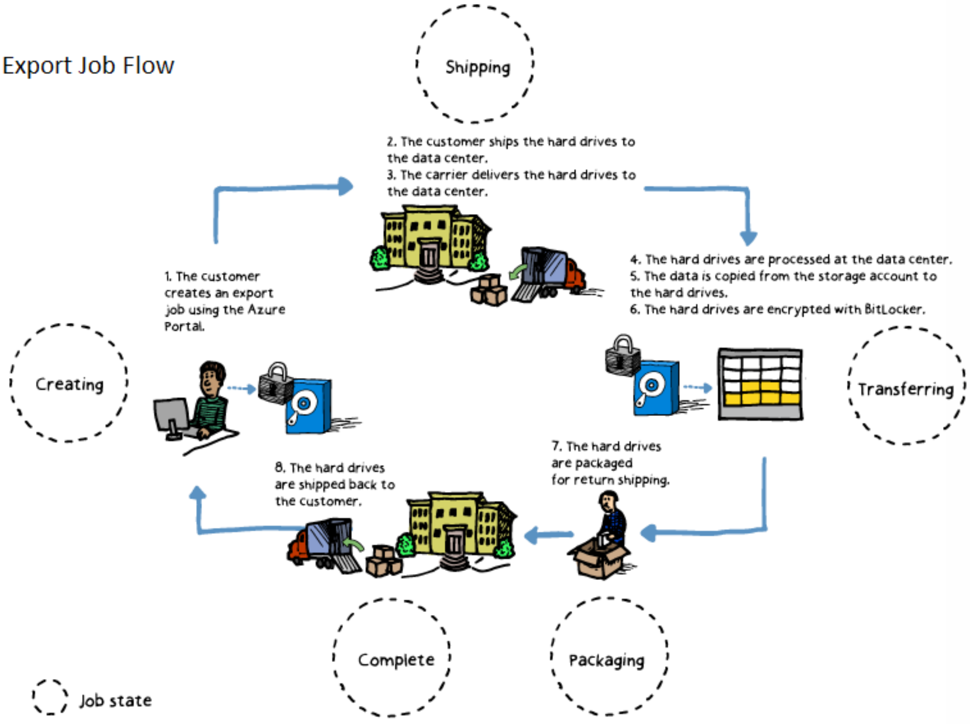
Blobs to export
● Export all
○ Selected containers and blobs
○ Export from blob list file (XML format)

☑ Save verbose log in the 'waimportexport' blob container

[16] https://docs.microsoft.com/en-us/azure/storage/common/storage-import-export-service

At a high level, an export job involves the following steps:

- Determine the data to be exported and the number of drives you need.
- Identify the source blobs or container paths of your data in Blob storage.
- Create an export job in your source storage account using the Azure portal or the Import/Export REST API.
- Specify the source blobs or container paths of your data in the export job.
- Provide the return address and carrier account number for to be used for shipping the drives back to you.
- Ship the hard disk drives to the shipping address provided during job creation.
- Update the delivery tracking number in the export job details and submit the export job.
- The drives are received and processed at the Azure data center.
- The drives are encrypted with BitLocker; the keys are available via the Azure portal.
- The drives are shipped using your carrier account to the return address provided in the import job.


Export Job Flow
Shipping
2. The customer ships the hard drives to the data center.
3. The carrier delivers the hard drives to the data center.
1. The customer creates an export job using the Azure Portal.
4. The hard drives are processed at the data center.
5. The data is copied from the storage account to the hard drives.
6. The hard drives are encrypted with BitLocker.
Creating
Transferring
8. The hard drives are shipped back to the customer.
7. The hard drives are packaged for return shipping.
Complete
Packaging
Job state

For more information, you can see:

Inside an export job - **https://docs.microsoft.com/en-us/azure/storage/common/storage-import-export-service#inside-an-export-job**[17]

AzCopy

An alternative method for transferring data is **AzCopy**. AzCopy v10 is the next-generation command-line utility for copying data to/from Microsoft Azure Blob and File storage, which offers a redesigned com-

[17] https://docs.microsoft.com/en-us/azure/storage/common/storage-import-export-service

mand-line interface and new architecture for high-performance reliable data transfers. Using AzCopy, you can copy data between a file system and a storage account, or between storage accounts.

What's new

- Synchronize a file system to Azure Blob or vice versa. Ideal for incremental copy scenarios.
- Supports Azure Data Lake Storage Gen2 APIs.
- Supports copying an entire account (Blob service only) to another account.
- Account to account copy is now using the new Put from URL APIs. No data transfer to the client is needed which makes the transfer faster.
- List/Remove files and blobs in a given path.
- Supports wildcard patterns in a path as well as –include and –exclude flags.
- Improved resiliency: every AzCopy instance will create a job order and a related log file. You can view and restart previous jobs and resume failed jobs. AzCopy will also automatically retry a transfer after a failure.
- General performance improvements.

Authentication options

- **Azure Active Directory** (Supported for Blob and ADLS Gen2 services). Use .\azcopy login to sign in using Azure Active Directory. The user should have *Storage Blob Data Contributor* role assigned to write to Blob storage using Azure Active Directory authentication.
- **SAS tokens** (supported for Blob and File services). Append the SAS token to the blob path on the command line to use it.

Getting started

AzCopy has a simple self-documented syntax. Here's how you can get a list of available commands:

```
AzCopy /?
```

The basic syntax for AzCopy commands is:

```
AzCopy /Source:<source> /Dest:<destination> [Options]
```

✓ AzCopy is available on Windows, Linux, and MacOS.

For more information, you can see:

Download and install AzCopy on Windows - **https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy#download-and-install-azcopy-on-windows**[18]

Demonstration - AzCopy

In this demonstration, we will explore AzCopy.

Install the AzCopy tool

1. Download the Windows 8.1 version - **https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy?toc=%2fazure%2fstorage%2ftables%2ftoc.json#download-and-install-azcopy-on-windows**.

[18] https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy

2. Install and launch the tool.

Explore the help

1. Launch the Microsoft Azure Storage AzCopy tool.
2. View the help.

```
azcopy /?
```

1. Scroll to the top of the Help information and read about the **Common options**, like: source, destination, source key, and destination key.
2. Scroll down the **Samples** section. We will be trying several of these examples. Does anything look particularly interesting?

Download a blob from Blob storage to the file system

Note: This example requires an Azure storage account with blob container and blob file. You will also need to capture parameters in a text editor like Notepad.

1. Access the Azure portal.
2. Access your storage account with the blob you want to download.
3. Select **Access keys** and copy the **Key Key1** value. This will be the *sourcekey:* value.
4. Drill down to the blob of interest, and view the file **Properties**.
5. Copy the **URL** information. This will be the *source:* value.
6. Locate a local destination directory. This will be the *dest:* value. A filename is also required.
7. Construct the command using your values.

```
azcopy  /source:sourceURL /dest:destinationdirectoryandfilename /sourcekey:"key"
```

1. If you have errors, read them carefully and make corrections.
2. Verify the blob was downloaded to your local directory.

Upload files to Azure blob storage

Note: The example continues from the previous example and requires a local directory with files.

1. The *source:* for the command will be a local directory with files.
2. The *dest:* will the blob URL used in the previous example. Be sure to remove the filename, just include the storage account and container.
3. The *destkey:* will the key used in the previous example.
4. Construct the command using your values.

```
azcopy /source:source /dest:destinationcontainer /destkey:key
```

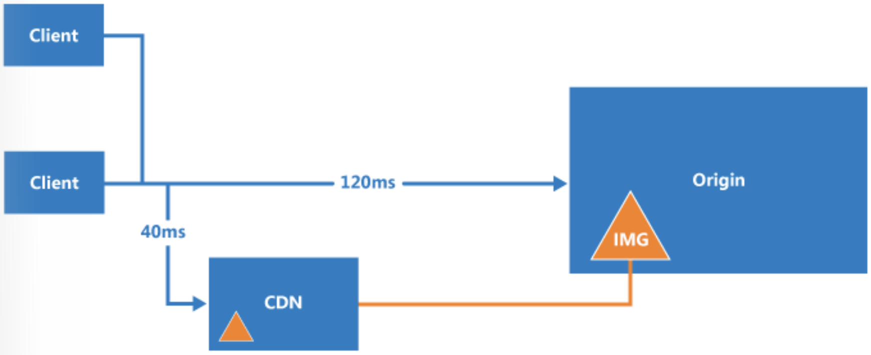
1. If you have errors, read them carefully and make corrections.
2. Verify your local files were copied to the Azure container.
3. Notice there are switches to recurse subdirectories and pattern match.

Content Delivery Network (CDN)

A content delivery network (CDN) is a distributed network of servers that can efficiently deliver content to users. CDNs store cached content on edge servers that are close to end-users.

CDNs are typically used to deliver static content such as images, style sheets, documents, client-side scripts, and HTML pages. The main benefits of using a CDN are:

- Lower latency and faster delivery of content to users, regardless of their geographical location in relation to the datacenter where the application is hosted.
- Helps to reduce load on a server or application, because it does not have to service requests for the content that is hosted in the CDN.



Typical uses for a CDN include:

- Delivering static resources for client applications, often from a website.
- Delivering public static and shared content to devices such as cell phones and tablet computers.
- Serving entire websites that consist of only public static content to clients, without requiring any dedicated compute resources.
- Streaming video files to the client on demand.
- Generally improving the experience for users, especially those located far from the datacenter hosting the application.
- Supporting IoT (Internet of Things) solutions, such as distributing firmware updates.
- Coping with peaks and surges in demand without requiring the application to scale, avoiding the consequent increased running costs.

✓ CDN provides a faster, more responsive user experience. Do you think your organization would be interested in this feature?

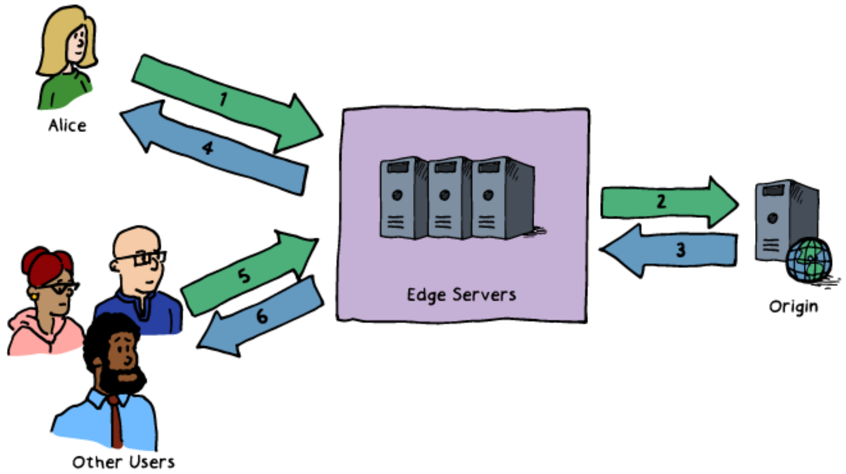
✓ Use the following link to review some of the challenges with deploying CDN including security, deployment, versioning, and testing.

For more information, you can see:

Content Delivery Network Documentation - **https://docs.microsoft.com/en-us/azure/cdn/cdn-overview**

How CDN Works

You can enable Azure Content Delivery Network (CDN) to cache content for the user. The Azure CDN is designed to send audio, video, images, and other files faster and more reliably to customers using servers that are closest to the users. This dramatically increases speed and availability, resulting in significant user experience improvements.



1. A user (Alice) requests a file (also called an asset) using a URL with a special domain name, such as endpointname.azureedge.net. DNS routes the request to the best performing Point-of-Presence (POP) location, which is usually the POP that is geographically closest to the user.
2. If the edge servers in the POP do not have the file in their cache, the edge server requests the file from the origin. The origin can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.
3. The origin returns the file to the edge server, including optional HTTP headers describing the file's Time-to-Live (TTL).
4. The edge server caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server until the TTL expires. Azure CDN automatically applies a default TTL of seven days unless you've set up caching rules in the Azure portal.
5. Additional users may then request the same file using that same URL and may also be directed to that same POP.
6. If the TTL for the file hasn't expired, the edge server returns the file from the cache.

✓ After you enable CDN access to a storage account, all publicly available objects are eligible for CDN edge caching. If you modify an object that's currently cached in the CDN, the updated content will not be available via CDN until CDN refreshes its content after the time-to-live period for the cached content expires.

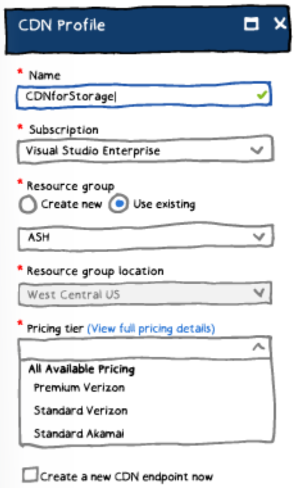
For more information, you can see:

Overview of the Azure Content Delivery Network - **https://docs.microsoft.com/en-us/azure/cdn/cdn-overview**

Azure CDN POP locations by region - **https://docs.microsoft.com/en-us/azure/cdn/cdn-pop-locations**

CDN Profiles

A CDN profile is a collection of CDN endpoints with the same pricing tier and provider (origin). You may create multiple profiles to organize endpoints. For example, you could have profiles with endpoints to different internet domains, web applications, or storage accounts. You can create up to 8 CDN profiles per subscription.



CDN Profile

* Name
CDNforStorage|

* Subscription
Visual Studio Enterprise

* Resource group
○ Create new ● Use existing
ASH

* Resource group location
West Central US

* Pricing tier (View full pricing details)
All Available Pricing
Premium Verizon
Standard Verizon
Standard Akamai

□ Create a new CDN endpoint now

You can create a CDN profile from the Azure portal.

The CDN service is global and not bound to a location, however you must specify a resource group location where the metadata associated with the CDN profile will reside. This location will not have any impact on the runtime availability of your profile.

Several pricing tiers are available. At the time of this writing, there are three tiers: Premium Verizon, Standard Verizon, and Standard Akamai. Pricing is based on TBs of outbound data transfers. Be sure to read about the pricing models in the link at the end of this topic.

Notice you can create your first profile endpoint directly from this blade (last checkbox).

✓ Can you think of different scenarios that would require different CDN profiles?

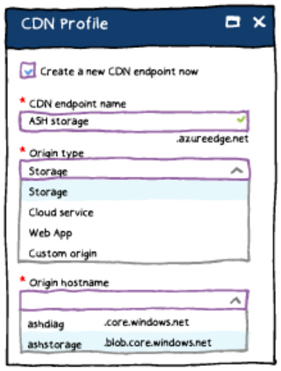
For more information, you can see:

CDN Pricing - **https://azure.microsoft.com/en-us/pricing/details/cdn/**

CDN Endpoints

When you create a new CDN endpoint directly from the CDN profile blade you are prompted for CDN endpoint name, Origin type, and Origin hostname. To access cached content on the CDN, use the CDN URL provided in the portal. In this case,

```
ASHStorage.azureedge.net/<myPublicContainer>/<BlobName>
```



CDN Profile

☑ Create a new CDN endpoint now

* CDN endpoint name

ASH storage ✓

.azureedge.net

* Origin type

Storage

- Storage
- Cloud service
- Web App
- Custom origin

* Origin hostname

ashdiag	.core.windows.net
ashstorage	.blob.core.windows.net

There are four choices for Origin type: Storage, Cloud Service, Web App, and Custom origin. In this course we are focusing on storage CDNs.

When you select Storage as the Origin type, the new CDN endpoint uses the host name of your storage account as the origin server.

There are additional CDN features for your delivery, such as compression, query string, and geo filtering. You can also add custom domain mapping to your CDN endpoint and enable custom domain HTTPS. These options are configured in the Settings blade for the endpoint.

✓ Because it takes time for the registration to propagate, the endpoint isn't immediately available for use. For Azure CDN from Akamai profiles, propagation usually completes within one minute. For Azure CDN from Verizon profiles, propagation usually completes within 90 minutes, but in some cases can take longer. Be sure to review the Troubleshooting pages reference link.

For more information, you can see:

Create a new CDN endpoint - **https://docs.microsoft.com/en-us/azure/cdn/cdn-create-new-endpoint#create-a-new-cdn-endpoint**[19]

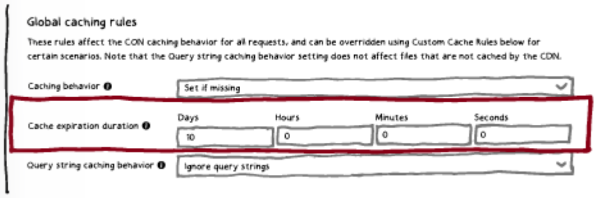
Troubleshooting CDN endpoints returning 404 statuses - **https://docs.microsoft.com/en-us/azure/cdn/cdn-troubleshoot-endpoint**

CDN Time-to-Live

Any publicly accessible blob content can be cached in Azure CDN until its time-to-live (TTL) elapses. The TTL is determined by Cache-directive headers in the HTTP response from the origin server. If the Cache-Control header does not provide the TTL information or if you prefer, you can configure caching rules to set the **Cache Expiration Duration**.

- **Global caching rules**. You can set the Cache Expiration Duration for each endpoint in your profile, which affects all requests to the endpoint. TTL is configured as days, hours, minutes, and seconds.

[19] https://docs.microsoft.com/en-us/azure/cdn/cdn-create-new-endpoint



Global caching rules

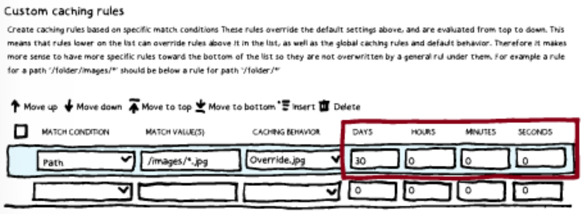
These rules affect the CDN caching behavior for all requests, and can be overridden using Custom Cache Rules below for certain scenarios. Note that the Query string caching behavior setting does not affect files that are not cached by the CDN.

Caching behavior: Set if missing

Cache expiration duration: Days 10, Hours 0, Minutes 0, Seconds 0

Query string caching behavior: Ignore query strings

- **Custom caching rules**. You can also create custom caching rules for each endpoint in your profile. Custom caching rules match specific paths and file extensions, are processed in order, and override the global caching rule.



Custom caching rules

Create caching rules based on specific match conditions These rules override the default settings above, and are evaluated from top to down. This means that rules lower on the list can override rules above it in the list, as well as the global caching rules and default behavior. Therefore it makes more sense to have more specific rules toward the bottom of the list so they are not overwritten by a general rul under them. For example a rule for a path '/folder/images/*' should be below a rule for path '/folder/*'

□	MATCH CONDITION	MATCH VALUE(S)	CACHING BEHAVIOR	DAYS	HOURS	MINUTES	SECONDS
	Path	/images/*.jpg	Override.jpg	30	0	0	0
				0	0	0	0

For more information, you can see:

Cache-directive Headers - **https://docs.microsoft.com/en-us/azure/cdn/cdn-how-caching-works#-cache-directive-headers**[20]

Control Azure CDN caching behavior with caching rules - **https://docs.microsoft.com/en-us/azure/cdn/cdn-caching-rules**

CDN Compression

File compression is a simple and effective method to improve file transfer speed and increase page-load performance by reducing a file's size before it is sent from the server. File compression can reduce bandwidth costs and provide a more responsive experience for your users.

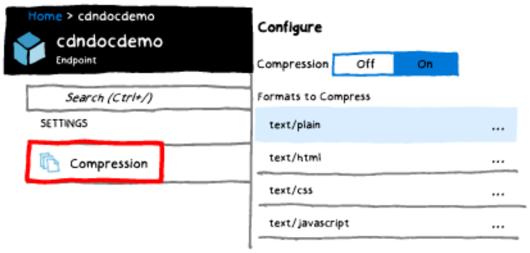
There are two ways to enable file compression:

- Enable compression on your origin server. In this case, the CDN passes along the compressed files and delivers them to clients that request them.
- Enable compression directly on the CDN edge servers. In this case, the CDN compresses the files and serves them to end users.

Enabling compression in the standard tiers

In the Azure portal, you can enable Compression and modify the MIME types list to tune which content formats to compress.

[20] https://docs.microsoft.com/en-us/azure/cdn/cdn-how-caching-works



✓ Although it is possible, it is not recommended to apply compression to compressed formats. For example, ZIP, MP3, MP4, or JPG.

For more information, you can see:

Premium tier compression - **https://docs.microsoft.com/en-us/azure/cdn/cdn-improve-performance#premium-tier**[21]

Compression behavior tables - **https://docs.microsoft.com/en-us/azure/cdn/cdn-improve-performance#compression-behavior-tables**[22]

Troubleshooting CDN file compression - **https://docs.microsoft.com/en-us/azure/cdn/cdn-troubleshoot-compression**

Practice: Optimize Your Content Delivery with Azure CDN

After you watch the videos, take a few minutes to try out the Azure content Delivery Network (CDN). In this **Quickstart**[23], you enable Azure Content Delivery Network (CDN) by creating a new CDN profile and CDN endpoint. You'll need to first create a storage account named mystorageacct123, which you will use for the origin hostname. You'll perform the following tasks:

- **Create a new CDN profile**[24].
- **Create a new CDN endpoint**[25].

Set Azure CDN caching rules

In this **tutorial**[26], try creating some caching rules to set of modify the default cache expiration behavior of your CDN profile, such as a URL path and file extension. You'll perform the following tasks:

- **Access the azure CDN caching rules page**[27].

[21] https://docs.microsoft.com/en-us/azure/cdn/cdn-improve-performance
[22] https://docs.microsoft.com/en-us/azure/cdn/cdn-improve-performance
[23] https://docs.microsoft.com/en-us/azure/cdn/cdn-create-new-endpoint
[24] https://docs.microsoft.com/en-us/azure/cdn/cdn-create-new-endpoint
[25] https://docs.microsoft.com/en-us/azure/cdn/cdn-create-new-endpoint
[26] https://docs.microsoft.com/en-us/azure/cdn/cdn-caching-rules-tutorial
[27] https://docs.microsoft.com/en-us/azure/cdn/cdn-caching-rules-tutorial

- **Set global caching rules**[28].
- **Set custom caching rules**[29].

For more information, you can see:

Troubleshooting CDN endpoints returning 404 statuses - **https://docs.microsoft.com/en-us/azure/cdn/cdn-troubleshoot-endpoint**

How caching works – **https://docs.microsoft.com/en-us/azure/cdn/cdn-how-caching-works**

Control Azure CDN caching behavior – **https://docs.microsoft.com/en-us/azure/cdn/cdn-caching-rules**

[28] https://docs.microsoft.com/en-us/azure/cdn/cdn-caching-rules-tutorial

[29] https://docs.microsoft.com/en-us/azure/cdn/cdn-caching-rules-tutorial

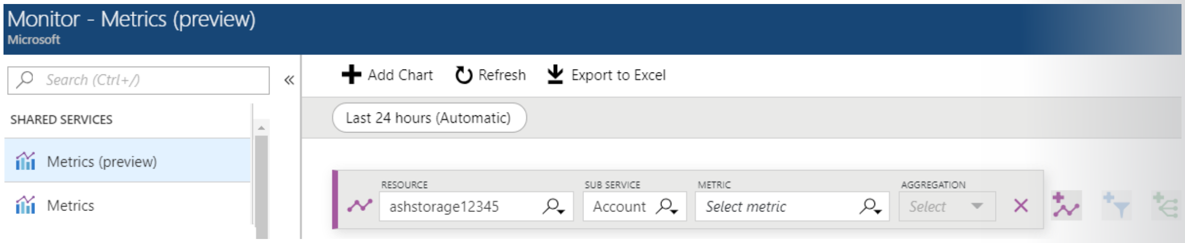
Monitoring Storage

Monitor Metrics

Azure Monitor provides unified user interfaces for monitoring across different Azure services. Azure Storage integrates Azure Monitor by sending metric data to the Azure Monitor platform. With metrics on Azure Storage, you can analyze usage trends, trace requests, and diagnose issues with your storage account.

Azure Monitor provides multiple ways to access metrics. You can access them from the Azure Portal, Monitor APIs (REST, and .Net) and analysis solutions such as the Operation Management Suite and Event Hubs. Metrics are enabled by default, and you can access the past 30 days of data. If you need to retain data for a longer period, you can archive metrics data to an Azure Storage account.

Metrics is a Shared Service where you can specify the resource, sub-service, metric, and aggregation criteria. Additionally, you specify more than one metric, filter by a metric, and Export to Excel.


Monitor - Metrics (preview)
Microsoft

RESOURCE: ashstorage12345
SUB SERVICE: Account
METRIC: Select metric
AGGREGATION: Select

✓ Take a minute to locate the Monitor service and then Metrics on the Shared Services blade.

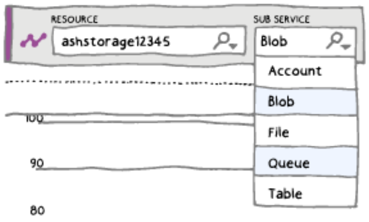
For more information, you can see:

Azure Storage metrics in Azure Monitor - **https://docs.microsoft.com/en-us/azure/storage/common/storage-metrics-in-azure-monitor?toc=%2fazure%2fstorage%2fblobs%2ftoc.json**

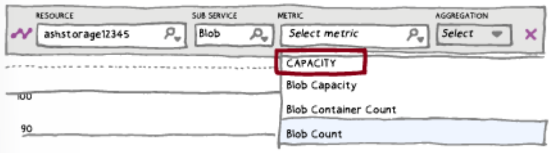
Monitoring Azure applications and resources - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview**

Capacity and Transaction Metrics

Let's look at an example of how the Monitor service can help you review your storage information. When you select a Storage Account resource your sub-service choice is: Account, Blob, File, Queue, and Table.


RESOURCE: ashstorage12345
SUB SERVICE: Blob

Capacity metrics values are sent to Azure Monitor every hour. The values are refreshed daily. Your sub-service selection determines what Capacity metrics are available. For example, if you choose the Blob sub-service, then the Capacity metrics are: Blob Capacity, Blob Container Count, and Blob Count.



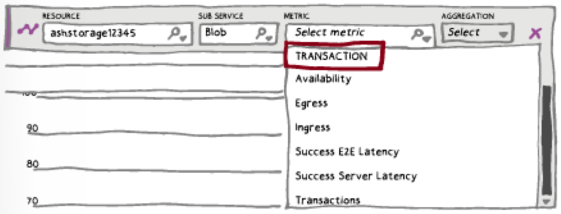
RESOURCE: ashstorage12345

SUB SERVICE: Blob

METRIC: Select metric

AGGREGATION: Select

Transaction metrics are sent from Azure Storage to Azure Monitor every minute. All transaction metrics are available at both account and service level (Blob storage, Table storage, Azure Files, and Queue storage).



RESOURCE: ashstorage12345

SUB SERVICE: Blob

METRIC: Select metric

AGGREGATION: Select

✓ Take few minutes to read about the Capacity and Transaction metrics and create a few metric graphs in the Azure portal.

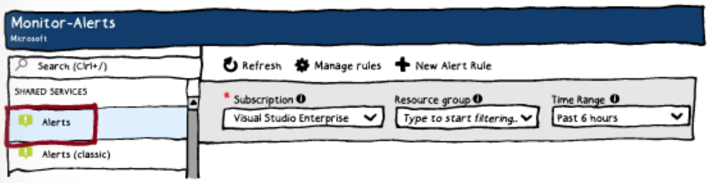
For more information, you can see:

Capacity Metrics - **https://docs.microsoft.com/en-us/azure/storage/common/storage-metrics-in-azure-monitor?toc=%2fazure%2fstorage%2fblobs%2ftoc.json#capacity-metrics**[30]

Transaction Metrics - **https://docs.microsoft.com/en-us/azure/storage/common/storage-metrics-in-azure-monitor?toc=%2fazure%2fstorage%2fblobs%2ftoc.json#transaction-metrics**[31]

Azure Monitor Alerts

Alerts has a new experience. The older alerts experience is now under the Alerts (Classic) tab.



Monitor-Alerts

Microsoft

* Subscription: Visual Studio Enterprise

Resource group: Type to start filtering...

Time Range: Past 6 hours

The new Alerts experience has many benefits.

- **Better notification system**. All newer alerts use action groups, which are named groups of notifications and actions that can be reused in multiple alerts.
- **A unified authoring experience**. All alert creation for metrics, logs and activity log across Azure Monitor, Log Analytics, and Application Insights is in one place.

[30] https://docs.microsoft.com/en-us/azure/storage/common/storage-metrics-in-azure-monitor?toc=%2fazure%2fstorage%2fblobs%2ftoc.json

[31] https://docs.microsoft.com/en-us/azure/storage/common/storage-metrics-in-azure-monitor?toc=%2fazure%2fstorage%2fblobs%2ftoc.json

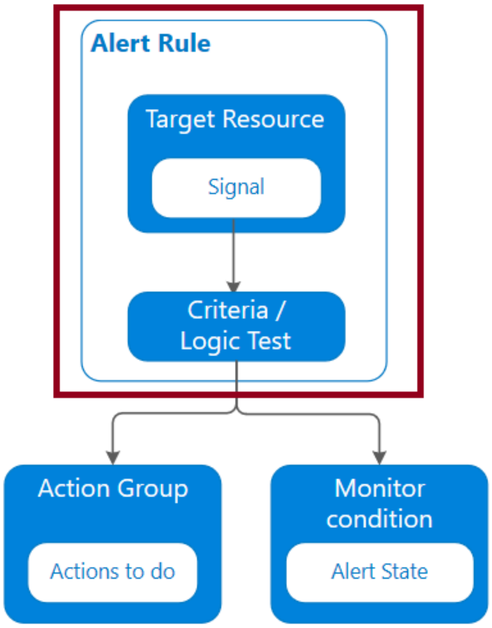
- **View Log Analytics alerts in Azure portal**. You can now also see Log Analytics alerts in your subscription. Previously these were in a separate portal.
- **Separation of Fired Alerts and Alert Rules**. Alert Rules (the definition of condition that triggers an alert), and Fired Alerts (an instance of the alert rule firing) are differentiated, so the operational and configuration views are separated.
- **Better workflow**. The new alerts authoring experience guides the user along the process of configuring an alert rule, which makes it simpler to discover the right things to get alerted on.

For more information, you can see:

The new alerts experience in Azure Monitor - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-unified-alerts**

Alert Rules

Alerts proactively notify you when important conditions are found in your monitoring data. They allow you to identify and address issues before the users of your system notice them. Alerts consists of alert rules, action groups, and monitor conditions.



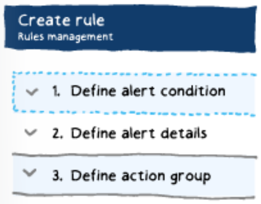
Alert rules are separated from alerts and the actions that are taken when an alert fires. The alert rule captures the target and criteria for alerting. The alert rule can be in an enabled or a disabled state. Alerts only fire when enabled. The key attributes of an alert rule are:

- **Target Resource** – Defines the scope and signals available for alerting. A target can be any Azure resource. Example targets: a virtual machine, a storage account, a virtual machine scale set, a Log Analytics workspace, or an Application Insights resource. For certain resources (like Virtual Machines), you can specify multiple resources as the target of the alert rule.
- **Signal** – Signals are emitted by the target resource and can be of several types. Metric, Activity log, Application Insights, and Log.
- **Criteria** – Criteria is a combination of Signal and Logic applied on a Target resource. Examples: + Percentage CPU > 70%; Server Response Time > 4 ms; and Result count of a log query > 100.

- **Alert Name** – A specific name for the alert rule configured by the user.
- **Alert Description** – A description for the alert rule configured by the user.
- **Severity** – The severity of the alert once the criteria specified in the alert rule is met. Severity can range from 0 to 4.
- **Action** – A specific action taken when the alert is fired. See the Action Groups topic coming up.

Alerts can be authored in a consistent manner regardless of the monitoring service or signal type. All alerts fired and related details are available in single page.

Authoring an alert is a three-step task where the user first picks a target for the alert, followed by selecting the right signal and then specifying the logic to be applied on the signal as part of the alert rule.



Create rule
Rules management

1. Define alert condition
2. Define alert details
3. Define action group

1. Define alert condition includes:

- **Target selection**. For example, storage account.

- **Alert criteria**. For example, Used Capacity.

- **Alert logic**. For example, over a six-hour period whenever the Used Capacity is over 1000000 bytes.

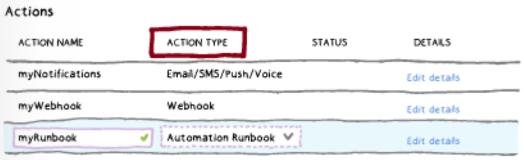
1. **Define alert details** includes: Alert rule name, description, and severity. There are five severity levels, Severity 0 to Severity 4.

2. **Define action group**. Create an action group to notify your team via email and text messages or automate actions using webhooks and runbooks.

✓ Take a few minutes to create an alert rule and look at the options.

Action Groups

Action groups enable you to configure a list of actions to take when the alert is triggered. Action groups ensure that the same actions are taken each time an alert is triggered. There are several action types you can select when defining the group: Select Email/**SMS**[32]/Push/Voice, **Logic App**[33], **Webhook**[34], **IT Service Management**[35], or Automation Runbook.

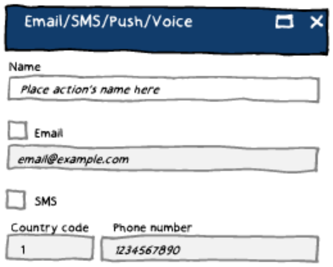


Actions

ACTION NAME	ACTION TYPE	STATUS	DETAILS
myNotifications	Email/SMS/Push/Voice		Edit details
myWebhook	Webhook		Edit details
myRunbook	Automation Runbook		Edit details

[32] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-activity-log-alerts-webhook
[33] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-action-groups-logic-app
[34] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-activity-log-alerts-webhook
[35] https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-itsmc-overview

Each action type is different in the details that must be provided. Here is a screenshot for the Email and SMS configuration.



Email/SMS/Push/Voice

Name: Place action's name here

□ Email: email@example.com

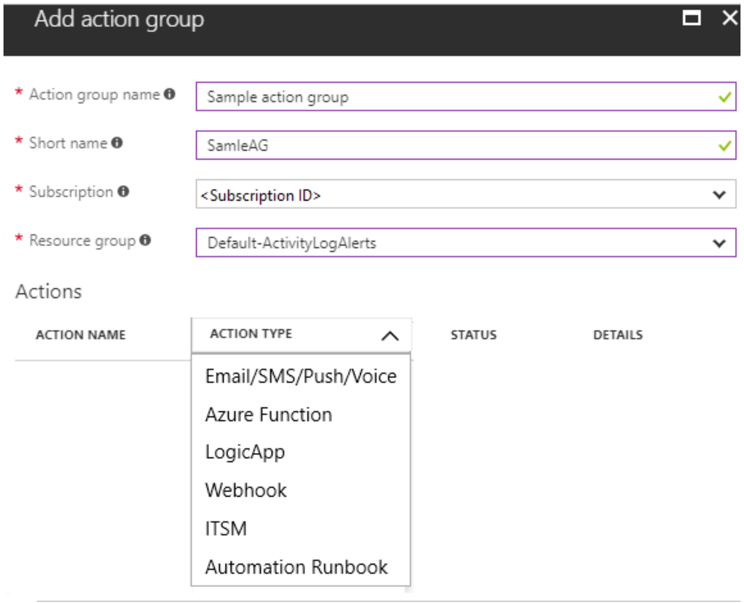
□ SMS

Country code: 1

Phone number: 1234567890

An action group is a collection of notification preferences defined by the owner of an Azure subscription. Azure Monitor and Service Health alerts use action groups to notify users that an alert has been triggered. Various alerts may use the same action group or different action groups depending on the user's requirements.

When an action is configured to notify a person by email or SMS the person will receive a confirmation indicating he / she has been added to the action group.



Add action group

* Action group name: Sample action group

* Short name: SamleAG

* Subscription: <Subscription ID>

* Resource group: Default-ActivityLogAlerts

Actions

ACTION NAME	ACTION TYPE	STATUS	DETAILS

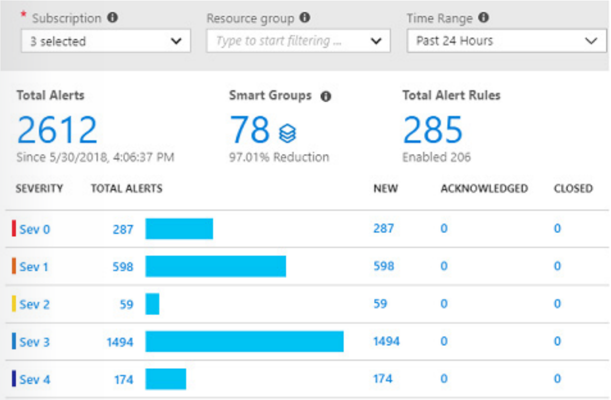
- **Email** – Emails will be sent to the email addresses. Ensure that your email filtering is configured appropriately. You may have up to 1000 email actions in an Action Group.
- **ITSM** – You may have up to 10 ITSM actions in an Action Group ITSM Action requires an ITSM Connection.
- **Logic App** – You may have up to 10 Logic App actions in an Action Group.
- **Function App** – The function keys for Function Apps configured as actions are read through the Functions API.
- **Runbook** – You may have up to 10 Runbook actions in an Action Group.
- **SMS** – You may have up to 10 SMS actions in an Action Group.

- **Voice** – You may have up to 10 Voice actions in an Action Group.
- **Webhook** – You may have up to 10 Webhook actions in an Action Group. Retry logic - The timeout period for a response is 10 seconds. The webhook call will be retried a maximum of 2 times when the following HTTP status codes are returned: 408, 429, 503, 504 or the HTTP endpoint does not respond. The first retry happens after 10 seconds. The second and last retry happens after 100 seconds.

✓ You may have up to 10 Azure app actions in an Action Group. At this time the Azure app action only supports ServiceHealth alerts.

Alerts Experience

The default Alerts page provides a summary of alerts that are created within a particular time window. It displays the total alerts for each severity with columns that identify the total number of alerts in each state for each severity.



Total Alerts 2612 Since 5/30/2018, 4:06:37 PM

Smart Groups 78 97.01% Reduction

Total Alert Rules 285 Enabled 206

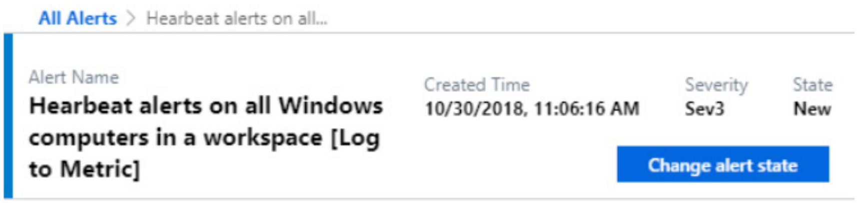
SEVERITY	TOTAL ALERTS	NEW	ACKNOWLEDGED	CLOSED
Sev 0	287	287	0	0
Sev 1	598	598	0	0
Sev 2	59	59	0	0
Sev 3	1494	1494	0	0
Sev 4	174	174	0	0

Column	Description
Subscription	Select up to five Azure subscriptions. Only alerts in the selected subscriptions are included in the view.
Resource group	Select a single resource group. Only alerts with targets in the selected resource group are included in the view.
Time range	Only alerts fired within the selected time window are included in the view. Supported values are the past hour, the past 24 hours, the past 7 days, and the past 30 days.

✓ You can select Total Alerts, Smart Groups, and Total Alert Rules to open a new page.

Alert Detail Page

The Alert detail page is displayed when you select an alert. It provides details of the alert and enables you to change its state.



All Alerts > Hearbeat alerts on all...

Alert Name: Hearbeat alerts on all Windows computers in a workspace [Log to Metric]

Created Time: 10/30/2018, 11:06:16 AM

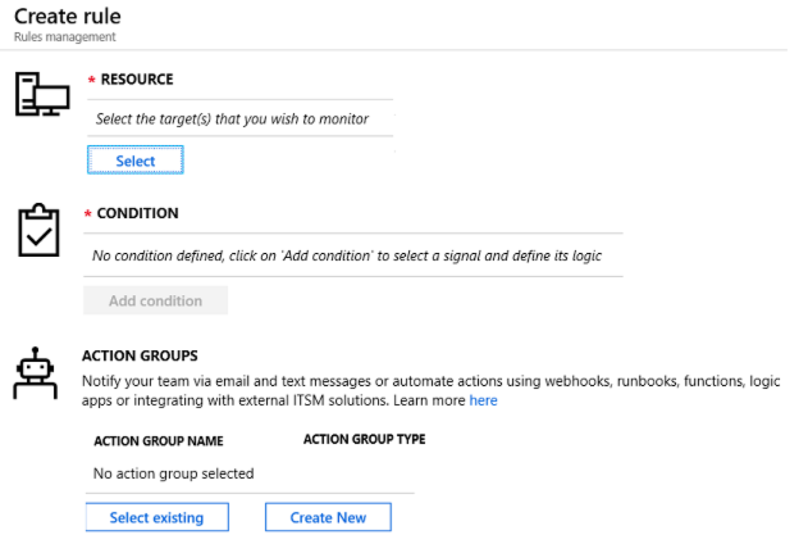
Severity: Sev3

State: New

Section	Description
Essentials	Displays the properties and other significant information about the alert.
History	Lists each action taken by the alert and any changes made to the alert. Currently limited to state changes.
Smart group	Information about the smart group the alert is included in. The alert count refers to the number of alerts that are included in the smart group. Includes other alerts in the same smart group that were created in the past 30 days regardless of the time filter in the alerts list page. Select an alert to view its detail.
More details	Displays further contextual information for the alert, which is typically specific to the type of source that created the alert.

Create and Alert

Alerts can be authored in a consistent manner regardless of the monitoring service or signal type. All fired alerts and related details are available in single page. You create a new alert rule with the following three steps:



Create rule

Rules management

* RESOURCE

Select the target(s) that you wish to monitor

* CONDITION

No condition defined, click on 'Add condition' to select a signal and define its logic

ACTION GROUPS

Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions. Learn more here

ACTION GROUP NAME	ACTION GROUP TYPE
No action group selected	

- **Resource**. Select the resource you want to monitor. For example, resource group, virtual machine, or storage account.

- **Condition**. Select the signal and define its logic. The signal could be All, Metrics, or Activity log.
- **Action Group**. Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions.
- **Alert rule name**. Specify a name to identify your alert.
- **Description**. Provide a description for your alert rule.
- **Enable rule upon creation**. You can enable and disable your alert rules.

✓ We currently support configuring only two metrics signals or one log search signal or one activity log signal per alert rule. An alert will be triggered when the conditions for all the above configured criteria are met.

Demonstration - Alerts

In this demonstration, we will create an alert rule.

Create an alert rule

1. In Azure portal, click on **Monitor**. The Monitor blade consolidates all your monitoring settings and data in one view.
2. Click **Alerts** then click **+ New alert rule**. As most resource blades also have Alerts in their resource menu under Monitoring, you could create alerts from there as well.

Explore alert targets

1. Click **Select** under Target, to select a target resource that you want to alert on. Use **Subscription** and **Resource type** drop-downs to find the resource you want to monitor. You can also use the search bar to find your resource.
2. If the selected resource has metrics you can create alerts on, Available signals on the bottom right will include metrics. You can view the full list of resource types supported for metric alerts in this article.
3. Click **Done** when you have made your selection.

Explore alert conditions

1. Once you have selected a target resource, click on **Add condition**.
2. You will see a list of signals supported for the resource, select the metric you want to create an alert on.
3. Optionally, refine the metric by adjusting Period and Aggregation. If the metric has dimensions, you will see the Dimensions table presented.
4. You will see a chart for the metric for the last 6 hours. Adjust the **Show history** drop-down.
5. Define the **Alert logic**. This will determine the logic which the metric alert rule will evaluate.
6. If you are using a static threshold, the metric chart can help determine what might be a reasonable threshold. If you are using a Dynamic Thresholds, the metric chart will display the calculated thresholds based on recent data.
7. Click **Done**.
8. Optionally, add another criteria if you want to monitor a complex alert rule.

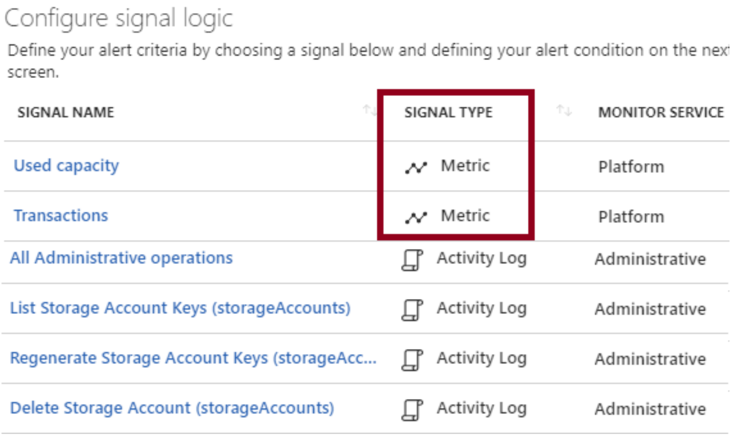
Explore alert details

1. Fill in Alert details like **Alert Rule Name**, **Description** and **Severity**.

2. Add an action group to the alert either by selecting an existing action group or creating a new action group.
3. Click **Done** to save the metric alert rule.

Signal Types and Metrics

Signals are emitted by the Target resource and can be of several types. Metric, Activity log, Application Insights, and Log are supported Signal types.



Configure signal logic

Define your alert criteria by choosing a signal below and defining your alert condition on the next screen.

SIGNAL NAME	SIGNAL TYPE	MONITOR SERVICE
Used capacity	Metric	Platform
Transactions	Metric	Platform
All Administrative operations	Activity Log	Administrative
List Storage Account Keys (storageAccounts)	Activity Log	Administrative
Regenerate Storage Account Keys (storageAcc...	Activity Log	Administrative
Delete Storage Account (storageAccounts)	Activity Log	Administrative

Newer metric alerts specifically have the following improvements:

- **Improved latency**: Newer metric alerts can run as frequently as every minute. Log alerts still have a longer than 1-minute delay due to the time is takes to ingest the logs.
- **Support for multi-dimensional metrics**: You can alert on dimensional metrics allowing you to monitor an interesting segment of the metric.
- **More control over metric conditions**: You can define richer alert rules. The newer alerts support monitoring the maximum, minimum, average, and total values of metrics.
- **Combined monitoring of multiple metrics**: You can monitor multiple metrics (currently, up to two metrics) with a single rule. An alert is triggered if both metrics breach their respective thresholds for the specified time-period.
- **Metrics from Logs** (limited public preview): Some log data going into Log Analytics can now be extracted and converted into Azure Monitor metrics and then alerted on just like other metrics.

For more information, you can see:

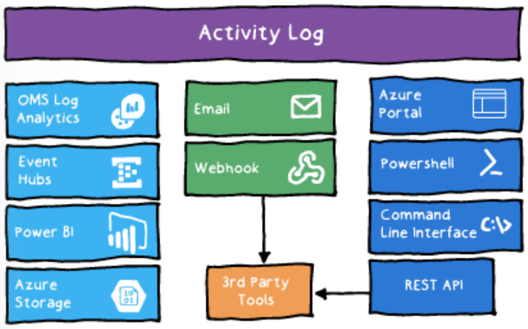
Alert rule terminology - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-unified-alerts#alert-rules-terminology**[36]

Activity Log

The **Azure Activity Log** is a subscription log that provides insight into subscription-level events that have occurred in Azure. This includes a range of data, from Azure Resource Manager operational data to updates on Service Health events. The Activity Log was previously known as “Audit Logs” or “Operational Logs”.

[36] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-unified-alerts

Using the Activity Log, you can determine the 'what, who, and when' for any write operation taken on the resources in your subscription. For example, who stopped a service. It provides an audit trail of the activities or operations performed on your resources by someone working on the Azure platform. You can also understand the status of the operation and other relevant properties.

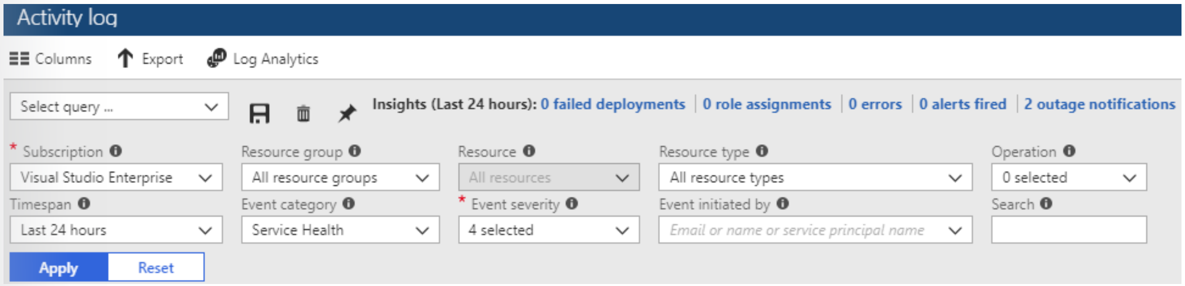


This diagram shows many of the things you can do with the activity log including:

- Send data to Log Analytics for advanced search and alerts.
- Query or manage events in the Portal, PowerShell, CLI, and REST API.
- Stream information to Event Hub.
- Archive data to a storage account.
- Analyze data with Power BI.

✓ The Activity Log differs from **Diagnostic Logs**[37]. Activity Logs provide data about the operations on a resource from the outside (the "control plane"). Diagnostics Logs are emitted by a resource and provide information about the operation of that resource (the "data plane").

Query the Activity Log



Select query ...

Insights (Last 24 hours): 0 failed deployments | 0 role assignments | 0 errors | 0 alerts fired | 2 outage notifications

* Subscription	Resource group	Resource	Resource type	Operation
Visual Studio Enterprise	All resource groups	All resources	All resource types	0 selected
Timespan	Event category	* Event severity	Event initiated by	Search
Last 24 hours	Service Health	4 selected	Email or name or service principal name	

In the Azure portal, you can filter your Activity Log by these fields:

- **Subscription**. One or more Azure subscription names.
- **Resource group**. One or more resource groups within those subscriptions.
- **Resource (name)**. The name of a specific resource.
- **Resource type**. The type of resource, for example, Microsoft.Compute/virtualmachines.

[37] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-of-diagnostic-logs

- **Operation name**. The name of an Azure Resource Manager operation, for example, Microsoft.SQL/servers/Write.
- **Timespan**. The start and end time for events.
- **Category**. The event category is described in the next topic.
- **Severity**. The severity level of the event (Informational, Warning, Error, Critical).
- **Event initiated by**. The 'caller,' or user who performed the operation.
- **Search** - This is an open text search box that searches for that string across all fields in all events.

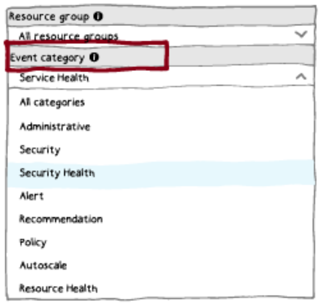
Once you have defined a set of filters, you can save it as a query that is persisted across sessions if you ever need to perform the same query with those filters applied again in the future. You can also pin a query to your Azure dashboard to always keep an eye on specific events.

For more information, you can see:

Query the Activity Log in the Azure portal - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs#query-the-activity-log-in-the-azure-portal**[38]

Event Categories

The Activity Log provides several event categories. You may select one or more.



Resource group: All resource groups

Event category: Service Health

- **Administrative** - This category contains the record of all create, update, delete, and action operations performed through Resource Manager. Examples of the types of events you would see in this category include "create virtual machine" and "delete network security group". The Administrative category also includes any changes to role-based access control in a subscription.
- **Service Health** - This category contains the record of any service health incidents that have occurred in Azure. An example of the type of event you would see in this category is "SQL Azure in East US is experiencing downtime." Service health events come in five varieties: Action Required, Assisted Recovery, Incident, Maintenance, Information, or Security.
- **Alert** - This category contains the record of all activations of Azure alerts. An example of the type of event you would see in this category is "CPU % on myVM has been over 80 for the past 5 minutes."
- **Autoscale** - This category contains the record of any events related to the operation of the autoscale engine based on any autoscale settings you have defined in your subscription. An example of the type of event you would see in this category is "Autoscale scale up action failed."

[38] https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs

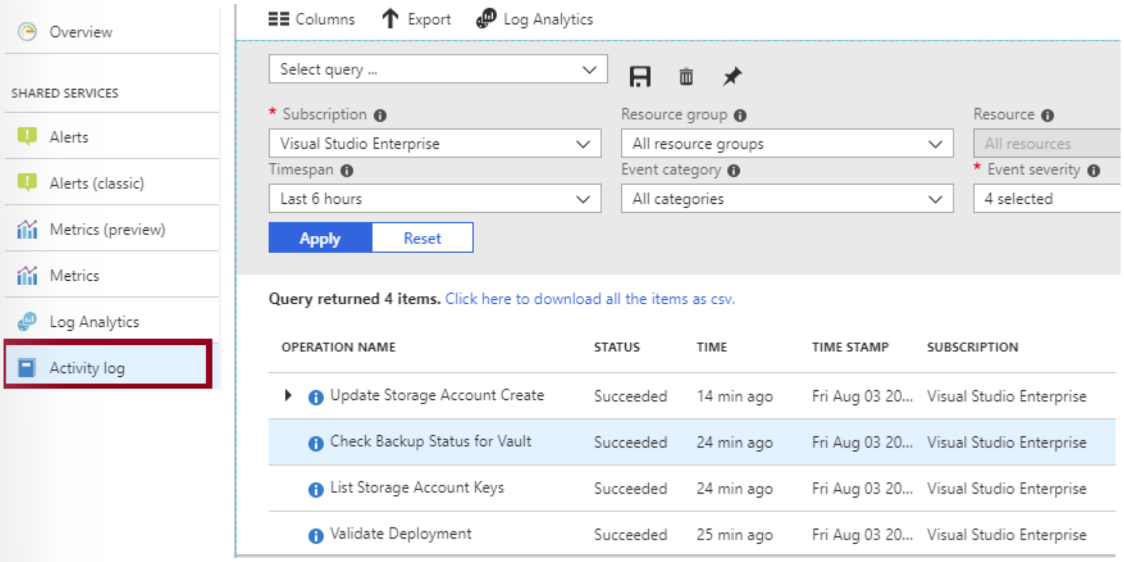
- **Recommendation** - This category contains recommendation events from certain resource types, such as web sites and SQL servers. These events offer recommendations for how to better utilize your resources.
- **Security** - This category contains the record of any alerts generated by Azure Security Center. An example of the type of event you would see in this category is "Suspicious double extension file executed."
- **Policy and Resource Health** - These categories do not contain any events; they are reserved for future use.

For more information, you can see:

Categories in the Activity Log - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs#categories-in-the-activity-log**

Activity Log and Log Analytics

It is easy to access the Activity Log Analytics solution.



* Subscription: Visual Studio Enterprise

Resource group: All resource groups

Resource: All resources

Timespan: Last 6 hours

Event category: All categories

* Event severity: 4 selected

Query returned 4 items. Click here to download all the items as csv.

OPERATION NAME	STATUS	TIME	TIME STAMP	SUBSCRIPTION
Update Storage Account Create	Succeeded	14 min ago	Fri Aug 03 20...	Visual Studio Enterprise
Check Backup Status for Vault	Succeeded	24 min ago	Fri Aug 03 20...	Visual Studio Enterprise
List Storage Account Keys	Succeeded	24 min ago	Fri Aug 03 20...	Visual Studio Enterprise
Validate Deployment	Succeeded	25 min ago	Fri Aug 03 20...	Visual Studio Enterprise

With the Azure Activity Logs tile, you can do many things:

- Analyze the activity logs with pre-defined views.
- Analyze and search activity logs from multiple Azure subscriptions.
- Keep activity logs for longer than the default of 90 days.
- Correlate activity logs with other Azure platform and application data.
- See operational activities aggregated by status.
- View trends of activities happening on each of your Azure services.
- Report on authorization changes on all your Azure resources.
- Identify outage or service health issues impacting your resources.
- Use Log Search to correlate user activities, auto-scale operations, authorization changes, and service health to other logs or metrics from your environment.

✓ Log Analytics collects activity logs free of charge and stores the logs for 90 days free of charge. If you store logs for longer than 90 days, you will incur data retention charges for the data stored longer than 90 days. When you're on the Free pricing tier, activity logs do not apply to your daily data consumption.

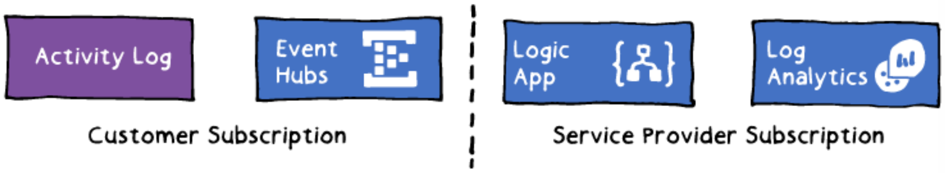
For more information, you can see:

Collect and analyze Azure activity logs in Log Analytics - **https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-activity**

Collect Across Subscriptions

This topic covers the strategy to collect Azure Activity Logs into a Log Analytics workspace using the Azure Log Analytics Data Collector connector for Logic Apps. Use this strategy when you need to send logs to a workspace in a different Azure Active Directory. For example, if you are a managed service provider, you may want to collect activity logs from a customer's subscription and store them in a Log Analytics workspace in your own subscription.

The basic strategy is to have Azure Activity Log send events to an **Event Hub**[39] where a **Logic App**[40] sends them to your Log Analytics workspace.



Advantages of this approach include:

- Low latency since the Azure Activity Log is streamed into the Event Hub. The Logic App is then triggered and posts the data to Log Analytics.
- Minimal code is required, and there is no server infrastructure to deploy.

✓ Do you think your organization would benefit from this strategy?

For more information, you can see:

Collect Azure Activity Logs into Log Analytics across subscriptions - **https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-activity-logs-subscriptions**

[39] https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-what-is-event-hubs
[40] https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview

Review Questions

Module 2 Review Questions

High Availability

An organization deploy a business-critical application. The application must always be available to users, even if a region-wide outage occurs.

You need to recommend a replication solution.

What replication options are available? Which option should you recommend and why?

Suggested Answer ↓

The following table provides a quick overview of the scope of durability and availability that each replication strategy will provide you for a given type of event (or event of similar impact).

Replication Option	LRS	ZRS	GRS	RA-GRS
Node unavailability within a data center	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes	Yes
A region-wide outage	No	No	Yes	Yes
Read access to your data (in a remote, geo-replicated region) in the event of region-wide unavailability	No	No	No	Yes

In this scenario, only the Geo-Replicated Storage option provides the functionality desired.

Storage Accounts

You have an Azure file share that supports a project. The file share uses storage of type General-purpose v1. Users frequently access and modify the files.

The project ends. You need to archive the content.

What should you do? How will your actions affect costs for the resource?

Suggested Answer ↓

You should convert the GPv1 storage account to GPv2 to take advantage of the Archive access tier. General-purpose v2 (GPv2) accounts are storage accounts that support all of the latest features for blobs, files, queues, and tables. GPv2 accounts support all APIs, services, and features supported by General-purpose v1 (GPv1) and Blob storage accounts. They also retain the same durability, availability, scalability, and performance provided by all storage account types.

Pricing for GPv2 accounts has been designed to deliver the lowest per gigabyte prices, and industry competitive transaction prices.

You can upgrade your GPv1 or Blob storage account to a GPv2 account using Azure portal, PowerShell, or Azure CLI.

For block blobs in a GPv2 storage account, you can choose between hot or cool storage access tiers at the account level and between hot, cool, or archive access tiers at the blob level based on usage patterns. Store frequently, infrequently, and rarely accessed data in the hot, cool, and archive storage tiers respectively to optimize storage and transaction costs.

In this scenario, upgrading the storage account will reduce Azure costs.

Azure Storage Explorer

You are an architect for a managed service provider.

You need to be able to manage and maintain the Azure storage environment.

Which tool should you use? How should you connect to client resources?

Suggested Answer ↓

Microsoft Azure Storage Explorer is a standalone app from Microsoft that allows you to easily work with Azure Storage data.

Some of the benefits of Azure Storage Explorer are:

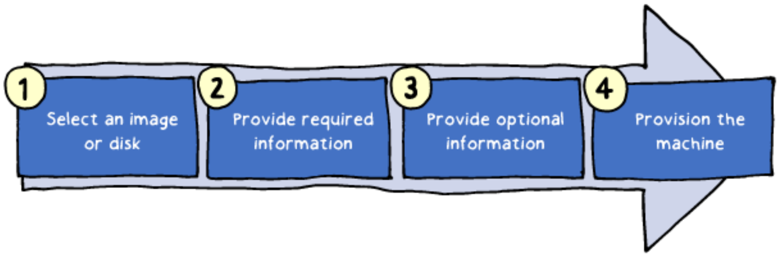
- Access multiple accounts and subscriptions.
- Create, delete, view, and edit storage resources.
- View and edit Blob, Queue, Table, File, Cosmos DB storage and Data Lake Storage.
- Obtain shared access signature (SAS) keys.
- Available for Windows, Mac, and Linux.

Module 3 Module Deploying and Managing Virtual Machines (VMs)

Creating Virtual Machines

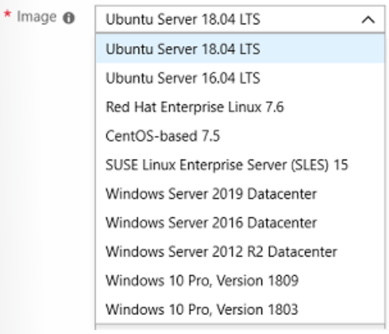
Creating Virtual Machines (Portal)

There are many methods for deploying virtual machines. No matter what method you use, these are the basic steps for deploying a virtual machine.



1. **Select an image or disk** to use for your new virtual machine. The image is from the Marketplace. The disk is a VHD you have created.
2. **Provide required information** such as host name, user name, and password for the new virtual machine.
3. **Provide optional information** like domain membership, virtual networks, storage account, cloud service, and availability set.
4. **Provision the machine**.

When you are creating virtual machines in the portal, one of your first decisions is the image to use. Azure supports Windows and Linux operating systems. There are server and client platforms.

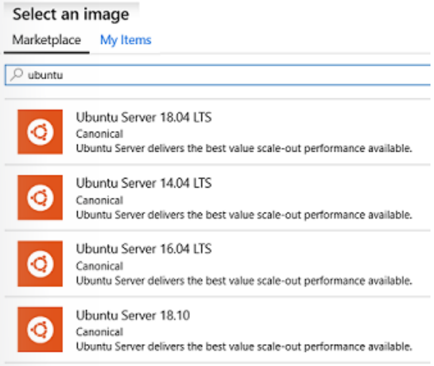


* Image

Ubuntu Server 18.04 LTS

- Ubuntu Server 18.04 LTS
- Ubuntu Server 16.04 LTS
- Red Hat Enterprise Linux 7.6
- CentOS-based 7.5
- SUSE Linux Enterprise Server (SLES) 15
- Windows Server 2019 Datacenter
- Windows Server 2016 Datacenter
- Windows Server 2012 R2 Datacenter
- Windows 10 Pro, Version 1809
- Windows 10 Pro, Version 1803

Additional images are available by searching the Marketplace.



Select an image

Marketplace My Items

ubuntu

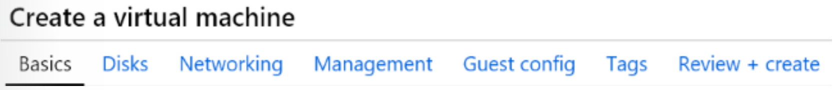
Ubuntu Server 18.04 LTS
Canonical
Ubuntu Server delivers the best value scale-out performance available.

Ubuntu Server 14.04 LTS
Canonical
Ubuntu Server delivers the best value scale-out performance available.

Ubuntu Server 16.04 LTS
Canonical
Ubuntu Server delivers the best value scale-out performance available.

Ubuntu Server 18.10
Canonical
Ubuntu Server delivers the best value scale-out performance available.

After selecting your image the portal will guide you through additional configuration informatin.



Create a virtual machine

Basics Disks Networking Management Guest config Tags Review + create

Basic - Project details, Administrator account, Inbound port rules

Disks - OS disk type, data disks

Networking - Virtual networks, load balancing

Management - Monitoring, Auto-shutdown, Backup

Guest config - Add additional configuration, agents, scripts or applications via virtual machine extensions or cloud-init.

Windows Virtual Machines

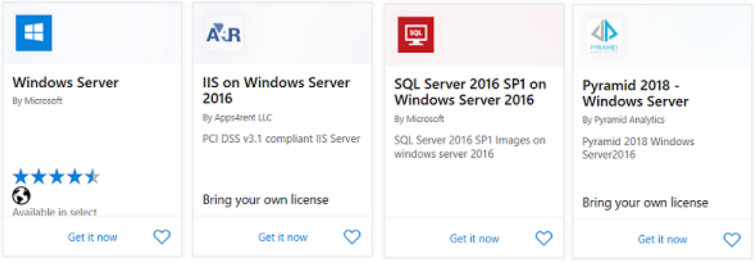
Windows Server is the operating system that bridges on-premises environments with Azure services enabling hybrid scenarios and maximizing existing investments, including:

- Unique hybrid capabilities with Azure to extend your datacenter and maximize investments, such as the Azure Hybrid Benefits.
- Advanced multi-layer security to help you elevate your security posture
- Faster innovation for applications enabling Developers and IT Pros to create new and modernize their apps, and
- Unprecedented Hyper-converged Infrastructure to evolve your datacenter infrastructure

Terms of Use

Your use of the Windows Server images from Azure Marketplace Virtual Machine Gallery are provided to you for use with virtual machine instances under your Azure subscription which are governed by the Online Services Terms. These virtual machine instances are limited for use with Azure.

Latest Images


Windows Server
By Microsoft
★★★★★
Available in select
Get it now
IIS on Windows Server 2016
By Apps4rent LLC
PCI DSS v3.1 compliant IIS Server
Bring your own license
Get it now
SQL Server 2016 SP1 on Windows Server 2016
By Microsoft
SQL Server 2016 SP1 Images on windows server 2016
Get it now
Pyramid 2018 - Windows Server
By Pyramid Analytics
Pyramid 2018 Windows Server2016
Bring your own license
Get it now

- Windows Server 2019 is the latest Long-Term Servicing Channel (LTSC) release with five years of mainstream support + five years of extended support. Choose the image that is right for your application needs: 1) Server with Desktop Experience includes all roles including the graphical user interface (GUI), 2) Server Core omits the GUI for a smaller OS footprint, or 3) Containers option includes the Server with Desktop Experience, plus ready-made container images.
  - Windows Server 2019 Datacenter - Server with Desktop Experience
  - Windows Server 2019 Datacenter - with Containers
  - Windows Server 2019 Datacenter - Server Core
  - Windows Server 2019 Datacenter - Server Core with Containers

Windows Server Semi-Annual Channel releases deliver new operating system capabilities at a faster pace and are based on the Server Core installation option of the Datacenter edition. A new release comes out every six months and is supported for 18 months. Check the Lifecycle Support Page for support dates and always use the latest release if possible.

✓ There are also a large number of Windows Server 2016 and Windows Server 2012 images.

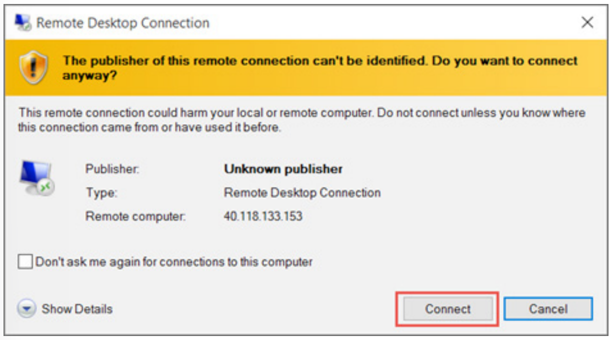
For more information, you can see:

Windows Virtual Machines Documentation - **https://docs.microsoft.com/en-us/azure/virtual-machines/windows/**

Windows VM Connections

To manage an Azure Windows VM, you can use the same set of tools that you used to deploy it. However, you will also want to interact with an operating system (OS) running within the VM. The methods you can use to accomplish this are OS-specific and include the following options:

- **Remote Desktop Protocol (RDP)** allows you to establish a graphical user interface (GUI) session to an Azure VM that runs any supported version of Windows. The Azure portal automatically enables the **Connect button** on the Azure Windows VM blade if the VM is running and accessible via a public or private IP address, and if it accepts inbound traffic on TCP port 3389. After you click this button, the portal will automatically provision an .rdp file, which you can either open or download. Opening the file initiates an RDP connection to the corresponding VM. You will get a warning that the .rdp file is from an unknown publisher. This is expected. When connecting be sure to credentials for the virtual machine. The Azure PowerShell **Get-AzRemoteDesktopFile** cmdlet provides the same functionality.



Remote Desktop Connection

The publisher of this remote connection can't be identified. Do you want to connect anyway?

This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.

Publisher: **Unknown publisher**
Type: Remote Desktop Connection
Remote computer: 40.118.133.153

□ Don't ask me again for connections to this computer

-

- **Windows Remote Management (WinRM)** allows you to establish a command-line session to an Azure VM that runs any supported version of Windows. You can also use WinRM to run noninteractive Windows PowerShell scripts. WinRM facilitates additional session security by using certificates. You can upload a certificate that you intend to use to Azure Key Vault prior to establishing a session. The process of setting up WinRM connectivity includes the following, high-level steps:
  - Creating a key vault.
  - Creating a self-signed certificate.
  - Uploading the certificate to the key vault.
  - Identifying the URL of the certificate uploaded to the key vault.
  - Referencing the URL in the Azure VM configuration.

WinRM uses by TCP port 5986 by default, but you can change it to a custom value. In either case, you must ensure that no network security groups are blocking inbound traffic on the port that you choose.

For more information, you can see:

Setting up WinRM access for Virtual Machines in Azure Resource Manager: **https://aka.ms/ljezi1**

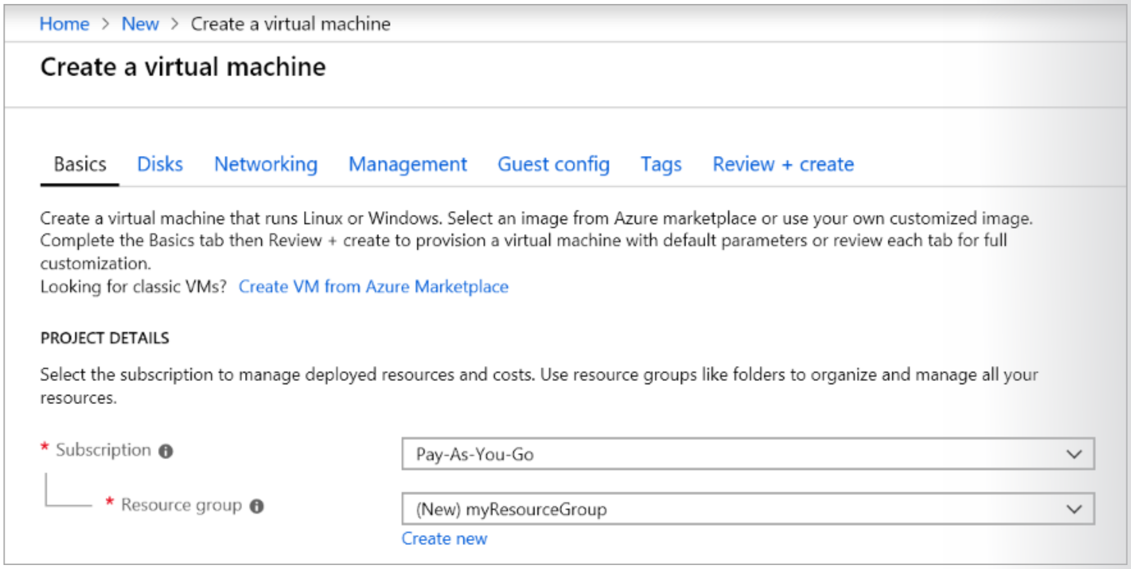
Demonstration - Creating a VM in the Portal

In this demonstration, we will create and access a Windows virtual machine in the portal.

Create the virtual machine

1. Choose **Create a resource** in the upper left-hand corner of the Azure portal.

2. In the search box above the list of Azure Marketplace resources, search for **Windows Server 2016 Datacenter**. After locating the image, click **Create**.
3. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose to **Create new** resource group. Type *myResourceGroup* for the name.



Home > New > Create a virtual machine

Create a virtual machine

Basics Disks Networking Management Guest config Tags Review + create

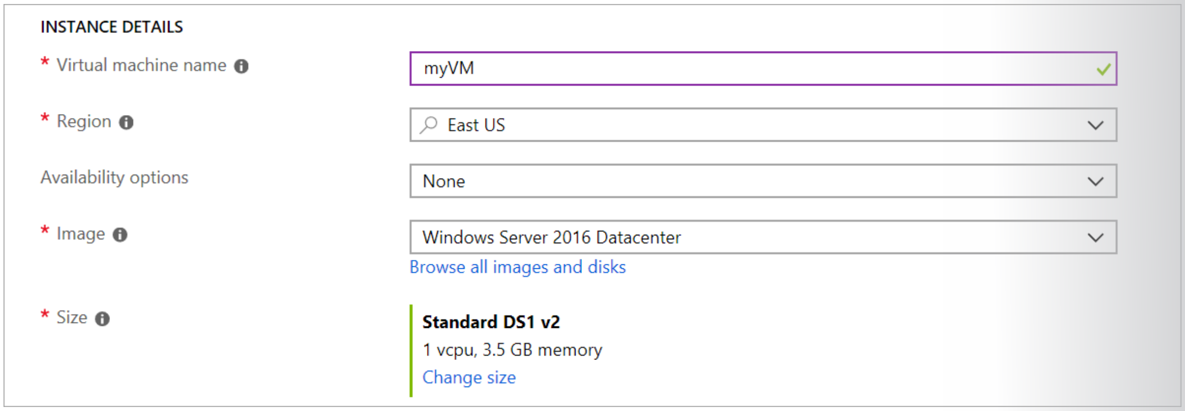
Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.
Looking for classic VMs? Create VM from Azure Marketplace

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription	Pay-As-You-Go
* Resource group	(New) myResourceGroup Create new

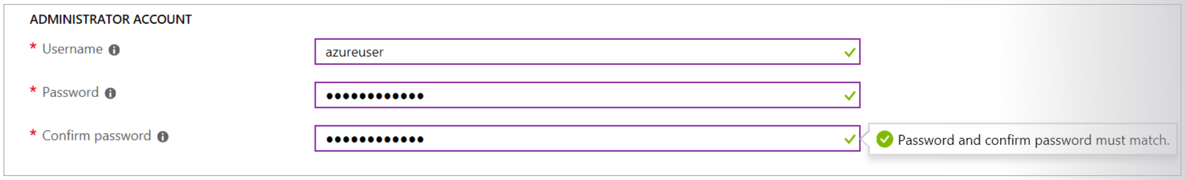
4.
5. Under **Instance details**, type *myVM* for the **Virtual machine name** and choose *East US* for your **Location**. Leave the other defaults.



INSTANCE DETAILS

* Virtual machine name	myVM
* Region	East US
Availability options	None
* Image	Windows Server 2016 Datacenter Browse all images and disks
* Size	**Standard DS1 v2** 1 vcpu, 3.5 GB memory Change size

6.
7. Under **Administrator account**, provide a username, such as *azureuser* and a password. The password must be at least 12 characters long and meet the defined complexity requirements.

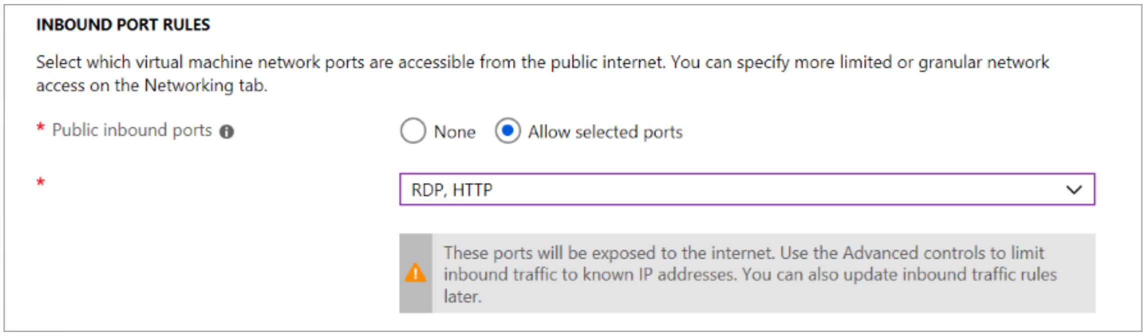


ADMINISTRATOR ACCOUNT

* Username	azureuser
* Password	••••••••••••
* Confirm password	••••••••••••

Password and confirm password must match.

8.
9. Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** and **HTTP** from the drop-down.



INBOUND PORT RULES

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

* Public inbound ports ○ None ● Allow selected ports

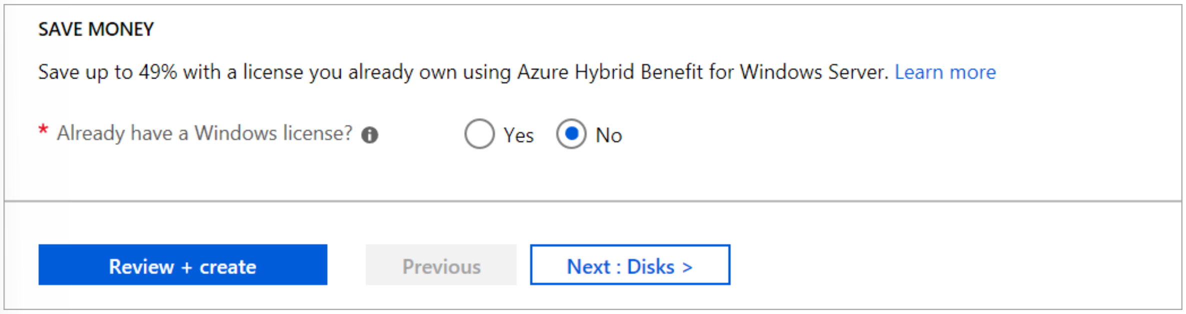
* RDP, HTTP

These ports will be exposed to the internet. Use the Advanced controls to limit inbound traffic to known IP addresses. You can also update inbound traffic rules later.

10.

11. Move to the **Management** tab, and under **Monitoring** turn **Off** Boot Diagnostics. This will eliminate validation errors.

12. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page. Wait for the validation, then click **Create**.



SAVE MONEY

Save up to 49% with a license you already own using Azure Hybrid Benefit for Windows Server. Learn more

* Already have a Windows license? ○ Yes ● No

13.

Connect to the virtual machine

Create a remote desktop connection to the virtual machine. These directions tell you how to connect to your VM from a Windows computer. On a Mac, you need to install an RDP client from the Mac App Store.

1. Select the **Connect** button on the virtual machine properties page.
2. In the **Connect to virtual machine** page, keep the default options to connect by DNS name over port 3389 and click **Download RDP file**.
3. Open the downloaded RDP file and select **Connect** when prompted.
4. In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the username as localhost\username, enter password you created for the virtual machine, and then select **OK**.
5. You may receive a certificate warning during the sign-in process. Select **Yes** or **Continue** to create the connection.

Install web server

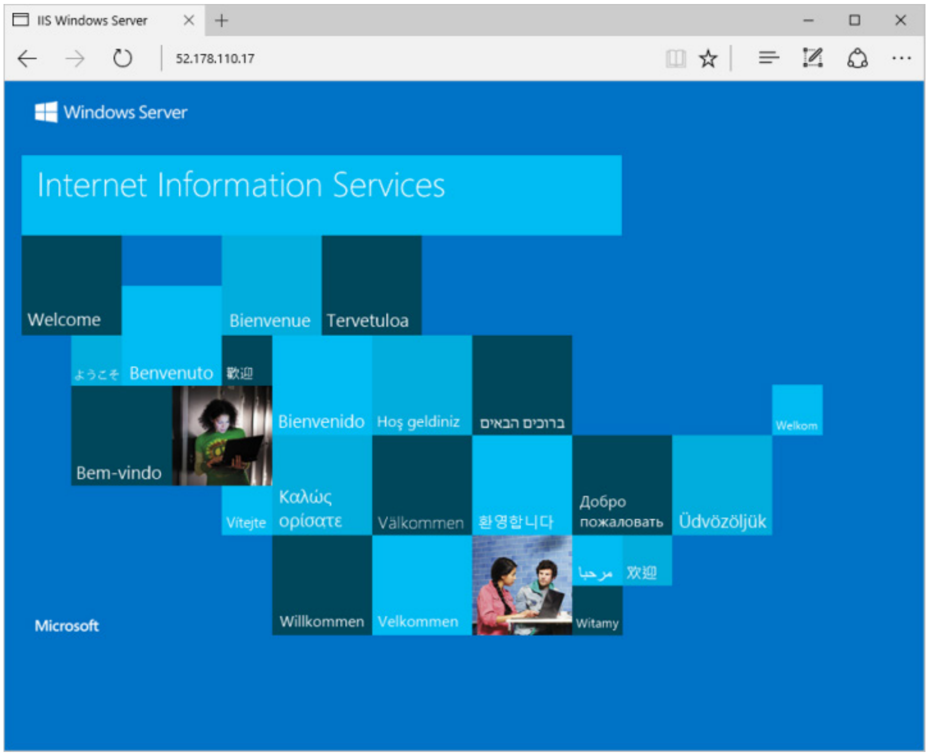
To see your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

When done, close the RDP connection to the VM.

View the IIS welcome page

In the portal, select the VM and in the overview of the VM, use the **Click to copy** button to the right of the public IP address to copy it and paste it into a browser tab. The default IIS welcome page will open, and should look like this:

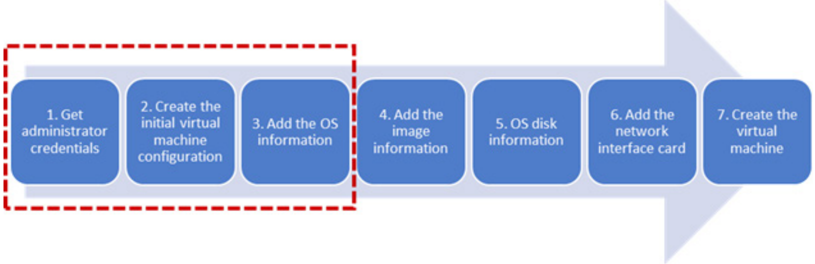


Clean up resources

✓ When no longer needed, you can delete the resource group, virtual machine, and all related resources. To do so, select the resource group for the virtual machine, select **Delete**, then confirm the name of the resource group to delete.

PowerShell - Example (Part 1)

You can also create a virtual machine using PowerShell. In this example, a virtual machine is created with a name of *myVM* running the latest version of Windows Server 2016 Datacenter.


1. Get administrator credentials
2. Create the initial virtual machine configuration
3. Add the OS information
4. Add the image information
5. OS disk information
6. Add the network interface card
7. Create the virtual machine

1. Set the username and password needed for the administrator account on the virtual machine with **Get-Credential**:

```
$cred = Get-Credential
```

2. Create the initial configuration for the virtual machine with **New-AzVMConfig**:

```
$vm = New-AzVMConfig -VMName myVM -VMSize Standard_D1
```

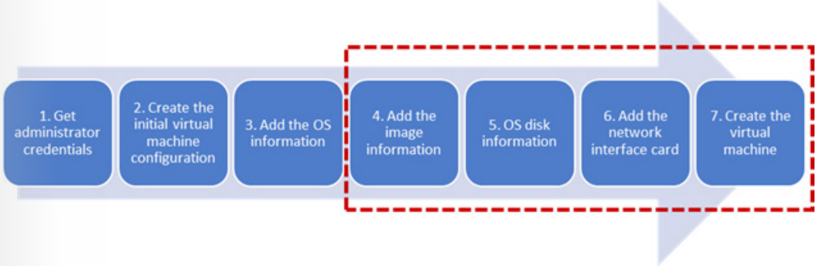
3. Add the operating system information to the virtual machine configuration with **Set-AzureRmVMOperatingSystem**:

```
$vm = Set-AzVMOperatingSystem `
-VM $vm `
-Windows `
-ComputerName myVM `
-Credential $cred `
-ProvisionVMAgent -EnableAutoUpdate
```

✓ Continues in the next topic.

PowerShell - Example (Part 2)

This continues the example on the previous page.


1. Get administrator credentials
2. Create the initial virtual machine configuration
3. Add the OS information
4. Add the image information
5. OS disk information
6. Add the network interface card
7. Create the virtual machine

1. Add the image information to the virtual machine configuration with **Set-AzVMSourceImage**:

```
$vm = Set-AzVMSourceImage `
-VM $vm `
-PublisherName MicrosoftWindowsServer `
-Offer WindowsServer `
-Skus 2016-Datacenter `
-Version latest
```

2. Add the operating system disk settings to the virtual machine configuration with **Set-AzVMOSDisk**:

```
$vm = Set-AzVMOSDisk `
-VM $vm `
-Name myOsDisk `
-DiskSizeInGB 128 `
-CreateOption FromImage `
-Caching ReadWrite
```

3. Add the network interface card that you previously created to the virtual machine configuration with **Add-AzureRmVMNetworkInterface**:

```
$vm = Add-AzVMNetworkInterface -VM $vm -Id $nic.Id
```

4. Create the virtual machine with **New-AzVM**.

```
New-AzVM -ResourceGroupName myResourceGroupVM -Location EastUS -VM $vm
```

Demonstration - Creating a Virtual Machine with PowerShell

In this demonstration, we will create a virtual machine using PowerShell.

Create the virtual machine

Note: You can use the Cloud Shell or a local version of PowerShell.

Note: There are many ways to create a virtual machine with PowerShell. This example is different from the one explained in the topic slides.

1. Launch the Cloud Shell.
2. Run this code:

```
# create a resource group
New-AzResourceGroup -Name myResourceGroup -Location EastUS

# create the virtual machine
# when prompted, provide a username and password to be used as the logon
credentials for the VM
New-AzVm `
    -ResourceGroupName "myResourceGroup" `
    -Name "myVM" `
    -Location "East US" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress" `
    -OpenPorts 80,3389
```

Verify the machine creation in the portal

1. Access the portal and view your virtual machines.
2. Verify **myVM** was created.
3. Review the VM settings.
4. Notice this is a Windows machine in a new VNet and subnet.
5. Notice the command started the machine.
6. At this point you could use either the portal or PowerShell to make changes.

Connect to the virtual machine

1. Retrieve the public IP address of the machine.

```
Get-AzPublicIpAddress -ResourceGroupName "myResourceGroup" | Select "IpAddress"
```

2. Create an RDP session from your local machine. Replace the IP address with the public IP address of your VM. This command runs from a cmd window.

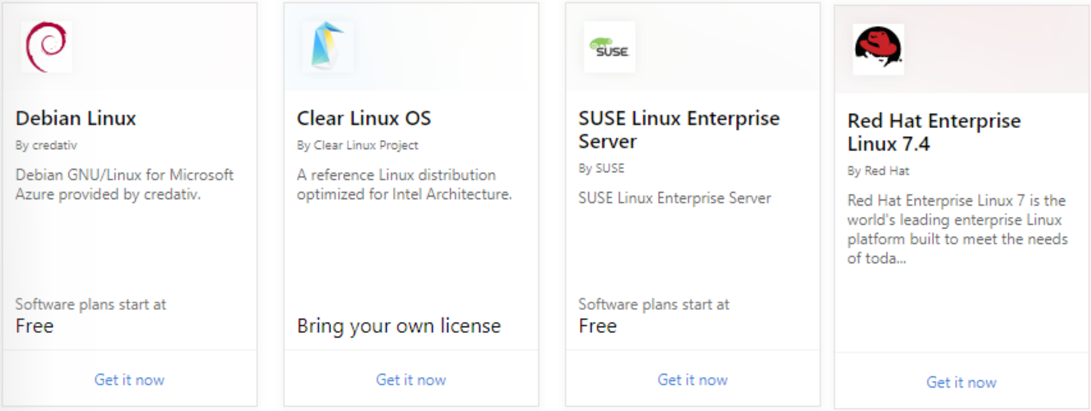
```
mstsc /v:publicIpAddress
```

3. When prompted, provide your login credentials for the machine. Be sure to **Use a different account**. Type the username as localhost\username, enter password you created for the virtual machine, and then select **OK**. You may receive a certificate warning during the sign-in process. Select **Yes** or **Continue** to create the connection
4. When done, close the RDP connection to the VM.
5. Clean up your resources. This will take a few minutes and remove the resource group and virtual machine.

```
Remove-AzResourceGroup -Name myResourceGroup
```

Linux Virtual Machines

Azure supports many Linux distributions and versions including CentOS by OpenLogic, Core OS, Debian, Oracle Linux, Red Hat Enterprise Linux, and Ubuntu.



Debian Linux
By credativ
Debian GNU/Linux for Microsoft Azure provided by credativ.
Software plans start at
Free
Get it now

Clear Linux OS
By Clear Linux Project
A reference Linux distribution optimized for Intel Architecture.
Bring your own license
Get it now

SUSE Linux Enterprise Server
By SUSE
SUSE Linux Enterprise Server
Software plans start at
Free
Get it now

Red Hat Enterprise Linux 7.4
By Red Hat
Red Hat Enterprise Linux 7 is the world's leading enterprise Linux platform built to meet the needs of toda...
Get it now

Here are a few things to know about the Linux distributions.

- There are hundreds of Linux images in the Azure Marketplace.
- Linux has the same deployment options as for Windows virtual machines: PowerShell (Resource Manager), Portal, and Command Line Interface. You can manage your Linux virtual machines with a host of popular open-source DevOps tools such as Puppet, and Chef.

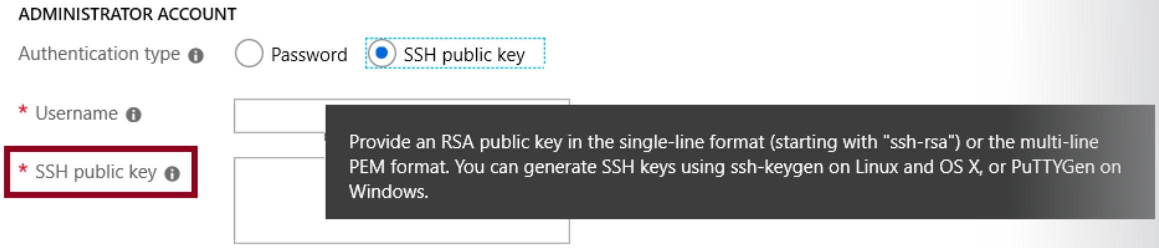
✓ Take a few minutes to look through the Marketplace at the Linux distributions. Are there any you are interested in?

For more information, you can see:

Linux virtual machines (Documentation) - **https://docs.microsoft.com/en-us/azure/virtual-machines/linux/**

Linux VM Connections

When you create a Linux VM, you can decide to authenticate with an **SSH public key** or **Password**.



ADMINISTRATOR ACCOUNT

Authentication type ○ Password ● SSH public key

* Username

* SSH public key

Provide an RSA public key in the single-line format (starting with "ssh-rsa") or the multi-line PEM format. You can generate SSH keys using ssh-keygen on Linux and OS X, or PuTTYGen on Windows.

What is SSH?

SSH is an encrypted connection protocol that allows secure sign-ins over unsecured connections. SSH is the default connection protocol for Linux VMs hosted in Azure. Although SSH itself provides an encrypted connection, using passwords with SSH connections still leaves the VM vulnerable to brute-force attacks or guessing of passwords. A more secure and preferred method of connecting to a VM using SSH is by using a public-private key pair, also known as SSH keys.

- The *public key* is placed on your Linux VM, or any other service that you wish to use with public-key cryptography.
- The *private key* remains on your local system. Protect this private key. Do not share it.

When you use an SSH client to connect to your Linux VM (which has the public key), the remote VM tests the client to make sure it possesses the private key. If the client has the private key, it's granted access to the VM.

Depending on your organization's security policies, you can reuse a single public-private key pair to access multiple Azure VMs and services. You do not need a separate pair of keys for each VM or service you wish to access.

Your public key can be shared with anyone, but only you (or your local security infrastructure) should possess your private key.

✓ Azure currently requires at least a 2048-bit key length and the SSH-RSA format for public and private keys.

Demonstration - Connect to Linux Virtual Machines

In this demonstration, we will create a Linux machine and access the machine with SSL.

Note: Ensure port 22 is open for the connection to work.

Create the SSH Keys

1. Download the PuTTY tool. This will include PuTTYgen - **https://putty.org/**.
2. Once installed, locate and open the **PuTTYgen** program.
3. In the **Parameters** option group choose **RSA**.
4. Click the **Generate** button.
5. Move your mouse around the blank area in the window to generate some randomness.
6. Copy the text of the **Public key for pasting into authorized keys file**.
7. Optionally you can specify a **Key passphrase** and then **Confirm passphrase.** You will be prompted for the passphrase when you authenticate to the VM with your private SSH key. Without a passphrase, if

someone obtains your private key, they can sign in to any VM or service that uses that key. We recommend you create a passphrase. However, if you forget the passphrase, there is no way to recover it.

8. Click **Save private key**.
9. Choose a location and filename and click **Save**. You will need this file to access the VM.

Create the Linux machine and assign the public SSH key

1. In the portal create a Linux machine of your choice.
2. Choose **SSH Public Key** for the **Authentication type** (instead of **Password**).
3. Provide a **Username**.
4. Paste the public SSH key from PuTTY into the **SSH public key** text area. Ensure the key validates with a checkmark.
5. Create the VM. Wait for it to deploy.
6. Access the running VM.
7. From the **Overview** blade, click **Connect**.
8. Make a note of your login information including user and public IP address.

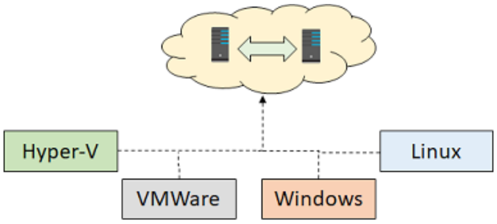
Access the server using SSH

1. Open the **PuTTY** tool.
2. Enter **username@publicIpAddress** where username is the value you assigned when creating the VM and publicIpAddress is the value you obtained from the Azure portal.
3. Specify **22** for the **Port**.
4. Choose **SSH** in the **Connection Type** option group.
5. Navigate to **SSH** in the Category panel, then click **Auth**.
6. Click the **Browse** button next to **Private key file for authentication**.
7. Navigate to the private key file saved when you generated the SSH keys and click **Open**.
8. From the main PuTTY screen click **Open.**
9. You will now be connected to your server command line.

Backup and Restore

Azure Site Recovery

You can use **Azure Site Recovery** to replicate on-premises physical or virtual machines running Windows or Linux. Azure Site Recovery includes support for both Hyper-V and VMware virtual machines. You can replicate data from your on-premises datacenter to Azure or to a secondary site. Orchestration is built in with Azure Site Recovery, which means that the management of replication, failover, and recovery is included. For example, should a virtual machine or service fail in your datacenter, you can use Azure Site Recovery to failover to the replicated resource in either Azure or your secondary site.



Azure Site Recovery works in the following three scenarios:

- **Hyper-V Virtual Machine Replication**. When Virtual Machine Manager (VMM) is used to manage Hyper-V virtual machines, you can use Azure Site Recovery to replicate them to Azure or to a secondary datacenter. If you do not use VMM to manage your virtual machines, you can use Azure Site Recovery to replicate them to Azure only.
- **VMware Virtual Machine Replication**. You can perform the replication of virtual machines by VMware to a secondary site that is also running VMware. You also can replicate to Azure.
- **Physical Windows and Linux machines**. You can replicate physical machines running either Windows or Linux to a secondary site or to Azure.

Azure Site Recovery Benefits

A migration to the cloud can result in significant business benefits. Here are some reasons to use Azure Site Recovery.

- **Eliminate the need for disaster recovery sites**. Your environment can be protected by automating the replication of the virtual machines based on policies that you set and control. Site Recovery is heterogeneous and can protect Hyper-V, VMware, and physical servers.
- **Reduce infrastructure costs**. Lower your on-premises infrastructure costs by using Azure as a secondary site for conducting business during outages. Or, eliminate datacenter costs altogether by moving to Azure and setting up disaster recovery between Azure regions. You can pre-assess network, storage, and compute resources needed to replicate applications from on-premises to Azure—and pay only for compute and storage resources needed to run apps in Azure during outages.
- **Automatically replicate to Azure**. Automate the orderly recovery of services in the event of a site outage at the primary datacenter. Automate the orderly recovery of services in the event of a site outage at the primary datacenter.
- **Safeguard against outages of complex workloads**. Protect applications in SQL Server, SharePoint, SAP, and Oracle.
- **Extend or boost capacity**. Applications can be Migrated to Azure with just a few clicks or burst to Azure temporarily when you encounter a surge in demand.

- **Continuous health monitoring**. Site Recovery monitors the state of your protected instances continuously and remotely from Azure. When replicating between two sites you control, your virtual machines' data and replication remains on your networks. All communication with Azure is encrypted.

✓ Are you considering using Azure Site Recovery and are you interested in any of these specific features? Which one is most important to you?

For more information, you can see:

Azure Site Recovery documentation - **https://azure.microsoft.com/en-us/services/site-recovery/**

Virtual Machine Data Protection

You can protect your data by taking backups at regular intervals. There are several backup options available for VMs, depending on your use-case.

Azure Backup

For backing up Azure VMs running production workloads, use Azure Backup. Azure Backup supports application-consistent backups for both Windows and Linux VMs. Azure Backup creates recovery points that are stored in geo-redundant recovery vaults. When you restore from a recovery point, you can restore the whole VM or just specific files. The topics in this lesson will focus on Azure Backup.

Azure Site Recovery

Azure Site Recovery protects your VMs from a major disaster scenario when a whole region experiences an outage due to major natural disaster or widespread service interruption. You can configure Azure Site Recovery for your VMs so that you can recover your application with a single click in matter of minutes. You can replicate to an Azure region of your choice.

Managed disk snapshots

In development and test environments, snapshots provide a quick and simple option for backing up VMs that use Managed Disks. A managed disk snapshot is a read-only full copy of a managed disk that is stored as a standard managed disk by default. With snapshots, you can back up your managed disks at any point in time. These snapshots exist independent of the source disk and can be used to create new managed disks. They are billed based on the used size. For example, if you create a snapshot of a managed disk with provisioned capacity of 64 GiB and actual used data size of 10 GiB, that snapshot is billed only for the used data size of 10 GiB.

Images

Managed disks also support creating a managed custom image. You can create an image from your custom VHD in a storage account or directly from a generalized (sysprepped) VM. This process captures a single image. This image contains all managed disks associated with a VM, including both the OS and data disks. This managed custom image enables creating hundreds of VMs using your custom image without the need to copy or manage any storage accounts.

Images versus snapshots

It's important to understand the difference between images and snapshots. With managed disks, you can take an image of a generalized VM that has been deallocated. This image includes all of the disks attached to the VM. You can use this image to create a VM, and it includes all of the disks.

- A snapshot is a copy of a disk at the point in time the snapshot is taken. It applies only to one disk. If you have a VM that has one disk (the OS disk), you can take a snapshot or an image of it and create a VM from either the snapshot or the image.
- A snapshot doesn't have awareness of any disk except the one it contains. This makes it problematic to use in scenarios that require the coordination of multiple disks, such as striping. Snapshots would need to be able to coordinate with each other and this is currently not supported.

✓ Have you tried any of these backup methods? Do you have a backup plan?

.

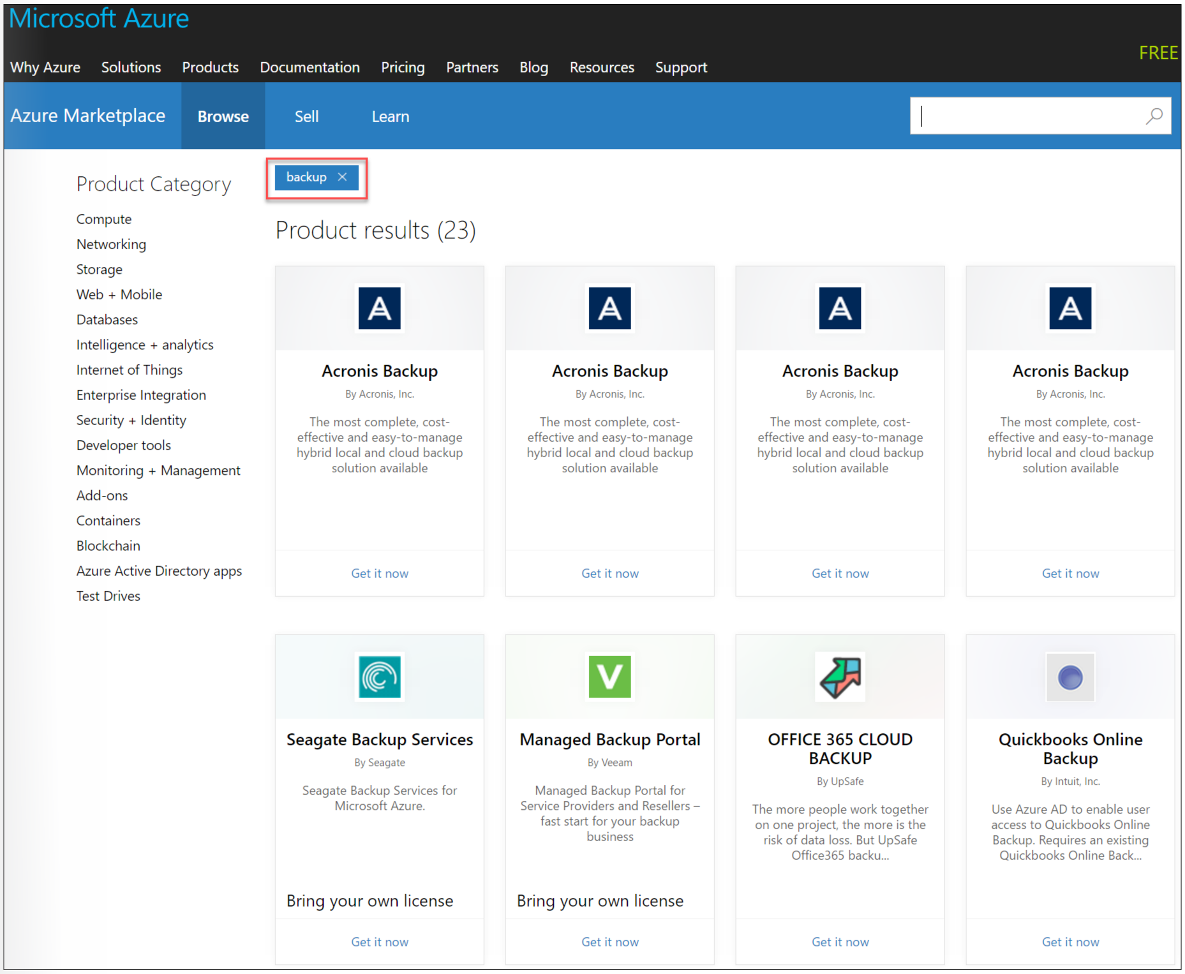
Workload Protection Needs

There are several methods for backing up virtual machines.

1. Enable backup for individual Azure VMs. When you enable backup, Azure Backup installs an extension to the Azure VM agent that's running on the VM. The agent backs up the entire VM.
2. Run the MARS agent on an Azure VM. This is useful if you want to back up individual files and folders on the VM.
3. Back up an Azure VM to a System Center Data Protection Manager (DPM) server or Microsoft Azure Backup Server (MABS) running in Azure. Then back up the DPM server/MABS to a vault using Azure Backup.

Often those that are new to deploying workloads in a public cloud do not consider how they will protect the workload once it is hosted there. This is, of course, a critical requirement for business continuity. Document how the workload is being protected today, including how often the workload is backed up, what types of backups are accomplished, and whether disaster recovery protection is in place for the workload. Options for workload protection include:

- Extending on-premises data protection solutions into Azure. In many cases, an organization can extend their backup strategy into Azure by choosing from many of the backup solutions available today in the Azure Marketplace.



Microsoft Azure

Why Azure Solutions Products Documentation Pricing Partners Blog Resources Support FREE

Azure Marketplace Browse Sell Learn

Product Category

- Compute
- Networking
- Storage
- Web + Mobile
- Databases
- Intelligence + analytics
- Internet of Things
- Enterprise Integration
- Security + Identity
- Developer tools
- Monitoring + Management
- Add-ons
- Containers
- Blockchain
- Azure Active Directory apps
- Test Drives

backup ×

Product results (23)

Acronis Backup
By Acronis, Inc.
The most complete, cost-effective and easy-to-manage hybrid local and cloud backup solution available
Get it now

Acronis Backup
By Acronis, Inc.
The most complete, cost-effective and easy-to-manage hybrid local and cloud backup solution available
Get it now

Acronis Backup
By Acronis, Inc.
The most complete, cost-effective and easy-to-manage hybrid local and cloud backup solution available
Get it now

Acronis Backup
By Acronis, Inc.
The most complete, cost-effective and easy-to-manage hybrid local and cloud backup solution available
Get it now

Seagate Backup Services
By Seagate
Seagate Backup Services for Microsoft Azure.
Bring your own license
Get it now

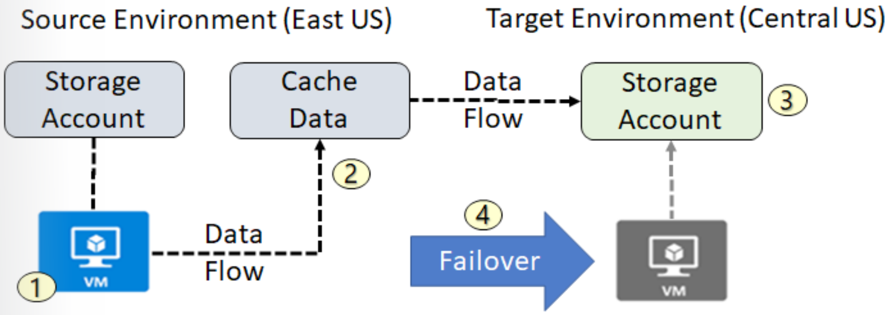
Managed Backup Portal
By Veeam
Managed Backup Portal for Service Providers and Resellers – fast start for your backup business
Bring your own license
Get it now

OFFICE 365 CLOUD BACKUP
By UpSafe
The more people work together on one project, the more is the risk of data loss. But UpSafe Office365 backu...
Get it now

Quickbooks Online Backup
By Intuit, Inc.
Use Azure AD to enable user access to Quickbooks Online Backup. Requires an existing Quickbooks Online Back...
Get it now

- Using native features in Azure to enable data protection, such as Azure Backup. Azure Backup is a native data protection service in Azure that allows for the protection of on-premises and Azure workloads.

Azure to Azure Architecture



When you enable replication for an Azure VM, the following happens:

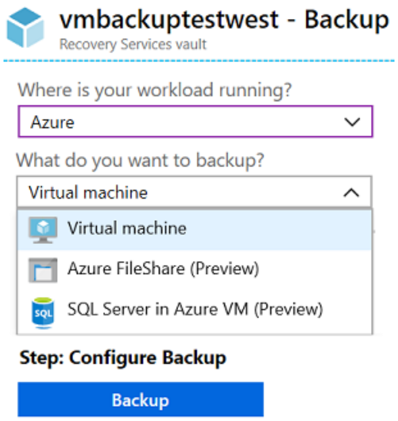
1. The Site Recovery Mobility service extension is automatically installed on the VM. The extension registers the VM with Site Recovery. Continuous replication begins for the VM. Disk writes are immediately transferred to the cache storage account in the source location.

2. Site Recovery processes the data in the cache, and sends it to the target storage account, or to the replica managed disks.
3. After the data is processed, crash-consistent recovery points are generated every five minutes. App-consistent recovery points are generated according to the setting specified in the replication policy.
4. When you initiate a failover, the VMs are created in the target resource group, target virtual network, target subnet, and in the target availability set. During a failover, you can use any recovery point.

Recovery Services Vault VM Backup Options

Recovery Services vault is a storage entity in Azure that houses data. The data is typically copies of data, or configuration information for virtual machines (VMs), workloads, servers, or workstations. You can use Recovery Services vaults to hold backup data for various Azure services such as IaaS VMs (Linux or Windows) and Azure SQL databases. Recovery Services vaults support System Center DPM, Windows Server, Azure Backup Server, and more. Recovery Services vaults make it easy to organize your backup data, while minimizing management overhead.

- The Recovery Services vault can be used to backup Azure virtual machines.



vmbackuptestwest - Backup
Recovery Services vault

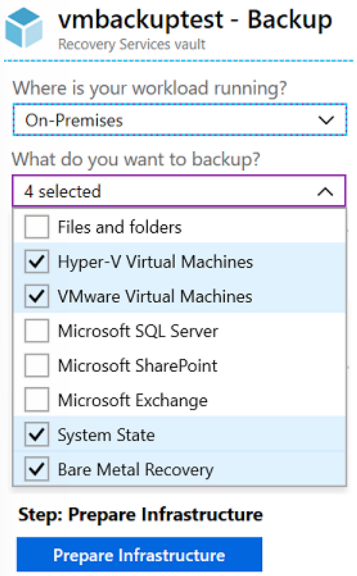
Where is your workload running?
Azure

What do you want to backup?
Virtual machine

Step: Configure Backup

-

- The Recovery Services vault can be used to backup on-premises virtual machines including: Hyper-V, VmWare, System State, and Bare Metal Recovery.



vmbackuptest - Backup
Recovery Services vault

Where is your workload running?
On-Premises

What do you want to backup?
4 selected

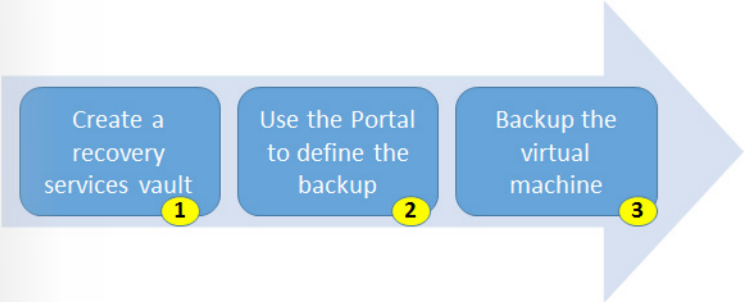
- ☐ Files and folders
- ☑ Hyper-V Virtual Machines
- ☑ VMware Virtual Machines
- ☐ Microsoft SQL Server
- ☐ Microsoft SharePoint
- ☐ Microsoft Exchange
- ☑ System State
- ☑ Bare Metal Recovery

Step: Prepare Infrastructure

-

Implementing VM Backups

Backing up Azure virtual machines using Azure Backup is easy and follows a simple process.


Create a recovery services vault
1
Use the Portal to define the backup
2
Backup the virtual machine
3

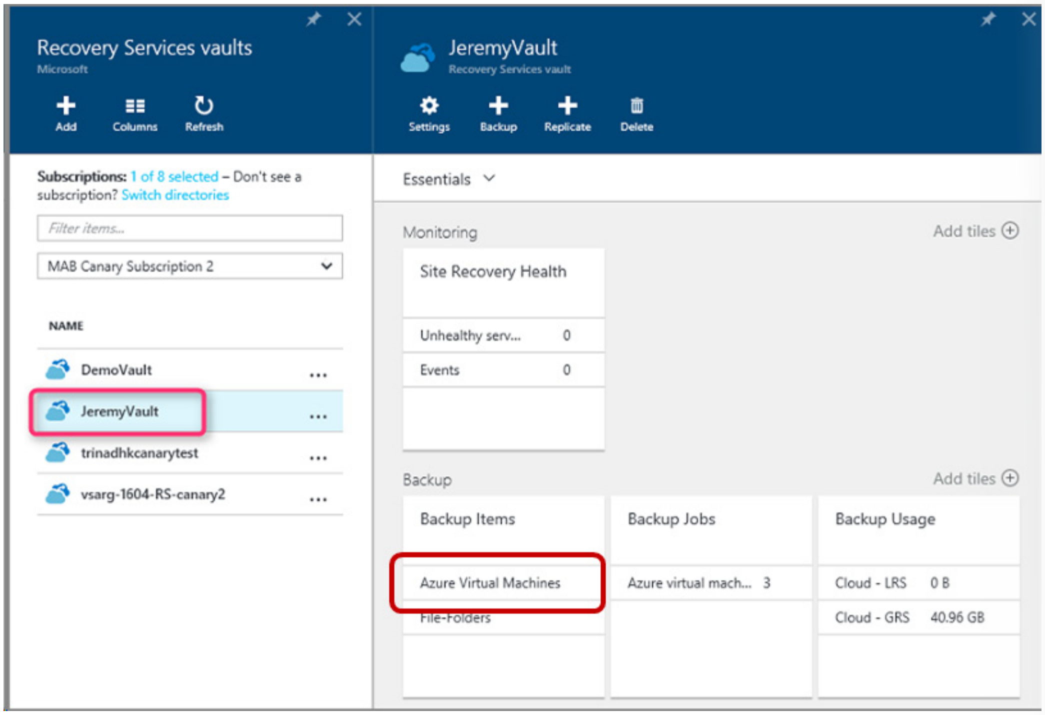
1. **Create a recovery services vault**. To back up your files and folders, you need to create a Recovery Services vault in the region where you want to store the data. You also need to determine how you want your storage replicated, either geo-redundant (default) or locally redundant. By default, your vault has geo-redundant storage. If you are using Azure as a primary backup storage endpoint, use the default geo-redundant storage. If you are using Azure as a non-primary backup storage endpoint, then choose locally redundant storage, which will reduce the cost of storing data in Azure.
2. **Use the Portal to define the backup**. Protect your data by taking snapshots of your data at defined intervals. These snapshots are known as recovery points, and they are stored in recovery services vaults. If or when it is necessary to repair or rebuild a VM, you can restore the VM from any of the saved recovery points. A backup policy defines a matrix of when the data snapshots are taken, and how long those snapshots are retained. When defining a policy for backing up a VM, you can trigger a backup job once a day.
3. **Backup the virtual machine**. The Azure VM Agent must be installed on the Azure virtual machine for the Backup extension to work. However, if your VM was created from the Azure gallery, then the VM Agent is already present on the virtual machine. VMs that are migrated from on-premises data centers would not have the VM Agent installed. In such a case, the VM Agent needs to be installed.

For more information, you can see:

Plan your VM backup infrastructure in Azure - **https://docs.microsoft.com/en-us/azure/backup/backup-azure-vms-introduction**

Implementing VM Restore

Once your virtual machine snapshots are safely in the recovery services vault it is easy to recover them.



Subscriptions: 1 of 8 selected – Don't see a subscription? Switch directories

Filter items...

MAB Canary Subscription 2

NAME
DemoVault
JeremyVault
trinadhkcanarytest
vsarg-1604-RS-canary2

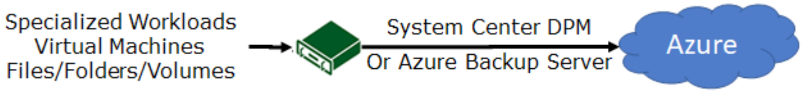
Site Recovery Health	
Unhealthy serv...	0
Events	0

Backup Items	Backup Jobs		Backup Usage	
Azure Virtual Machines	Azure virtual mach...	3	Cloud - LRS	0 B
File-Folders			Cloud - GRS	40.96 GB

Once you trigger the restore operation, the Backup service creates a job for tracking the restore operation. The Backup service also creates and temporarily displays notifications, so you monitor how the backup is proceeding.

Azure Backup Server

Another method of backing up virtual machines is using a Data Protection Manager (DPM) or Microsoft Azure Backup Server (MABS) server. This method can be used for specialized workloads, virtual machines, or files, folders, and volumes. Specialized workloads can include SharePoint, Exchange, and SQL Server.



Advantages

The advantages of backing up machines and apps to MABS/DPM storage, and then backing up DPM/MABS storage to a vault are as follows:

- Backing up to MABS/DPM provides app-aware backups optimized for common apps such as SQL Server, Exchange, and SharePoint, in additional to file/folder/volume backups, and machine state backups (bare-metal, system state).

- For on-premises machines, you don't need to install the MARS agent on each machine you want to back up. Each machine runs the DPM/MABS protection agent, and the MARS agent runs on the MABS/DPM only.
- You have more flexibility and granular scheduling options for running backups.
- You can manage backups for multiple machines that you gather into protection groups in a single console. This is particularly useful when apps are tiered over multiple machines and you want to back them up together.

Backup steps

1. Install the DPM or MABS protection agent on machines you want to protect. You then add the machines to a DPM protection group.
2. To protect on-premises machines, the DPM or MABS server must be located on-premises.
3. To protect Azure VMs, the MABS server must be located in Azure, running as an Azure VM.
4. With DPM/MABS, you can protect backup volumes, shares, files, and folders. You can also protect a machine's system state (bare metal), and you can protect specific apps with app-aware backup settings.
5. When you set up protection for a machine or app in DPM/MABS, you select to back up to the MABS/DPM local disk for short-term storage and to Azure for online protection. You also specify when the backup to local DPM/MABS storage should run and when the online backup to Azure should run.
6. The disk of the protected workload is backed up to the local MABS/DPM disks, according to the schedule you specified.
7. The DPM/MABS disks are backed up to the vault by the MARS agent that's running on the DPM/MABS server.

Backup Component Comparison

This table summarizes the Azure Backup (MARS) agent and the Azure Backup Server usage cases.

Component	Benefits	Limits	What is protected?	Where are backups stored?
Azure Backup (MARS) agent	Backup files and folders on physical or virtual Windows OS; no separate backup server required	Backup 3x per day; not application aware; file, folder, and volume-level restore only; no support for Linux	Files and folders	Recovery services vault

Component	Benefits	Limits	What is protected?	Where are backups stored?
Azure Backup Server	App aware snapshots; full flex for when to backups; recovery granularity; linux support on Hyper-V and VMware VMs; backup and restore VMware VMs, doesn't require a System Center license	Cannot backup Oracle workloads; always requires live Azure subscription; no support for tape backup	Files, folders, volumes, VMs, applications, and workloads	Recovery services vault, locally attached disk

Virtual Machine Extensions

Virtual Machine Extensions

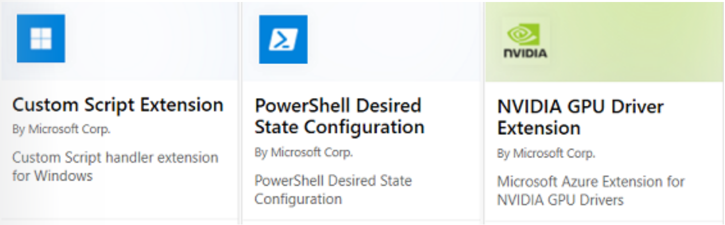
Creating and maintaining virtual machines can be a lot of work, and much of it is repetitive, requiring the same steps each time. Fortunately, there are several ways to automate the tasks of creating, maintaining, and removing virtual machines. One way is to use a virtual machine **extension**.

Azure virtual machine extensions are small applications that provide post-deployment configuration and automation tasks on Azure VMs. For example, if a virtual machine requires software installation, anti-virus protection, or a configuration script inside, a VM extension can be used. Extensions are all about managing your virtual machines.

Azure VM extensions can be:

- Managed with Azure CLI, PowerShell, Azure Resource Manager templates, and the Azure portal.
- Bundled with a new VM deployment or run against any existing system. For example, they can be part of a larger deployment, configuring applications on VM provision, or run against any supported extension operated systems post deployment.

There are different extensions for Windows and Linux machines and a large choice of first and third-party extensions.



Custom Script Extension
By Microsoft Corp.
Custom Script handler extension for Windows

PowerShell Desired State Configuration
By Microsoft Corp.
PowerShell Desired State Configuration

NVIDIA GPU Driver Extension
By Microsoft Corp.
Microsoft Azure Extension for NVIDIA GPU Drivers

✓ In this lesson we will focus on two extensions: Custom Script Extensions and Desired State Configuration. Both tools are based on PowerShell.

For more information, you can see:

Virtual machine extensions and features for Windows - **https://docs.microsoft.com/en-us/azure/virtual-machines/extensions/features-windows?toc=%2Fazure%2Fvirtual-machines%2Fwindows%2Ftoc.json**

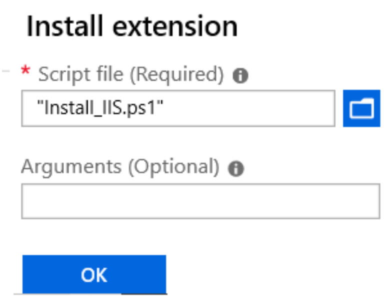
Virtual machine extensions and features for Linux - **https://docs.microsoft.com/en-us/azure/virtual-machines/extensions/features-linux**

Custom Script Extensions

Custom Script Extension (CSE) can be used to automatically launch and execute virtual machine customization tasks post configuration. Your script extension may perform very simple tasks such as stopping the virtual machine or installing a software component. However, the script could be more complex and perform a series of tasks.

Implementation

You can install the CSE from the Azure portal by accessing the virtual machines **Extensions** blade. Once the CSE resource is created, you will provide a PowerShell script file. Your script file will include the PowerShell commands you want to execute on the virtual machine. Optionally, you can pass in arguments, such as param1, param2. Once the file is uploaded it executes immediately.



Install extension

* Script file (Required)

"Install_IIS.ps1"

Arguments (Optional)

You could also use the PowerShell **Set-AzVmCustomScriptExtension** command. You need to upload the script file to a blob container and provide the URI in the command like this:

```
Set-AzVmCustomScriptExtension -FileUri https://scriptstore.blob.core.
windows.net/scripts/Install_IIS.ps1 -Run "PowerShell.exe" -VmName vmName
-ResourceGroupName resourceGroup -Location "location"
```

Things to consider

- **Timeout**. Custom Script extensions have 90 minutes to run. If your deployment exceeds this time, it is marked as a timeout. Keep this in mind when designing your script. And, of course, your virtual machine must be running to perform the tasks.
- **Dependencies**. If your extension requires networking or storage access, make sure that content is available.
- **Failure events**. Be sure to account for any errors that might occur when running your script. For example, running out of disk space, or security and access restrictions. What will the script do if there is an error?
- **Sensitive data**. Your extension may need sensitive information such as credentials, storage account names, and storage account access keys. How will you protect/encrypt this information?

✓ Can you think of any custom script extensions that you might want to create?

Desired State Configuration

Desired State Configuration (DSC) is a management platform in Windows PowerShell that enables deploying and managing configuration data for software services and managing the environment in which these services run. DSC provides a set of Windows PowerShell language extensions, Windows PowerShell cmdlets, and resources that you can use to declaratively specify how you want your software environment to be configured. It also provides a means to maintain and manage existing configurations.

DSC centers around creating *configurations*. A configuration is an easy-to-read script that describes an environment made up of computers (nodes) with specific characteristics. These characteristics can be as simple as ensuring a specific Windows feature is enabled or as complex as deploying SharePoint. Use DSC when the CSE will not work for your application.

In this example we are installing IIS on the localhost. The configuration will saved as a .ps1 file.

```
configuration IISInstall
{
 Node “localhost”
 {
 WindowsFeature IIS
```

```
    {
    Ensure = "Present"
    Name = "Web-Server"
    } } }
```

Notice the DSC script consists of the following:

- The **Configuration** block. This is the outermost script block. You define it by using the **Configuration** keyword and providing a name. In this case, the name of the configuration is *IISInstall*.
- One or more **Node** blocks. These define the nodes (computers or VMs) that you are configuring. In the above configuration, there is one Node block that targets a computer named "localhost".
- One or more resource blocks. This is where the configuration sets the properties for the resources that it is configuring. In this case, there is one resource block that uses the **WindowsFeature resource**[1]. WindowsFeature indicates the name (Web-Server) of the role or feature that you want to ensure is added or removed. Ensure indicates if the role or feature is added. Your choices are Present and Absent.

✓ The Windows PowerShell DSC comes with a set of built-in configuration resources. For example, File Resource, Log Resource, and User Resource. Use the reference link to view the resources that are available to you. Are there any resources that you might be interested in?

For more information, you can see:

Built-In Windows PowerShell Desired State Configuration Resources - **https://docs.microsoft.com/en-us/powershell/dsc/resources/resources#built-in-resources**

Demonstration - Custom Script Extension

In this demonstration, we will explore Custom Script Extensions.

Run a PowerShell script on a virtual machine

Note: This scenario requires a Windows virtual machine in the running state.

1. Connect (RDP) to your Windows virtual machine and open a PowerShell prompt.
2. Run this command and verify the Web Server feature status is **Available** but not Installed.

```
Get-WindowsFeature -name Web-Server
```

3. Create a file **Install_IIS.ps1** on your local machine.
4. Edit the file and add this command:

```
Install-WindowsFeature -Name Web-Server
```

5. In the Azure Portal, access your virtual machine, and select **Extensions**.
6. Click **+ Add**. Take a minute to review the many different extensions that are available.
7. Locate the **Custom Script Extension** resource, select, and click **Create**.
8. Browse to your PowerShell script and upload the file. You should see a notification that the file was uploaded.
9. Click **OK**.

[1] https://docs.microsoft.com/en-us/powershell/dsc/windowsfeatureresource

10. Select your **CustomScriptExtension**.
11. Click **View detailed status** and verify provisioning succceeded.
12. Return to your virtual machine RDP session.
13. Verify the Web Server role was installed. This may take a couple of minutes.

```
Get-WindowsFeature  -name Web-Server
```

Note: You could also use the PowerShell Set-AzVmCustomScriptExtension command to deploy the extension. You would need to upload the script to blob container and use the URI. We will do this in the next demonstration.

```
Set-AzVmCustomScriptExtension -FileUri https://scriptstore.blob.core.
windows.net/scripts/DeployWebServer.ps1 -Run "PowerShell.exe" -VmName
vmName -ResourceGroupName resourceGroup -Location "location"
```

Monitoring Virtual Machines

Monitoring Virtual Machines

Azure Monitor for VMs monitors your Azure virtual machines (VM) and virtual machine scale sets at scale. The service analyzes the performance and health of your Windows and Linux VMs, monitoring their processes and their dependencies on other resources and external processes.
As a solution, Azure Monitor for VMs includes support for monitoring performance and application dependencies for VMs that are hosted on-premises or in another cloud provider. Three key features deliver in-depth insight:

- Logical components of Azure VMs that run Windows and Linux: Are measured against pre-configured health criteria, and they alert you when the evaluated condition is met.
- Pre-defined, trending performance charts: Display core performance metrics from the guest VM operating system.
- Dependency map: Displays the interconnected components with the VM from various resource groups and subscriptions.

✓ Note Currently, the Health feature is offered only for Azure virtual machines and virtual machine scale sets. The Performance and Map features support both Azure VMs and virtual machines that are hosted in your environment or other cloud provider.

Integration with Log Analytics delivers powerful aggregation and filtering, and it can analyze data trends over time. Such comprehensive workload monitoring can't be achieved with Azure Monitor, Service Map, or Log Analytics alone.
You can view this data in a single VM from the virtual machine directly, or you can use Azure Monitor to deliver an aggregated view of your VMs. This view is based on each feature's perspective:

- Health: The VMs are related to a resource group.
- Map and Performance: The VMs are configured to report to a specific Log Analytics workspace.

Azure Monitor for VMs can deliver predictable performance and availability of vital applications. It identifies critical operating system events, performance bottlenecks, and network issues. Azure Monitor for VMs can also help you understand whether an issue is related to other dependencies.
Data usage
When you deploy Azure Monitor for VMs, the data that's collected by your VMs is ingested and stored in Azure Monitor. Based on the pricing that's published on the Azure Monitor pricing page, Azure Monitor for VMs is billed for:

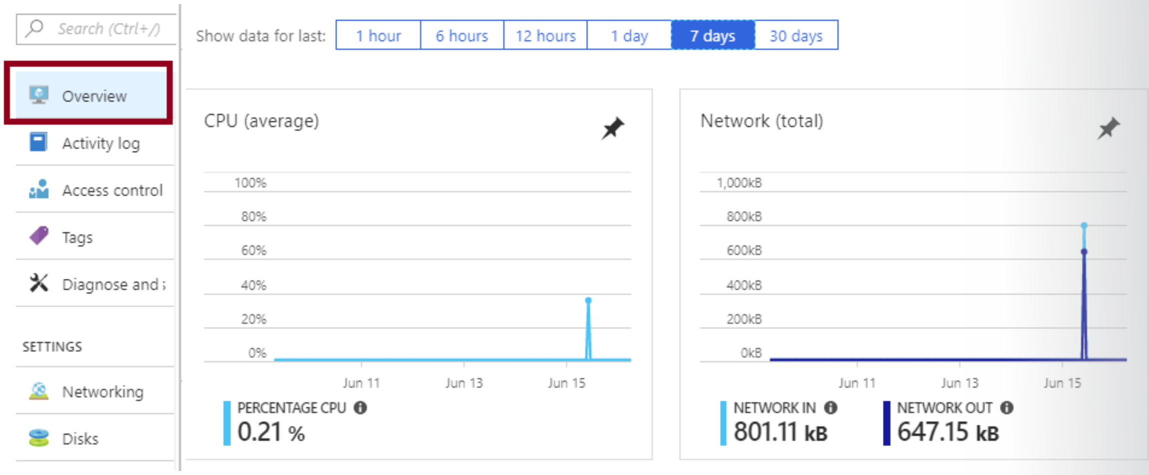
The data that's ingested and stored.
The number of health criteria metric time-series that are monitored.
The alert rules that are created.
The notifications that are sent.

The log size varies by the string lengths of counters, and it can increase with the number of logical disks and network adapters. If you already have a workspace and are collecting these counters, no duplicate charges are applied. If you're already using Service Map, the only change you’ll see is the additional connection data that's sent to Azure Monitor.

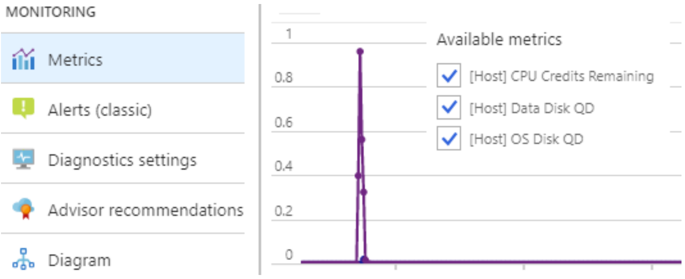
Monitoring

You can take advantage of many opportunities to monitor your VMs by collecting, viewing, and analyzing diagnostic and log data.

To do simple monitoring the Overview screen of the Azure portal shows CPU, Network, Disk bytes, and Disk operations. You can also show the data for different periods of time.



The Monitoring section provides access to Metrics, Diagnostic settings, Advisor recommendations, and Diagram. Azure Monitoring provides the metrics and they are specific to virtual machines.



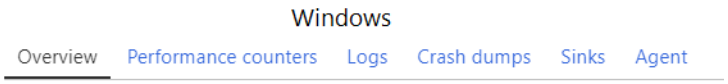
✓ Take a few minutes to navigate the Overview page and the Monitoring section to see what is available for your virtual machine.

For more information, you can see:

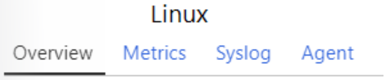
How to monitor virtual machines in Azure - **https://docs.microsoft.com/en-us/azure/virtual-machines/windows/monitor**

Diagnostic settings

The Virtual Machine's Diagnostic setting blade is different for Windows and Linux machines. On Windows machines you have access to Performance counters, Logs, Crash dumps, Sinks, and Agent. Sinks in this context refers to sending your diagnostic data to other services, like Application Insights (additional charges may apply).



On Linux machines your choices are Metrics, Syslog, and Agent. You will need to have version 3.0 or higher of the Linux Diagnostic extension installed in order to edit your diagnostic settings through the portal.



✓ The diagnostics settings can't be updated when the virtual machine isn't running.

For more information, you can see:

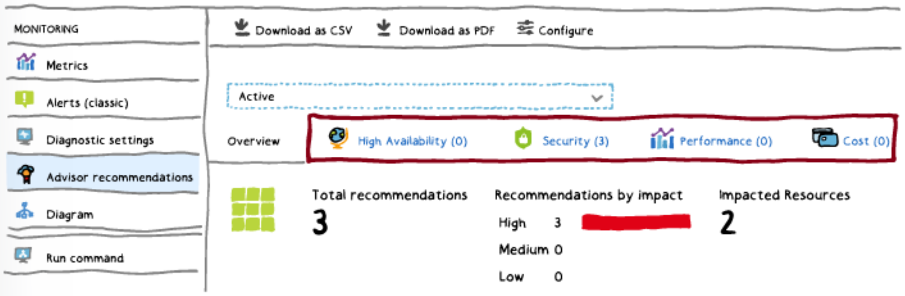
Overview of metrics in Microsoft Azure - **https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-metrics**

Advisor Recommendations

Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, high availability, and security of your Azure resources.

With Advisor, you can:

- Get proactive, actionable, and personalized best practices recommendations.
- Improve the performance, security, and high availability of your resources, as you identify opportunities to reduce your overall Azure spend.



Get recommendations with proposed actions inline.

- **High Availability recommendations**[2] helps you ensure and improve the continuity of your business-critical applications.
- **Security recommendations**[3] to detect threats and vulnerabilities that might lead to security breaches.
- **Performance recommendations**[4] to improve the speed of your applications.
- **Cost recommendations**[5] to optimize and reduce your overall Azure spending.

For more information, you can see:

Introduction to Azure Advisor - **https://docs.microsoft.com/en-us/azure/advisor/advisor-overview**

[2] https://docs.microsoft.com/en-us/azure/advisor/advisor-high-availability-recommendations
[3] https://docs.microsoft.com/en-us/azure/advisor/advisor-security-recommendations
[4] https://docs.microsoft.com/en-us/azure/advisor/advisor-performance-recommendations
[5] https://docs.microsoft.com/en-us/azure/advisor/advisor-cost-recommendations

Online Lab - Implementing custom Azure VM images

Lab Steps

Online Lab: Implementing custom Azure VM images

NOTE: For the most recent version of this online lab, see: **https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies**

Scenario

Adatum Corporation wants to create custom Azure VM images

Objectives

After completing this lab, you will be able to:

- Install and configure HashiCorp Packer
- Create a custom VM image
- Deploy an Azure VM based on a custom image

Lab Setup

Estimated Time: 45 minutes

User Name: **Student**

Password: **Pa55w.rd**

Exercise 1: Installing and configuring HashiCorp Packer

The main tasks for this exercise are as follows:

1. Download HashiCorp Packer
2. Configure HashiCorp Packer prerequisites

Task 1: Download HashiCorp Packer

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. In the Azure portal, in the Microsoft Edge window, start a **Bash** session within the **Cloud Shell**.
3. If you are presented with the **You have no storage mounted** message, configure storage using the following settings:
    - Subsciption: the name of the target Azure subscription
    - Cloud Shell region: the name of the Azure region that is available in your subscription and which is closest to the lab location

- Resource group: **az3000300-LabRG**
- Storage account: a name of a new storage account
- File share: a name of a new file share

4. From the Cloud Shell pane, run the following to download the Packer compressed installation media:

```
wget https://releases.hashicorp.com/packer/1.3.1/packer_1.3.1_linux_amd64.
zip
```

5. From the Cloud Shell pane, run the following to unzip the Packer installation media:

```
unzip packer_1.3.1_linux_amd64.zip
```

Task 2: Configure HashiCorp Packer prerequisites

1. From the Cloud Shell pane, run the following to create a resource group and store the JSON output in a variable (replace the `<Azure region>` placeholder with the name of the Azure region that is available in your subscription and which is closest to the lab location):

```
RG=$(az group create --name az3000301-LabRG --location <Azure region>)
```

2. From the Cloud Shell pane, run the following to create a service principal that will be used by Packer and store the JSON output in a variable:

```
AAD_SP=$(az ad sp create-for-rbac)
```

Result: After you completed this exercise, you have downloaded HashiCorp Packer and configured its prerequisites.

Exercise 2: Creating a custom image

The main tasks for this exercise are as follows:

1. Configure a Packer template
2. Build a Packer-based image

Task 1: Configure a Packer template

1. From the Cloud Shell pane, run the following to retrieve the value of the service principal appId and store it in a variable

```
CLIENT_ID=$(echo $AAD_SP | jq .appId | tr -d '"')
```

2. From the Cloud Shell pane, run the following to retrieve the value of the service principal password and store it in a variable

```
CLIENT_SECRET=$(echo $AAD_SP | jq .password | tr -d '"')
```

3. From the Cloud Shell pane, run the following to retrieve the value of the service principal tenant ID and store it in a variable

```
TENANT_ID=$(echo $AAD_SP | jq .tenant | tr -d '"')
```

4. From the Cloud Shell pane, run the following to retrieve the value of the subscription ID and store it in a variable:

   ```
   SUBSCRIPTION_ID=$(az account show --query id | tr -d '"')
   ```

5. From the Cloud Shell pane, run the following to retrive the value of the resource group location and store it in a variable:

   ```
   LOCATION=$(echo $RG | jq .location | tr -d '"')
   ```

6. From the Cloud Shell pane, upload the Packer template **\allfiles\AZ-300T01\Module_03\template03.json** into the home directory.
7. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **client_id** parameter with the value of the **$CLIENT_ID** variable in the Packer template:

   ```
   sed -i.bak1 's/"$CLIENT_ID"/"'"$CLIENT_ID"'"/' ~/template03.json
   ```

8. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **client_secret** parameter with the value of the **$CLIENT_SECRET** variable in the Packer template:

   ```
   sed -i.bak2 's/"$CLIENT_SECRET"/"'"$CLIENT_SECRET"'"/' ~/template03.json
   ```

9. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **tenant_id** parameter with the value of the **$TENANT_ID** variable in the Packer template:

   ```
   sed -i.bak3 's/"$TENANT_ID"/"'"$TENANT_ID"'"/' ~/template03.json
   ```

10. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **subscription_id** parameter with the value of the **$SUBSCRIPTION_ID** variable in the Packer template:

    ```
    sed -i.bak4 's/"$SUBSCRIPTION_ID"/"'"$SUBSCRIPTION_ID"'"/' ~/template03.
    json
    ```

11. From the Cloud Shell pane, run the following to replace the placeholder for the value of the **location** parameter with the value of the **$LOCATION** variable in the Packer template:

    ```
    sed -i.bak5 's/"$LOCATION"/"'"$LOCATION"'"/' ~/template03.json
    ```

Task 2: Build a Packer-based image

1. From the Cloud Shell pane, run the following to build the packer-based image:

   ```
   ./packer build template03.json
   ```

2. Monitor the built progress until it completes.
3. **Note**: The build process might take about 10 minutes.

Result: After you completed this exercise, you have created a Packer template and used it to build a custom image.

Exercise 3: Deploying a custom image

The main tasks for this exercise are as follows:

1. Deploy an Azure VM based on a custom image
2. Validate Azure VM deployment

Task 1: Deploy an Azure VM based on a custom image

1. From the Cloud Shell pane, run the following to deploy an Azure VM based on the custom image.

   ```
   az vm create --resource-group az3000301-LabRG --name az3000301-vm --image
   az3000301-image --admin-username student --generate-ssh-keys
   ```

2. Wait for the deployment to complete
3. **Note**: The deployment process might take about 3 minutes.
4. Once the deployment completes, from the Cloud Shell pane, run the following to allow inbound traffic to the newly deployed VM on TCP port 80:

   ```
   az vm open-port --resource-group az3000301-LabRG --name az3000301-vm --port
   80
   ```

Task 2: Validate Azure VM deployment

1. From the Cloud Shell pane, run the following to identify the IP address associated with the newly deployed Azure VM.

   ```
   az network public-ip show --resource-group az3000301-LabRG --name
   az3000301-vmPublicIP --query ipAddress
   ```

2. Start Microsoft Edge and navigate to the IP address you identified in the previous step.
3. Verify that Microsoft Edge displays the **Welcome to nginx!** page.

Result: After you completed this exercise, you have deployed an Azure VM based on a custom image and validated the deployment.

Exercise 4: Remove lab resources

Task 1: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

   ```
   az group list --query "[?starts_with(name,'az3000301')]".name --output tsv
   ```

3. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

Task 2: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

```
az group list --query "[?starts_with(name,'az3000301')]".name --output tsv
| xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

Result: In this exercise, you removed the resources used in this lab.

Review Questions

Module 3 Review Questions

Understanding Azure PowerShell

A company hires a new administrator to manage Azure resources. You are reviewing the following virtual machine (VM) deployment script:

```
$resourceGroup = "myResourceGroup"
$location = "westeurope"
$vmName = "myVM"

$cred = Get-Credential -Message "Enter a username and password for the virtual machine."

New-AzureRmResourceGroup -Name $resourceGroup -Location $location

$subnetConfig = New-AzureRmVirtualNetworkSubnetConfig -Name mySubnet -AddressPrefix 
192.168.1.0/24

$vnet = New-AzureRmVirtualNetwork -ResourceGroupName $resourceGroup -Location $location `
  -Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig

$pip = New-AzureRmPublicIpAddress -ResourceGroupName $resourceGroup -Location $location `
  -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -IdleTimeoutInMinutes 4

$nsgRuleRDP = New-AzureRmNetworkSecurityRuleConfig -Name myNetworkSecurityGroupRuleRDP 
-Protocol Tcp `
  -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange * -DestinationAddressPrefix 
* `
  -DestinationPortRange 3389 -Access Allow

$nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $resourceGroup -Location $location 
`
  -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP

$nic = New-AzureRmNetworkInterface -Name myNic -ResourceGroupName $resourceGroup -Location 
$location `
  -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

$vmConfig = New-AzureRmVMConfig -VMName $vmName -VMSize Standard_D1 | `
Set-AzureRmVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | `
Set-AzureRmVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus 
2016-Datacenter -Version latest | `
Add-AzureRmVMNetworkInterface -Id $nic.Id

New-AzureRmVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig
```

You need to modify the script to deploy the following VM:

- VM Name: “FinanceApp_Server1”
- OS Version: Windows Server 2016 standard

Which changes should be made to the Azure PowerShell script? Wich part of the PowerShell script creates the VM?

Suggested Answer ↓

In the existing PowerShell script, the following lines needs to be modified:

$vmName = "myVM"

Set-AzureRmVMSourceImage -PublisherName MicrosoftWindowsServer -Offer WindowsServer -Skus 2016-Datacenter -Version latest | `

The command that creates the VM is:

New-AzureRmVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig

Moving On-Premises VMs to Azure

You create custom virtual machine (VM) images in an on-premises staging environment. Once approved and tested, the VMs must be moved to Azure.

You need to move a VM that supports an application to Azure.

What steps should you perform? Which configuration options should you consider?

Suggested Answer ↓

If you cannot find a virtual machine in the Marketplace that meets your needs, you can upload a VHD from that you create by using an on-premises virtualization tool such as Hyper-V.

Considerations:

- In Azure, you can only use Generation 1 virtual machines that use the VHD file format.
- You must ensure the VM is configured to retrieve an IP address and DNS settings from DHCP.
- You must remove any guest virtualization tools and agents from the VM.

Moving Virtual Machines Between Resource Groups

An organization has several virtual machines (VMs) that were created by using the classic deployment model.

You need to move the classic VMs to Azure Resource Groups to improve administration and delegation.

How can you move the VMs to resource groups? What restrictions are there to moving the VMS?

Suggested Answer ↓

With the older classic deployment model, when resources were created, there was no support for resource group management, leading to administration of larger numbers of resources, and inability to perform simple management of those resources in a consolidated fashion. However, the newer Azure Resource Manager deployment model gives you the ability to move resources, such as virtual machines, between resources, including VMs created in the classic deployment model.

You can use the Azure portal, Azure PowerShell, or Azure CLI to move classic VMs to resource groups.

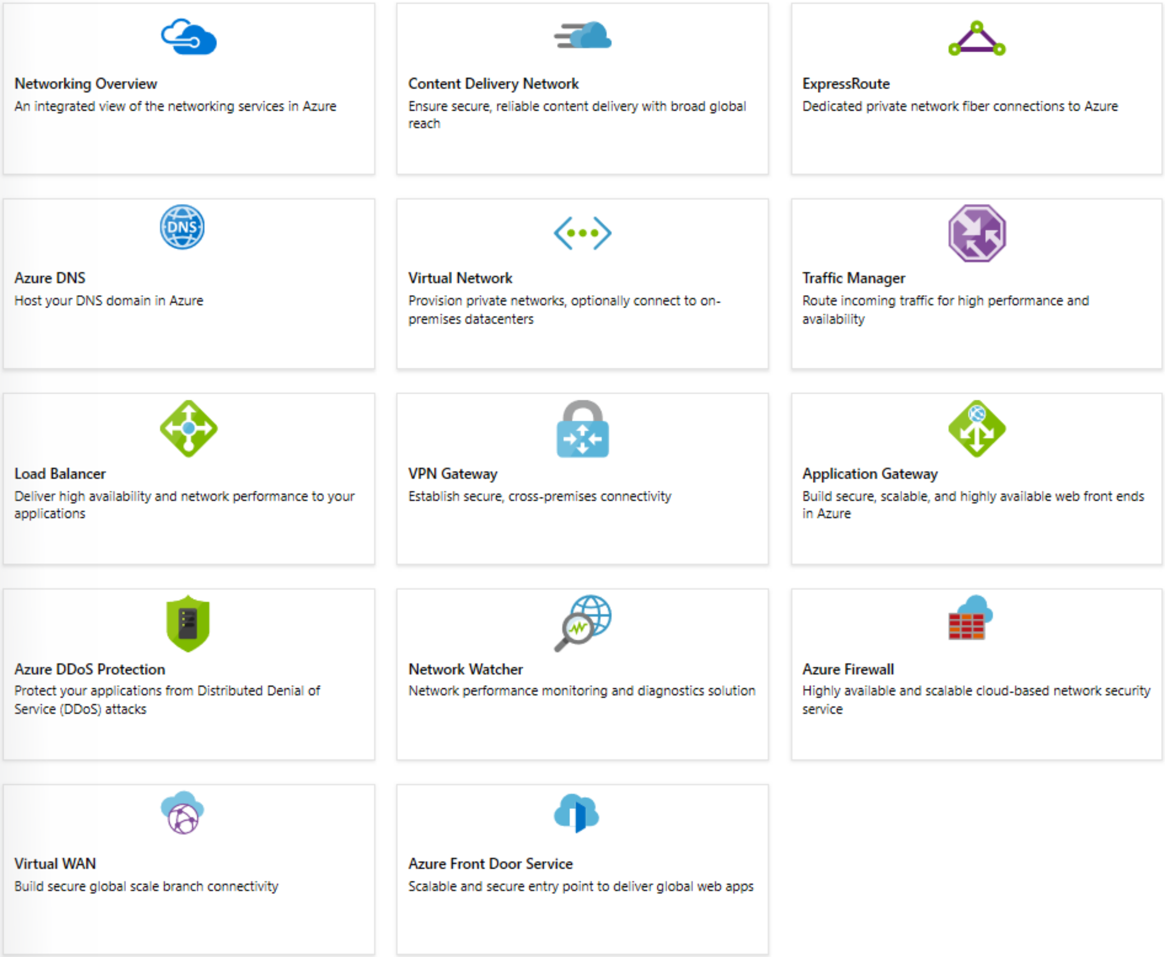
Module 4 Module Configuring and Managing Virtual Networks

Azure Virtual Networks

Azure Networking Components

A major incentive for adopting cloud solutions such as Azure is to enable information technology (IT) departments to move server resources to the cloud. This can save money and simplify operations by removing the need to maintain expensive datacenters with uninterruptible power supplies, generators, multiple fail-safes, clustered database servers, and so on. For small and medium-sized companies, which might not have the expertise to maintain their own robust infrastructure, moving to the cloud is particularly appealing.

Once the resources are moved to Azure, they require the same networking functionality as an on-premises deployment, and in specific scenarios require some level of network isolation. Azure networking components offer a range of functionalities and services that can help organizations design and build cloud infrastructure services that meet their requirements. Azure has many networking components.



Networking Overview
An integrated view of the networking services in Azure

Content Delivery Network
Ensure secure, reliable content delivery with broad global reach

ExpressRoute
Dedicated private network fiber connections to Azure

Azure DNS
Host your DNS domain in Azure

Virtual Network
Provision private networks, optionally connect to on-premises datacenters

Traffic Manager
Route incoming traffic for high performance and availability

Load Balancer
Deliver high availability and network performance to your applications

VPN Gateway
Establish secure, cross-premises connectivity

Application Gateway
Build secure, scalable, and highly available web front ends in Azure

Azure DDoS Protection
Protect your applications from Distributed Denial of Service (DDoS) attacks

Network Watcher
Network performance monitoring and diagnostics solution

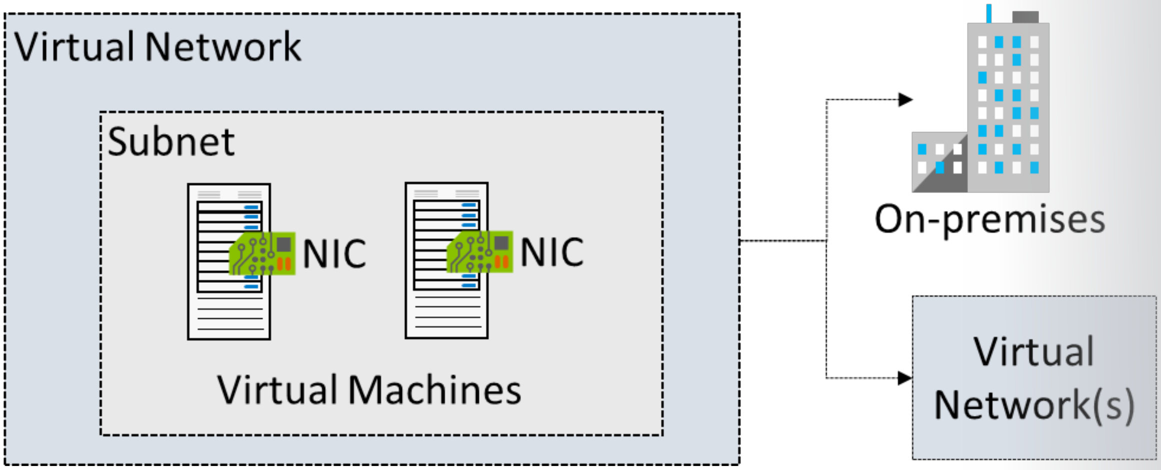
Azure Firewall
Highly available and scalable cloud-based network security service

Virtual WAN
Build secure global scale branch connectivity

Azure Front Door Service
Scalable and secure entry point to deliver global web apps

Virtual Networks

An Azure Virtual Network (VNet) is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can use VNets to provision and manage virtual private networks (VPNs) in Azure and, optionally, link the VNets with other VNets in Azure, or with your on-premises IT infrastructure to create hybrid or cross-premises solutions. Each VNet you create has its own CIDR block and can be linked to other VNets and on-premises networks if the CIDR blocks do not overlap. You also have control of DNS server settings for VNets, and segmentation of the VNet into subnets.



You can use virtual networks to:

- **Create a dedicated private cloud-only VNet**. Sometimes you don't require a cross-premises configuration for your solution. When you create a VNet, your services and VMs within your VNet can communicate directly and securely with each other in the cloud. You can still configure endpoint connections for the VMs and services that require internet communication, as part of your solution.
- **Securely extend your data center With VNets**. You can build traditional site-to-site (S2S) VPNs to securely scale your datacenter capacity. S2S VPNs use IPSEC to provide a secure connection between your corporate VPN gateway and Azure.
- **Enable hybrid cloud scenarios**. VNets give you the flexibility to support a range of hybrid cloud scenarios. You can securely connect cloud-based applications to any type of on-premises system such as mainframes and Unix systems.

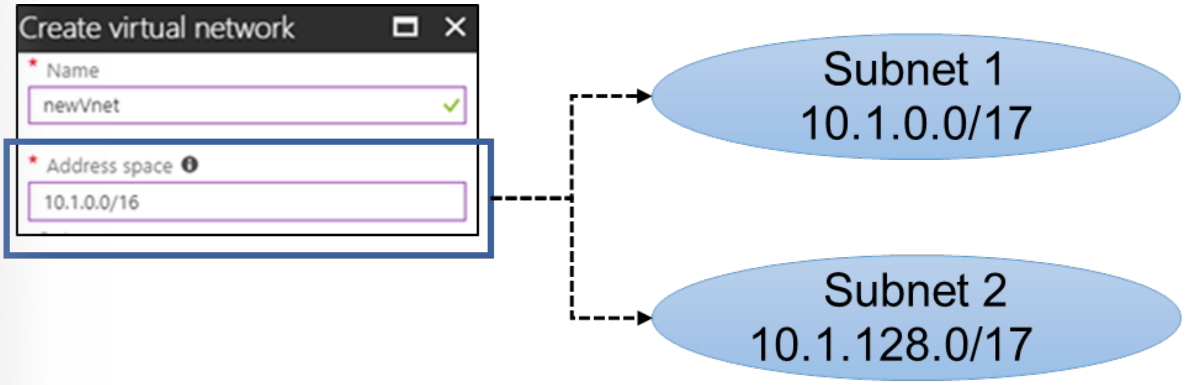
For more information, you can see:

Virtual Network Documentation - **https://docs.microsoft.com/en-us/azure/virtual-network/**

Subnets

A virtual network can be segmented into one or more subnets. Subnets provide logical divisions within your network. Subnets can help improve security, increase performance, and make it easier to manage the network.

Each subnet contains a range of IP addresses that fall within the virtual network address space. Each subnet must have a unique address range, specified in CIDR format. The address range cannot overlap with other subnets in the virtual network in the same subscription.



Create virtual network

* Name

newVnet

* Address space

10.1.0.0/16

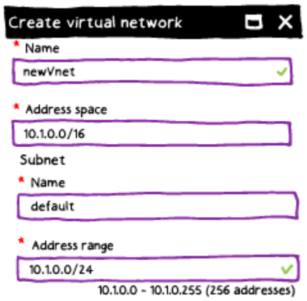
It is important to carefully plan your subnets. Here are some things to think about.

- **Service requirements**. Each service directly deployed into virtual network has specific requirements for routing and the types of traffic that must be allowed into and out of subnets. A service may require, or create, their own subnet, so there must be enough unallocated space for them to do so. For example, if you connect a virtual network to an on-premises network using an Azure VPN Gateway, the virtual network must have a dedicated subnet for the gateway.
- **Virtual appliances**. Azure routes network traffic between all subnets in a virtual network, by default. You can override Azure's default routing to prevent Azure routing between subnets, or to route traffic between subnets through a network virtual appliance. So, if you require that traffic between resources in the same virtual network flow through a network virtual appliance (NVA), deploy the resources to different subnets.
- **Service endpoints**. You can limit access to Azure resources such as an Azure storage account or Azure SQL database, to specific subnets with a virtual network service endpoint. Further, you can deny access to the resources from the internet. You may create multiple subnets, and enable a service endpoint for some subnets, but not others.
- **Network security groups**. You can associate zero or one network security group to each subnet in a virtual network. You can associate the same, or a different, network security group to each subnet. Each network security group contains rules, which allow or deny traffic to and from sources and destinations.

✓ Azure reserves the first three IP addresses and the last IP address in each subnet address range.

Implementing Virtual Networks

You can create new virtual networks at any time. You can also add virtual networks when you create a virtual machine. Either way you will need to define the address space, and at least one subnet. By default, you can create up to 50 virtual networks per subscription per region, although you can increase this limit to 500 by contacting Azure support.



Create virtual network

* Name

newVnet

* Address space

10.1.0.0/16

Subnet

* Name

default

* Address range

10.1.0.0/24

10.1.0.0 - 10.1.0.255 (256 addresses)

✓ Always plan to use an address space that is not already in use in your organization, either on-premises or in other VNets. Even if you plan for a VNet to be cloud-only, you may want to make a VPN connection to it later. If there is any overlap in address spaces at that point, you will have to reconfigure or recreate the VNet. The next lesson will focus on IP addressing.

For more information, you can see:

What is Azure Virtual Network - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview**

Networking Limits - **https://docs.microsoft.com/en-us/azure/azure-subscription-service-limits#networking-limits-1**

Demonstration - Creating Virtual Networks

In this demonstration, you will create virtual networks.

Note: You can use the suggested values for the settings, or your own custom values if you prefer.

Create a virtual network in the portal

1. Sign in the to the Azure portal and search for **Virtual Networks**.
2. On the Virtual Networks page, click **Add**.
   - **Name**: *myVNet1*.
   - **Address**:*10.1.0.0/16*.
   - **Subscription**: Select your subscription.
   - **Resource group**: Select new or choose an existing resource group
   - **Location** - Select your location
   - **Subnet** - Enter *mySubnet1*.
   - **Subnet - Address range**: *10.1.0.0/24*
3. Leave the rest of the default settings and select **Create**.
4. Verify your virtual network was created.

Create a virtual network using PowerShell

1. Create a virtual network. Use values as appropriate.

```
$myVNet2 = New-AzVirtualNetwork -ResourceGroupName myResourceGroup -Location EastUS -Name myVNet2 -AddressPrefix 10.0.0.0/16
```

2. Verify your new virtual network information.

```
Get-AzVirtualNetwork -Name myVNet2
```

3. Create a subnet. Use values as appropriate.

```
$mySubnet2 = Add-AzVirtualNetworkSubnetConfig -Name mySubnet2 -AddressPrefix
10.0.0.0/24 -VirtualNetwork $myVNet2
```

4. Verify your new subnet information.

```
Get-AzVirtualNetworkSubnetConfig -Name mySubnet2 -VirtualNetwork $myVNet2
```

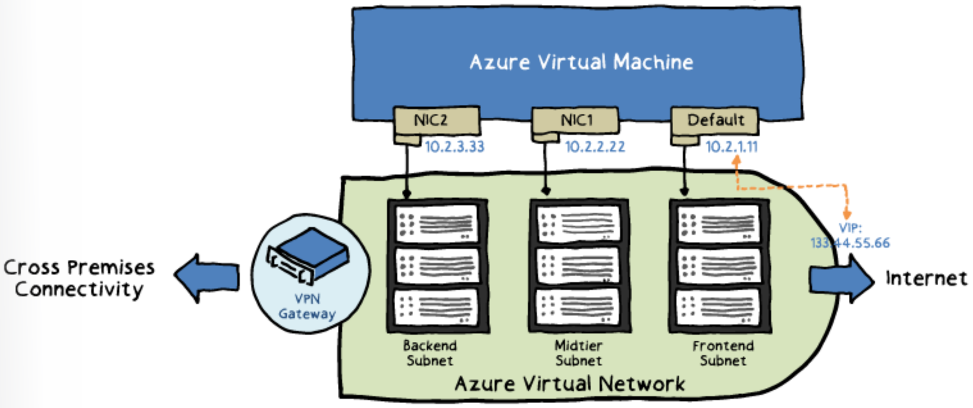
5. Associate the subnet to the virtual network.

```
$mySubnet2 | Set-AzVirtualNetwork
```

6. Return to the portal and verify your new virtual network with subnet was created.

Multiple NICs in Virtual Machines

You can create virtual machines in Azure and attach multiple network interfaces (NICs) to each of your VMs. Having multiple NICs is a requirement for many network virtual appliances, such as application delivery and WAN optimization solutions. Having multiple NICs also provides more network traffic management functionality, including isolation of traffic between a front-end NIC and back-end NIC(s), or separation of data plane traffic from management plane traffic.



The figure above shows a VM with three NICs, each connected to a different subnet.

- The order of the NICs from inside the VM will be random and could also change across Azure infrastructure updates. However, the IP addresses, and the corresponding ethernet MAC addresses will remain the same. For example, assume Eth1 has IP address 10.1.0.100 and MAC address 00-0D-3A-B0-39-0D; after an Azure infrastructure update and reboot, it could be changed to Eth2, but the IP and MAC pairing will remain the same. When a restart is customer-initiated, the NIC order will remain the same.
- The address for each NIC on each VM must be in a subnet and multiple NICs on a single VM can each be assigned addresses that are in the same subnet.

- The VM size determines the number of NICS that you can create for a VM.

The following limitations are applicable when using the multiple NIC feature:

- All VMs in an availability set need to use either multiple NICs or a single NIC. You cannot have a mixture of multi NIC VMs and single NIC VMs within an availability set. Same rules apply for VMs in a cloud service.
- A VM with single NIC cannot be configured with multiple NICs (and vice-versa) once it is deployed, without deleting and re-creating it.

For more information, you can see:

Add network interfaces to or remove network interfaces from virtual machines - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-network-interface-vm**

Demonstration - Create VMs with Multiple NICs

In this demonstration, you will learn how to create and configure multiple NICs and then attach those NICs to a VM. You can replace example parameter names with your own values if you prefer.

This demonstration uses the Azure CLI and assumes the following preparatory steps:

1. You are using the latest version of the **Azure CLI**[1] and are logged in to your Azure account.
2. You have created a resource group in an appropriate location and a virtual network with a subnet, an additional backend subnet, and a network security group.For example, using **az network vnet create**, create a virtual network named *myVnet* and subnet named *mySubnetFrontEnd*:

```
az network vnet create \
    --resource-group myResourceGroup \
    --name myVnet \
    --address-prefix 10.0.0.0/16 \
    --subnet-name mySubnetFrontEnd \
    --subnet-prefix 10.0.1.0/24
```

1. Using **az network vnet subnet** create a subnet for the back-end traffic named *mySubnetBackEnd*:

```
az network vnet subnet create \
    --resource-group myResourceGroup \
    --vnet-name myVnet \
    --name mySubnetBackEnd \
    --address-prefix 10.0.2.0/24
```

1. Now using **az network nsg create**, create a network security group named *myNetworkSecurityGroup*:

```
az network nsg create \
    --resource-group myResourceGroup \
    --name myNetworkSecurityGroup
```

Create and configure multiple NICs

- Using **az network nic create**, create two NICs, named *myNic1* and *myNic2*, connect the network security group,

[1] https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest

with one NIC connecting to each subnet:

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic1 \
    --vnet-name myVnet \
    --subnet mySubnetFrontEnd \
    --network-security-group myNetworkSecurityGroup
```

Create a VM and attach the NICs

- When you create the VM, specify the NICs you created with the *–nics* parameter. You also need to take care when you select the VM size. There are limits for the total number of NICs that you can add to a VM. Using **az vm create**, create a Linux VM named *myVM*:

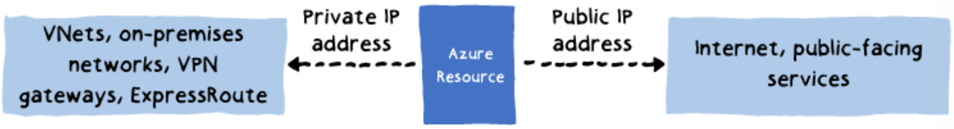
```
az vm create \
     --resource-group myResourceGroup \
     --name myVM \
     --image UbuntuLTS \
     --size Standard_DS3_v2 \
     --admin-username azureuser \
     --generate-ssh-keys \
     --nics myNic1 myNic2
```

Note: Return to the portal and verify the virtual machine now has two interfaces.

Review of IP Addressing

Overview of IP Addressing

You can assign IP addresses to Azure resources to communicate with other Azure resources, your on-premises network, and the Internet. There are two types of IP addresses you can use in Azure. Virtual networks can contain both public and private IP address spaces.


VNets, on-premises networks, VPN gateways, ExpressRoute
Private IP address
Azure Resource
Public IP address
Internet, public-facing services

1. **Private IP addresses**: Used for communication within an Azure virtual network (VNet), and your on-premises network, when you use a VPN gateway or ExpressRoute circuit to extend your network to Azure.
2. **Public IP addresses**: Used for communication with the Internet, including Azure public-facing services.

IP addresses can also be statically assigned or dynamically assigned. Static IP addresses do not change and are best for certain situations such as:

- DNS name resolution, where a change in the IP address would require updating host records.
- IP address-based security models which require apps or services to have a static IP address.
- SSL certificates linked to an IP address.
- Firewall rules that allow or deny traffic using IP address ranges.
- Role-based VMs such as Domain Controllers and DNS servers.

✓ As a best practice you may decide to separate dynamically and statically assigned IP resources into different subnets. And, IP Addresses are never managed from within a virtual machine.

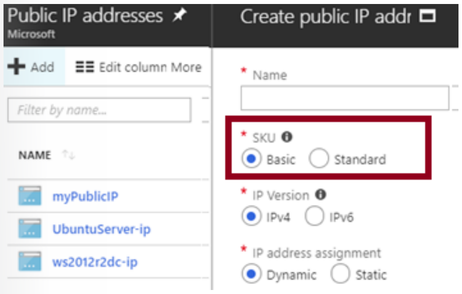
Public IP Addresses

A public IP address resource can be associated with virtual machine network interfaces, internet-facing load balancers, VPN gateways, and application gateways. Azure can provide an IP address (dynamic assignment) or you can assign the IP address (static assignment). The type of resource affects the assignment.

Public IP addresses	IP address association	Dynamic	Static
Virtual Machine	NIC	Yes	Yes
Load Balancer	Front-end configuration	Yes	Yes
VPN Gateway	Gateway IP configuration	Yes	No
Application Gateway	Front-end configuration	Yes	No

Address SKUs

When you create a public IP address you are given a SKU choice of either Basic or Standard.



* Name

* SKU
● Basic ○ Standard

* IP Version
● IPv4 ○ IPv6

* IP address assignment
● Dynamic ○ Static

Your SKU choice affects the IP assignment method, security, available resources, and redundancy. This table summarizes the differences.

Feature	Basic SKU	Standard SKU
IP assignment	Static or dynamic	Static
Security	Open by default	Are secure by default and closed to inbound traffic
Resources	Network interfaces, VPN Gateways, Application Gateways, and Internet-facing load balancers	Network interfaces or public standard load balancers
Redundancy	Not zone redundant	Zone redundant by default

Private IP Addresses

A private IP address resource can be associated with virtual machine network interfaces, internal load balancers, and application gateways. Azure can provide an IP address (dynamic assignment) or you can assign the IP address (static assignment).

Private IP Addresses	IP address association	Dynamic	Static
Virtual Machine	NIC	Yes	Yes
Internal Load Balancer	Front-end configuration	Yes	Yes
Application Gateway	Front-end configuration	Yes	Yes

A private IP address is allocated from the address range of the virtual network subnet a resource is deployed in.

- **Dynamic**. Azure assigns the next available unassigned or unreserved IP address in the subnet's address range. For example, Azure assigns 10.0.0.10 to a new resource, if addresses 10.0.0.4-10.0.0.9 are already assigned to other resources. Dynamic is the default allocation method.
- **Static**. You select and assign any unassigned or unreserved IP address in the subnet's address range. For example, if a subnet's address range is 10.0.0.0/16 and addresses 10.0.0.4-10.0.0.9 are already assigned to other resources, you can assign any address between 10.0.0.10 - 10.0.255.254.

Demonstration - Manage IP Addresses

In this demonstration, you will learn how to retrieve static private IP address information for a network interface. You will run PowerShell commands to view the static private IP address information for the VM that was created in the previous demonstration. You will also remove the static private IP address that was added to the VM.

Retrieve static private IP address information

1. To view the static private IP address information for the VM created with the script above, run the following PowerShell command and note the values for *PrivateIpAddress* and *PrivateIpAllocationMethod*:

```
Get-AzNetworkInterface -Name TestNIC -ResourceGroupName TestRG
```

1. Review the information returned which includes: Name, ResourceGroupName, Location, Id, ProvisioningState, VirtualMachine, IpConfigurations, DnsSettings, EnableIPForwarding, and NetworkSecurityGroup. The information also includes whether the NIC is primary.
2. Notice in the IpConfigurations area there is a PrivateIPAddress and the PrivateIpAllocationMethod is static.

Remove a static private IP address

1. To remove the static private IP address added to the VM in the previous demonstration, run the following PowerShell commands:

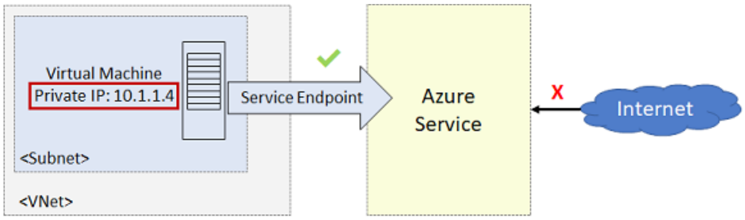
```
$nic=Get-AzNetworkInterface -Name TestNIC -ResourceGroupName TestRG
$nic.IpConfigurations[0].PrivateIpAllocationMethod = "Dynamic"
Set-AzNetworkInterface -NetworkInterface $nic
```

1. Review the output.
2. Notice in the IpConfigurations area, The PrivateIPAllocationMethod is now Dynamic.

Service Endpoints

A virtual network service endpoint provides the identity of your virtual network to the Azure service. Once service endpoints are enabled in your virtual network, you can secure Azure service resources to your virtual network by adding a virtual network rule to the resources.

Today, Azure service traffic from a virtual network uses public IP addresses as source IP addresses. With service endpoints, service traffic switches to use virtual network private addresses as the source IP addresses when accessing the Azure service from a virtual network. This switch allows you to access the services without the need for reserved, public IP addresses used in IP firewalls.



Why use a service endpoint?

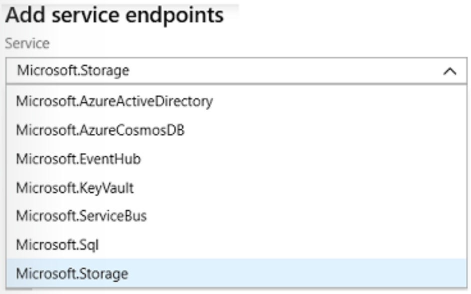
- **Improved security for your Azure service resources**. VNet private address space can be overlapping and so, cannot be used to uniquely identify traffic originating from your VNet. Service endpoints provide the ability to secure Azure service resources to your virtual network, by extending VNet identity to the service. Once service endpoints are enabled in your virtual network, you can secure Azure service resources to your virtual network by adding a virtual network rule to the resources. This provides improved security by fully removing public Internet access to resources, and allowing traffic only from your virtual network.

- **Optimal routing for Azure service traffic from your virtual network**. Today, any routes in your virtual network that force Internet traffic to your premises and/or virtual appliances, known as forced-tunneling, also force Azure service traffic to take the same route as the Internet traffic. Service endpoints provide optimal routing for Azure traffic.
- **Endpoints always take service traffic directly from your virtual network to the service on the Microsoft Azure backbone network**. Keeping traffic on the Azure backbone network allows you to continue auditing and monitoring outbound Internet traffic from your virtual networks, through forced-tunneling, without impacting service traffic. Learn more about user-defined routes and forced-tunneling.
- **Simple to set up with less management overhead**. You no longer need reserved, public IP addresses in your virtual networks to secure Azure resources through IP firewall. There are no NAT or gateway devices required to set up the service endpoints. Service endpoints are configured through a simple click on a subnet. There is no additional overhead to maintaining the endpoints.

✓ With service endpoints, the source IP addresses of the virtual machines in the subnet for service traffic switches from using public IPv4 addresses to using private IPv4 addresses. Existing Azure service firewall rules using Azure public IP addresses will stop working with this switch. Please ensure Azure service firewall rules allow for this switch before setting up service endpoints. You may also experience temporary interruption to service traffic from this subnet while configuring service endpoints.

Service Endpoint Services

It is easy to add a service endpoint to the virtual network. Several services are available including: Azure Active Directory, Azure Cosmos DB, EventHub, KeyVault, Service Bus, SQL, and Storage.



Add service endpoints

Service

Microsoft.Storage

Azure Storage. Generally available in all Azure regions. This endpoint gives traffic an optimal route to the Azure Storage service. Each storage account supports up to 100 virtual network rules.

Azure SQL Database and Azure SQL Data Warehouse. Generally available in all Azure regions. A firewall security feature that controls whether the database server for your single databases and elastic pool in Azure SQL Database or for your databases in SQL Data Warehouse accepts communications that are sent from particular subnets in virtual networks.

Azure Database for PostgreSQL server and MySQL. Generally available in Azure regions where database service is available. Virtual Network (VNet) services endpoints and rules extend the private address space of a Virtual Network to your Azure Database for PostgreSQL server and MySQL server.

Azure Cosmos DB. Generally available in all Azure regions. You can configure the Azure Cosmos account to allow access only from a specific subnet of virtual network (VNet). By enabling Service endpoint to access Azure Cosmos DB on the subnet within a virtual network, the traffic from that subnet is sent to Azure Cosmos DB with the identity of the subnet and Virtual Network. Once the Azure Cosmos DB service endpoint is enabled, you can limit access to the subnet by adding it to your Azure Cosmos account.

Azure Key Vault. Generally available in all Azure regions. The virtual network service endpoints for Azure Key Vault allow you to restrict access to a specified virtual network. The endpoints also allow you to restrict access to a list of IPv4 (internet protocol version 4) address ranges. Any user connecting to your key vault from outside those sources is denied access.

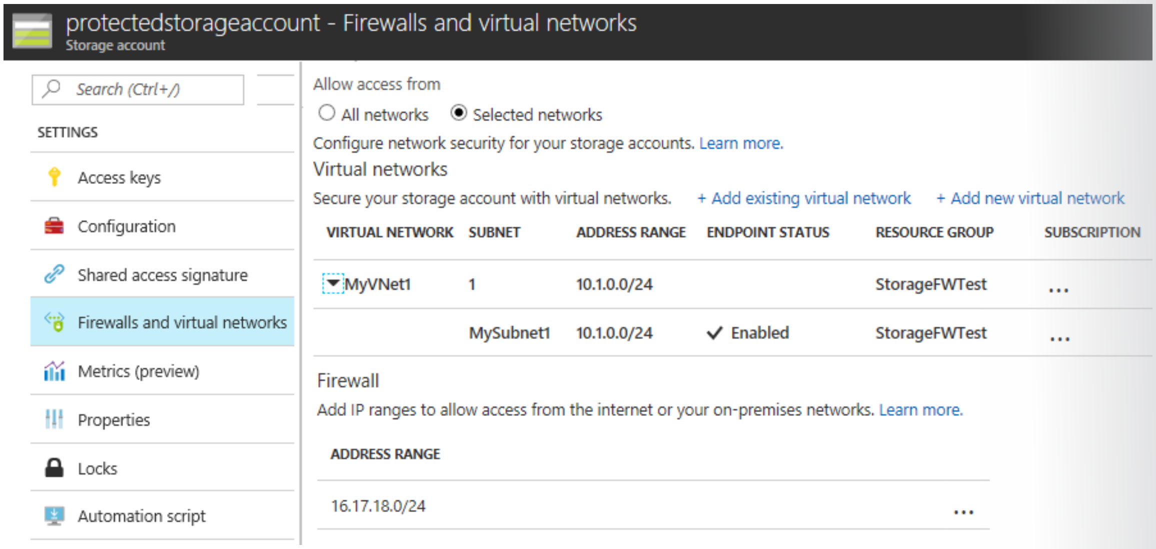
Azure Service Bus and Azure Event Hubs. Generally available in all Azure regions. The integration of Service Bus with Virtual Network (VNet) service endpoints enables secure access to messaging capabilities from workloads like virtual machines that are bound to virtual networks, with the network traffic path being secured on both ends.

Azure Data Lake Store Gen 1. Generally available in all Azure regions where ADLS Gen1 is available. This feature helps to secure your Data Lake Storage account from external threats.

✓ Adding service endpoints can take up to 15 minutes to complete. Each service endpoint integration has its own Azure documentation page.

Secure Access to Storage Endpoints

The steps necessary to restrict network access to Azure services varies across services. For accessing a storage account, you would use the **Firewalls and virtual networks** blade to add the virtual networks that will have access. Notice you can also configure to allow access to one or more public IP ranges.



protectedstorageaccount - Firewalls and virtual networks
Storage account

Allow access from

○ All networks ● Selected networks

Configure network security for your storage accounts. Learn more.

Virtual networks

Secure your storage account with virtual networks. + Add existing virtual network + Add new virtual network

VIRTUAL NETWORK	SUBNET	ADDRESS RANGE	ENDPOINT STATUS	RESOURCE GROUP	SUBSCRIPTION
MyVNet1	1	10.1.0.0/24		StorageFWTest	...
	MySubnet1	10.1.0.0/24	✓ Enabled	StorageFWTest	...

Firewall

Add IP ranges to allow access from the internet or your on-premises networks. Learn more.

ADDRESS RANGE	
16.17.18.0/24	...

✓ It is important to test and ensure the service endpoint is limiting access as expected.

Demonstration - Service Endpoints

In this demonstration, you will work with virtual network endpoints.

Note: This demonstration requires a Storage Account with an uploaded file.
Note: You could use Storage Explorer (Preview) in the portal.

Create a storage account

1. Create a **Storage Account**.
2. Within the Storage Account, create a **file share**, and **upload** a file.
3. For the Storage Account, use the **Shared Access Signature** blade to **Generate SAS and connection string**.

4. Use Storage Explorer and the connection string to access the file share.
5. Ensure you can view your uploaded file.

Note: This part of the demonstration requires a virtual network with a subnet.

Create a subnet service endpoint

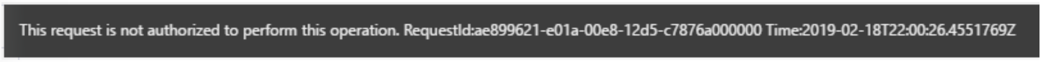
1. Select your virtual network, and then select a subnet in the virtual network.
2. Under **Service Endpoints**, view the **Services** drop-down and the different services that can be secured with an endpoint.
3. Check the **Microsoft.Storage** option.
4. **Save** your changes.

Secure the storage to the service endpoint

1. Return to your **storage account**.
2. Select **Firewalls and virtual networks**.
3. Change to **Selected networks**.
4. Add existing virtual network, verify your subnet with the new service endpoint is listed.
5. **Save** your changes.

Test the storage endpoint

1. Return to the Storage Explorer.
2. **Refresh** the storage account.
3. You should now have an access error similar to this one:



```
This request is not authorized to perform this operation. RequestId:ae899621-e01a-00e8-12d5-c7876a000000 Time:2019-02-18T22:00:26.4551769Z
```

Note: If you plan to use the storage account in other scenarios be sure to return the account to **All networks** in the **Firewalls and virtual networks** blade.

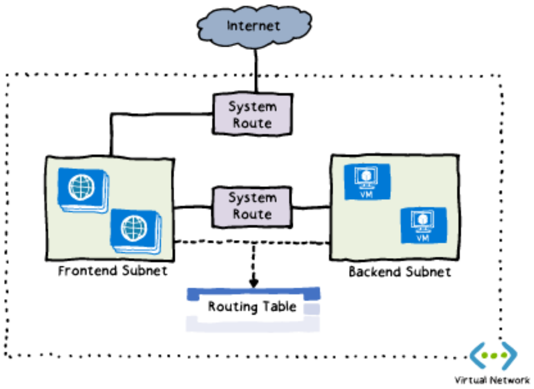
Network Routing

System Routes

Azure uses **system routes** to direct network traffic between virtual machines, on-premises networks, and the Internet. The following situations are managed by these system routes:

- Traffic between VMs in the same subnet.
- Between VMs in different subnets in the same virtual network.
- Data flow from VMs to the Internet.
- Communication between VMs using a VNet-to-VNet VPN.
- Site-to-Site and ExpressRoute communication through the VPN gateway.

For example, consider this virtual network with two subnets. Communication between the subnets and from the frontend to the internet are all managed by Azure using the default system routes.



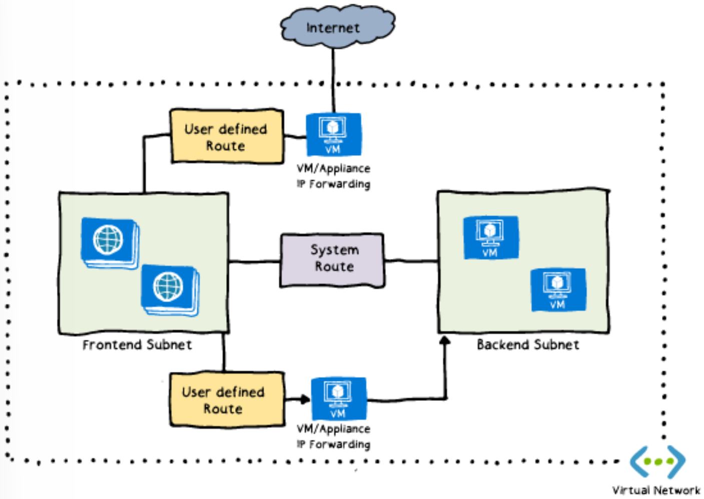
Information about the system routes is recorded in a route table. A route table contains a set of rules, called routes, that specifies how packets should be routed in a virtual network. Route tables are associated to subnets, and each packet leaving a subnet is handled based on the associated route table. Packets are matched to routes using the destination. The destination can be an IP address, a virtual network gateway, a virtual appliance, or the internet. If a matching route can't be found, then the packet is dropped.

For more information, you can see:

System routes - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview#system-routes**

User Defined Routes

As you have just read, Azure automatically handles all network traffic routing. But, what if you want to do something different? For example, you may have a VM that performs a network function, such as routing, firewalling, or WAN optimization. You may want certain subnet traffic to be directed to this virtual appliance. For example, you might place an appliance between subnets or a subnet and the internet.



In these situations, you can configure user-defined routes (UDRs). UDRs control network traffic by defining routes that specify the next hop of the traffic flow. This hop can be a virtual network gateway, virtual network, internet, or virtual appliance.

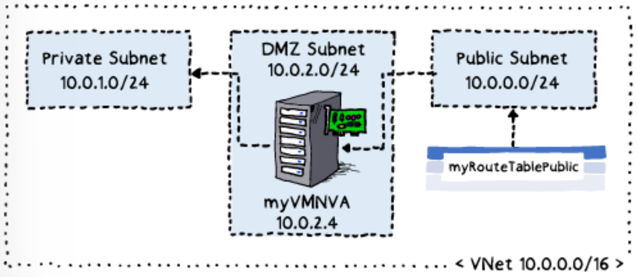
✓ Each route table can be associated to multiple subnets, but a subnet can only be associated to a single route table. There are no additional charges for creating route tables in Microsoft Azure. Do you think you will need to create custom routes?

For more information, you can see:

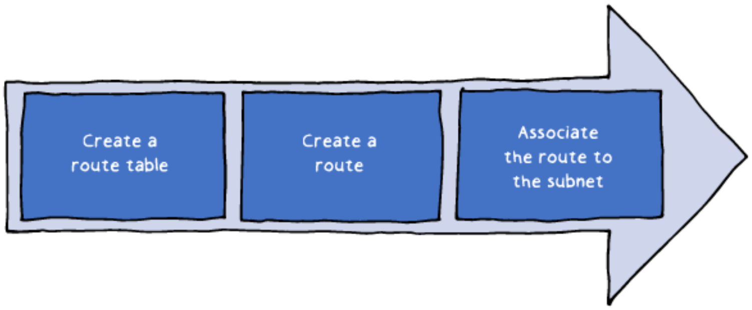
Custom routes - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview#custom-routes**

Routing Example

Let's look at a specific example where you have a virtual network that includes 3 subnets: Private, DMZ, and Public. In the DMZ subnet there is a network virtual appliance (NVA). You want to ensure all traffic from the Public subnet goes through the NVA to the Private subnet.



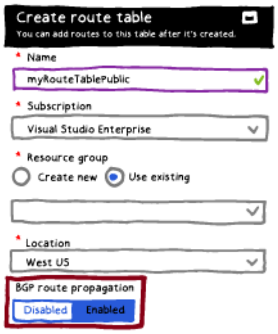
Let's look at how we could implement this scenario by creating the route table, creating the route, and associating the route to the subnet.



✓ There is practice exercise that includes a complete set of steps for this scenario, including creating the virtual appliance and testing.

Create Route Table

Creating a route table is very straightforward, but pay attention to the Border Gateway Protocol (BGP) route propagation setting. In this case, we will want to enable BGP route propagation.



Create route table
You can add routes to this table after it's created.

* Name
myRouteTablePublic ✓

* Subscription
Visual Studio Enterprise

* Resource group
○ Create new ● Use existing

* Location
West US

BGP route propagation
Disabled [✓ Enabled]

BGP is the standard routing protocol commonly used on the Internet to exchange routing and reachability information between two or more networks. Routes are automatically added to the route table of all subnets with BGP propagation enabled. In many situations this is what you want. For example, if you are using ExpressRoute you would want all subnets to know about that routing. Read more at the reference links.

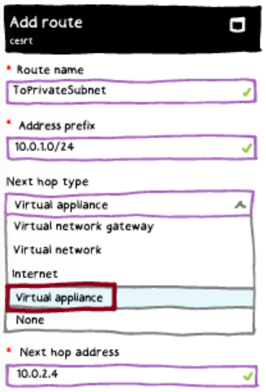
For more information, you can see:

Border gateway protocol - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview#border-gateway-protocol**

Overview of BGP with Azure VPN Gateways - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-bgp-overview?toc=%2fazure%2fvirtual-network%2ftoc.json**

Create and Associate the Route

When you create a route there are several Next hop types. In this example, we are using virtual appliance. Other choices are virtual network gateway, virtual network, internet, and none.


Add route
cesrt

* Route name
ToPrivateSubnet

* Address prefix
10.0.1.0/24

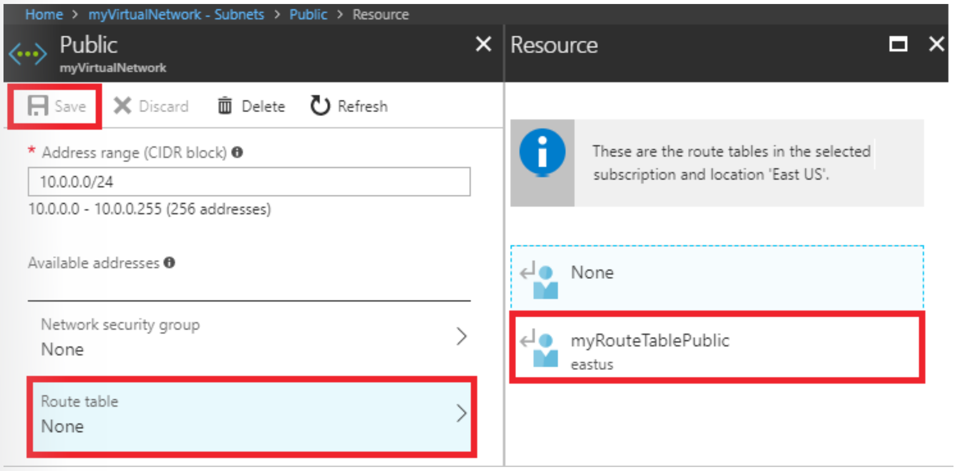
Next hop type
Virtual appliance

* Next hop address
10.0.2.4

Notice this route applies to any address prefixes in 10.0.1.0/24 (private subnet). Traffic headed to these addresses will be sent to the virtual appliance with a 10.0.2.4 address.

Associate Route to Subnet

Each subnet can have zero or one route table associated to it. In this example, our Public subnet will be associated with the routing table.



* Address range (CIDR block)
10.0.0.0/24
10.0.0.0 - 10.0.0.255 (256 addresses)

Available addresses

Network security group
None

Route table
None

These are the route tables in the selected subscription and location 'East US'.

None

myRouteTablePublic
eastus

✓ In this case the virtual appliance should not have a public IP address and IP forwarding should be enabled. Be sure to try the practice.

Routing Algorithms

So far routing has been fairly straightforward, but what if a destination address matches two routes in the routing table? Azure sorts this out in two ways: longest prefix match algorithm, and route priorities.

Longest prefix match algorithm

For example, if the destination address is 10.0.0.5 and there are two routes: One route specifies the 10.0.0.0/24 address prefix, while the other route specifies the 10.0.0.0/16 address prefix. In this case, Azure selects a route using the longest prefix match algorithm, which is the 10.0.0.0/24 route.

Source	Address prefixes	Next hop type
System	10.0.0.0/24	Internet (selected)
System	10.0.0.0/16	Virtual network gateway

Route priorities

When the address prefixes are the same, Azure selects the route type, based on the following priority:

1. User-defined route
2. BGP route
3. System route

In our example, address 10.0.0.5, Azure selects the route with the User source, because user-defined routes are higher priority than system default routes.

Source	Address prefixes	Next hop type
User	10.0.0.0/16	Internet (selected)
System	10.0.0.0/16	Virtual network gateway

For more information, you can see:

How Azure selects a route - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview#how-azure-selects-a-route**

Demonstration - Custom Routing Tables

In this demonstration, you will learn how to create a route table, define a custom route, and associate the route with a subnet.

Note: This demonstration requires a virtual network with at least one subnet.

Create a routing table

1. Access the Azure portal.
2. On the upper-left side of the screen, select **Services**, and then navigate to **Route tables**.
3. Select **+ Add**.

    - **Name**: *myRouteTablePublic*
    - **Subscription**: *select your subscription*
    - **Resource group**: *create or select a resource group*
    - **Location**: *select your location*
    - **BGP route propagation**: *Enabled*

4. Select **Create**.
5. Wait for the new routing table to be deployed.

Add a route

1. Select your new routing table, and then select **Routes**.

2. Select **+ Add**.

   - **Name**: *ToPrivateSubnet*
   - **Address prefix**: *10.0.1.0/24*
   - **Next hop type**: *Virtual appliance*
   - **Next hop address**: *10.0.2.4*

3. Read the information note: Ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings.
4. Select **Create**.
5. Wait for the new route to be deployed.

Associate a route table to a subnet

1. Navigate to the subnet you want to associate with the routing table.
2. Select **Route table**.
3. Select your new routing table, **myRouteTablePublic**.
4. **Save** your changes.

Use PowerShell to view your routing information

1. Open the Cloud Shell.
2. View information about your new routing table.

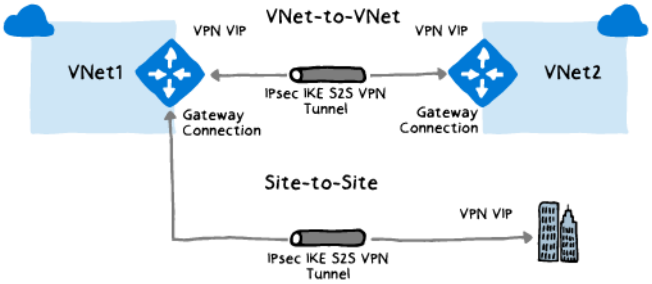
```
Get-AzRouteTable
```

1. Verify the **Routes** and **Subnet** information is correct.

Intersite Connectivity

VNet-to-VNet Connections

You can connect your VNets with a VNet-to-VNet VPN connection. Using this connection method, you create a VPN gateway in each virtual network. The VPN gateway can also be used to provide a connection to an on-premises network. This is called a Site-to-Site (S2S) connection. In both cases a secure tunnel using IPsec/IKE provides the communication between the networks.



With a VNet-to-VNet connection your VNets can be:

- in the same or different regions.
- in the same or different subscriptions.
- in the same or different deployment models.
- in Azure or on-premises.

Benefits

Cross region geo-redundancy and geo-presence

- You can set up your own geo-replication or synchronization with secure connectivity without going over Internet-facing endpoints.
- With Azure Traffic Manager and Load Balancer, you can set up highly available workload with geo-redundancy across multiple Azure regions.

Regional multi-tier applications with isolation or administrative boundary

- Within the same region, you can set up multi-tier applications with multiple virtual networks connected together due to isolation or administrative requirements.
- VNet-to-VNet communication can be combined with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

✓ You will use VNet-to-VNet connections when you cannot use VNet peering.

✓ Connections to on-premises viirtual networks are called Site-to-Site (S2S) connections.

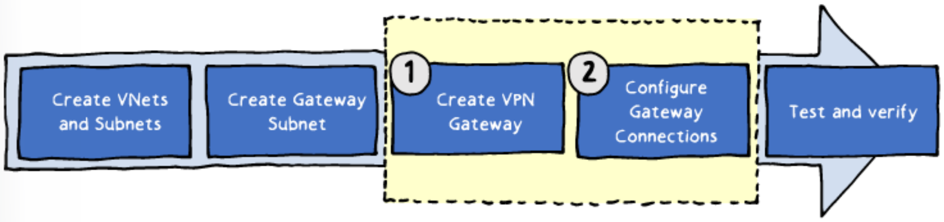
For more information, you can see:

VNet-to-VNet Connectivity - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-vnet-vnet-resource-manager-portal#vnet-to-vnet**

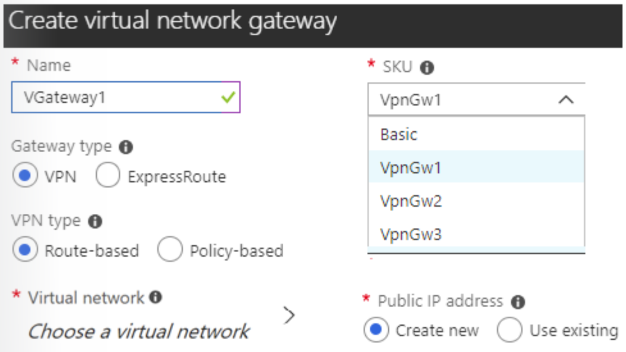
Site-to-Site Connectivity - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-site-to-site-resource-manager-portal**

Implementing VNet-to-VNet VPN

The steps to implement VNet-to-VNet connections are the same as for VNet peering with the addition of configuring the VPN Gateway. You still need to create VNets, subnets, and a gateway subnet in each virtual network. When everything is configured you will need to test and verify.



Create VPN Gateway (1)



Create virtual network gateway

* Name
VGateway1 ✓

Gateway type
● VPN ○ ExpressRoute

VPN type
● Route-based ○ Policy-based

* Virtual network
Choose a virtual network

* SKU
VpnGw1
Basic
VpnGw1
VpnGw2
VpnGw3

* Public IP address
● Create new ○ Use existing

- **Name and Gateway Type.** Name your gateway and use the VPN Gateway type.
- **VPN Type.** Most VPN types are Route-based.
- **SKU.** Use the drop-down to select a **gateway SKU**[2]. Your choice will affect the number of tunnels you can have and the aggregate throughput benchmark. The benchmark is based on measurements of multiple tunnels aggregated through a single gateway. It is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.
- **Virtual Networks.** Associate a virtual network with the gateway. Before you do this, you must configure the gateway subnet. Each virtual network will need its own VPN gateway.
- **IP Address.** The gateway needs a public IP address to its IP configuration to enable it to communicate with the remote network.

It can take up to 45 minutes to provision the VPN gateway.

✓ Be sure to use the reference link and read more about the VPN gateway configuration. And, continue to the next page for configuring the connections between the VPN gateways.

For more information, you can see:

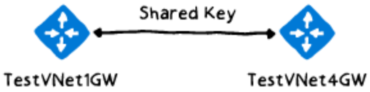
Create a virtual network gateway - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-vnet-vnet-resource-manager-portal#VNetGateway**

[2] https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings

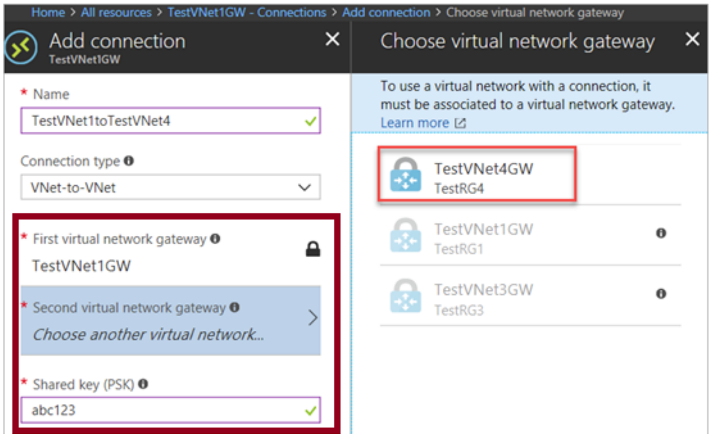
VPN Types - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings#vpntype**

Configuring Gateway Connections

Once your VPN gateways are created, you can create the connection between them. If your VNets are in the same subscription, you can use the portal.



For example, you could add a connection between TestVNet1GW and TestVNet4GW. In the **Shared key** field, type a shared key for your connection. You can generate or create this key yourself.


Home > All resources > TestVNet1GW - Connections > Add connection > Choose virtual network gateway

Add connection
TestVNet1GW

* Name
TestVNet1toTestVNet4 ✓

Connection type
VNet-to-VNet

* First virtual network gateway
TestVNet1GW

* Second virtual network gateway
Choose another virtual network...

* Shared key (PSK)
abc123 ✓

Choose virtual network gateway

To use a virtual network with a connection, it must be associated to a virtual network gateway.
Learn more

TestVNet4GW
TestRG4

TestVNet1GW
TestRG1

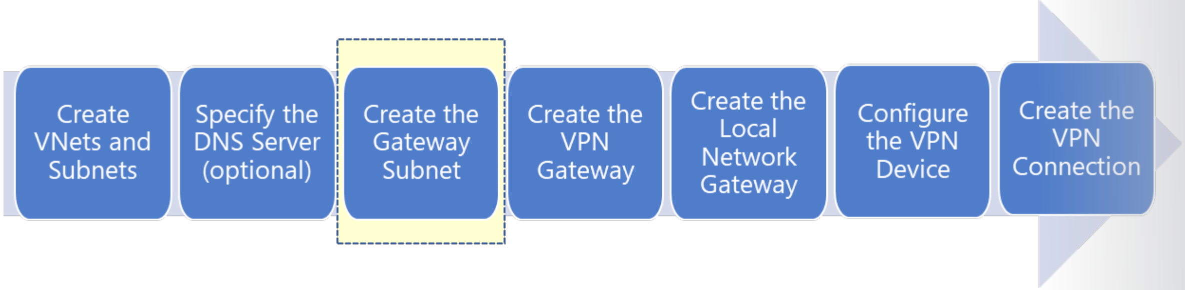
TestVNet3GW
TestRG3

✓ If your VNets are in different subscriptions, you must use PowerShell to make the connection. You can use the **New-AzureRmVirtualNetworkGatewayConnection**[3] command. This command can also be used for Site-to-Site connections.

For more information, you can see:

Configure the TestVNet1 gateway connection - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-vnet-vnet-resource-manager-portal#TestVNet1Connection**

Create the Gateway Subnet

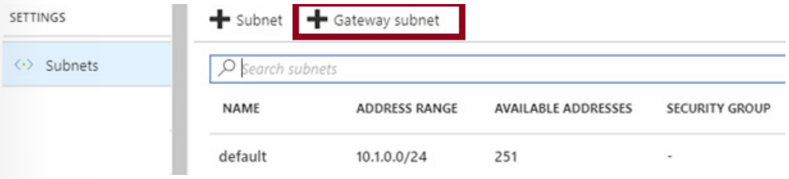

Create VNets and Subnets
Specify the DNS Server (optional)
Create the Gateway Subnet
Create the VPN Gateway
Create the Local Network Gateway
Configure the VPN Device
Create the VPN Connection

[3] https://docs.microsoft.com/en-us/powershell/module/azurerm.network/new-azurermvirtualnetworkgatewayconnection?view=azurermps-6.3.0

Before creating a virtual network gateway for your virtual network, you first need to create the gateway subnet. The gateway subnet contains the IP addresses that are used by the virtual network gateway. If possible, it's best to create a gateway subnet by using a CIDR block of /28 or /27 to provide enough IP addresses to accommodate future additional configuration requirements.

When you create your gateway subnet, gateway VMs are deployed to the gateway subnet and configured with the required VPN gateway settings. You must never deploy other resources (for example, additional VMs) to the gateway subnet. The gateway subnet must be named *GatewaySubnet*.

To deploy a gateway in your virtual network simply add a gateway subnet.

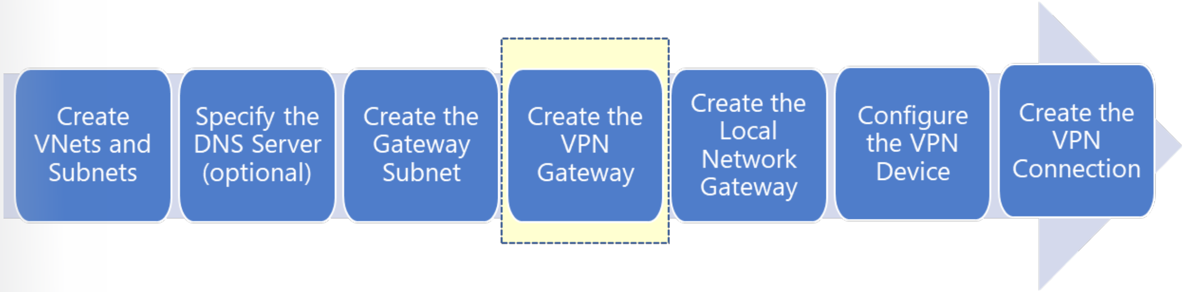


NAME	ADDRESS RANGE	AVAILABLE ADDRESSES	SECURITY GROUP
default	10.1.0.0/24	251	-

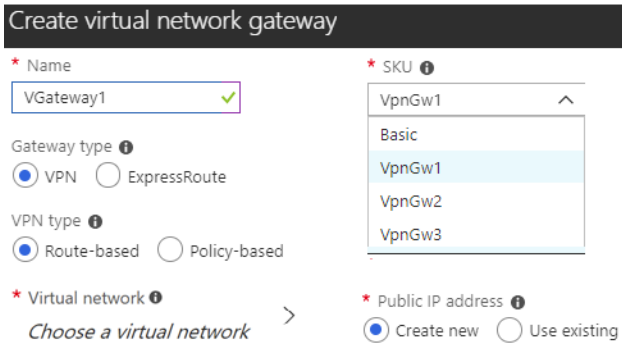
✓ When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your VPN gateway to stop functioning as expected.

✓ This is the same step in configuring VNet Peering.

Create the VPN Gateway


Create VNets and Subnets
Specify the DNS Server (optional)
Create the Gateway Subnet
Create the VPN Gateway
Create the Local Network Gateway
Configure the VPN Device
Create the VPN Connection

A VPN gateway is a specific type of virtual network gateway that is used to send encrypted traffic between an Azure virtual network and an on-premises location over the public Internet. You can also use a VPN gateway to send encrypted traffic between Azure virtual networks over the Microsoft network. Each virtual network can have only one VPN gateway. However, you can create multiple connections to the same VPN gateway. When you create multiple connections to the same VPN gateway, all VPN tunnels share the available gateway bandwidth.



Create virtual network gateway

* Name
VGateway1 ✓

Gateway type
● VPN ○ ExpressRoute

VPN type
● Route-based ○ Policy-based

* Virtual network
Choose a virtual network

* SKU
VpnGw1
Basic
VpnGw1
VpnGw2
VpnGw3

* Public IP address
● Create new ○ Use existing

- **Name and Gateway Type**. Name your gateway and use the VPN Gateway type.
- **VPN Type**. Most VPN types are Route-based.
- **SKU**. Use the drop-down to select a gateway SKU. Your choice will affect the number of tunnels you can have and the aggregate throughput benchmark. The benchmark is based on measurements of multiple tunnels aggregated through a single gateway. It is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.
- **Virtual Networks**. Associate a virtual network with the gateway. Before you do this, you must configure the Gateway subnet. Each virtual network will need its own VPN gateway.
- **Public IP Address**. The gateway needs a public IP address to enable it to communicate with the remote network. Make a note of this information. You will need the address when you configure your VPN device.

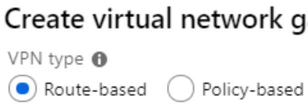
It can take up to 45 minutes to provision the VPN gateway.

✓ After the gateway is created, view the IP address that has been assigned to it by looking at the virtual network in the portal. The gateway should appear as a connected device. In this last step you will create a connection for the device.

VPN Types

When you create the virtual network gateway for a VPN gateway configuration, you must specify a VPN type. The VPN type that you choose depends on the connection topology that you want to create. For example, a Point-to-Site (P2S) connection requires a Route-based VPN type. A VPN type can also depend on the hardware that you are using. Site-to-Site (S2S) configurations require a VPN device. Some VPN devices only support a certain VPN type.

The VPN type you select must satisfy all the connection requirements for the solution you want to create. For example, if you want to create a S2S VPN gateway connection and a P2S VPN gateway connection for the same virtual network, you would use VPN type Route-based because P2S requires a Route-based VPN type. You would also need to verify that your VPN device supported a Route-based VPN connection.



Create virtual network gateway

VPN type
● Route-based ○ Policy-based

There are two VPN types:

- **Policy-based VPNs**. Policy-based VPNs encrypt and direct packets through IPsec tunnels based on the IPsec policies configured with the combinations of address prefixes between your on-premises network and the Azure VNet. The policy (or traffic selector) is usually defined as an access list in the VPN device configuration. When using a Policy-based VPN, keep in mind the following limitations:

- Policy-Based VPNs can only be used on the Basic gateway SKU and is not compatible with other gateway SKUs.
- You can have only 1 tunnel when using a Policy-based VPN.
- You can only use Policy-based VPNs for S2S connections, and only for certain configurations. Most VPN Gateway configurations require a Route-based VPN.

- **Route-based VPNs**. Route-based VPNs use *routes* in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The tunnel interfaces then encrypt or decrypt the packets in and out of the tunnels. The policy (or traffic selector) for Route-based VPNs are configured as any-to-any (or wild cards).

Once a virtual network gateway has been created, you can't change the VPN type.

Gateway SKUs

When you create a virtual network gateway, you need to specify the gateway SKU that you want to use. Select the SKU that satisfies your requirements based on the types of workloads, throughputs, features, and SLAs.

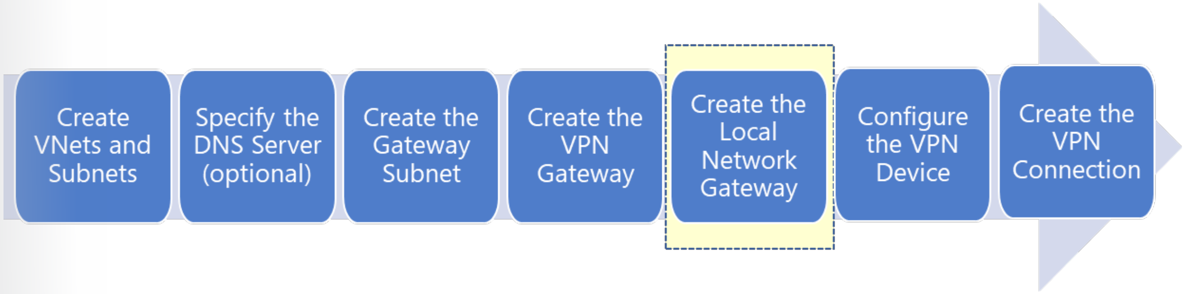
SKU	S2S/VNet-to-VNet Tunnels	P2S SSTP Connections	P2S IKEv2 Connections	Aggregate Throughput Benchmark
Basic	Max. 10	Max. 128	Not Supported	100 Mbps
VpnGw1	Max. 30	Max. 128	Max. 250	650 Mbps
VpnGw2	Max. 30	Max. 128	Max. 500	1 Gbps
VpnGw3	Max. 30	Max. 128	Max. 1000	1.25 Gbps

Aggregate Throughput Benchmark is based on measurements of multiple tunnels aggregated through a single gateway. The Aggregate Throughput Benchmark for a VPN Gateway is S2S + P2S combined. The Aggregate Throughput Benchmark is not a guaranteed throughput due to Internet traffic conditions and your application behaviors.

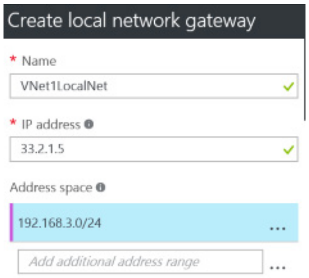
These connection limits are separate. For example, you can have 128 SSTP connections and also 250 IKEv2 connections on a VpnGw1 SKU.

✓ The Basic SKU is considered a legacy SKU. The Basic SKU has certain feature limitations. You can't resize a gateway that uses a Basic SKU to one of the new gateway SKUs, you must instead change to a new SKU, which involves deleting and recreating your VPN gateway.

Create the Local Network Gateway


Create VNets and Subnets
Specify the DNS Server (optional)
Create the Gateway Subnet
Create the VPN Gateway
Create the Local Network Gateway
Configure the VPN Device
Create the VPN Connection

The local network gateway typically refers to the on-premises location. You give the site a name by which Azure can refer to it, then specify the IP address of the on-premises VPN device for the connection. You also specify the IP address prefixes that will be routed through the VPN gateway to the VPN device. The address prefixes you specify are the prefixes located in the on-premises network.



Create local network gateway

* Name: VNet1LocalNet

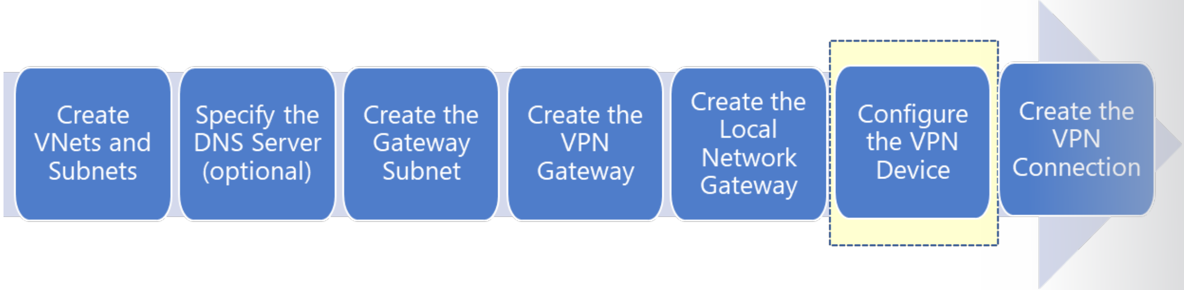
* IP address: 33.2.1.5

Address space: 192.168.3.0/24

IP Address. The public IP address of the local gateway.

Address Space. One or more IP address ranges (in CIDR notation) that define your local network's address space. For example: 192.168.0.0/16. If you plan to use this local network gateway in a BGP-enabled connection, then the minimum prefix you need to declare is the host address of your BGP Peer IP address on your VPN device.

Configure the On-Premises VPN Device



Create VNets and Subnets
Specify the DNS Server (optional)
Create the Gateway Subnet
Create the VPN Gateway
Create the Local Network Gateway
Configure the VPN Device
Create the VPN Connection

Microsoft has validated a list of standard VPN devices that should work well with the VPN gateway. This list was created in partnership with device manufacturers like Cisco, Juniper, Ubiquiti, and Barracuda Networks. If you don't see your device listed in the validated VPN devices table (reference link), your device may still work with a Site-to-Site connection. Contact your device manufacturer for additional support and configuration instructions.

To configure your VPN device, you need the following:

- **A shared key**. This is the same shared key that you will specify when creating the VPN connection (next step).
- **The public IP address of your VPN gateway**. When you created the VPN gateway you may have configured a new public IP address or used an existing IP address.

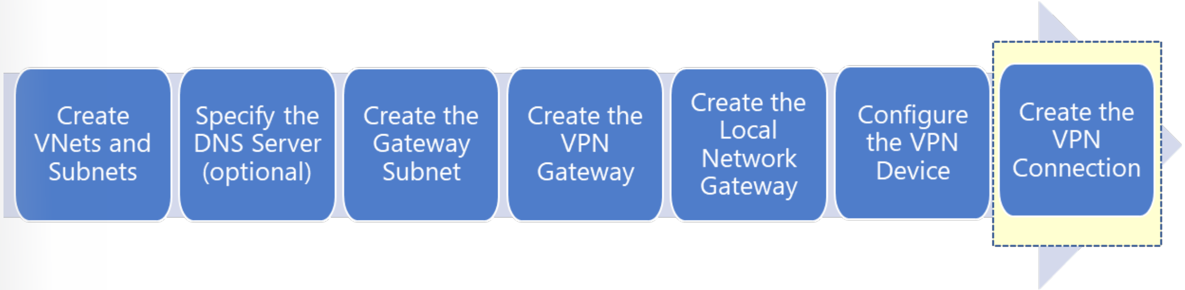
✓ Depending on the VPN device that you have, you may be able to **download a VPN device configuration script**[4].

For more information, you can see:

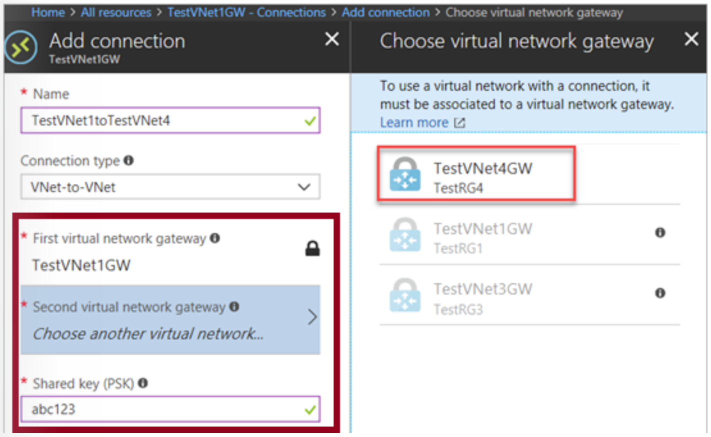
[4] https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-download-vpndevicescript

Validated VPN devices list - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-devices#devicetable**[5]

Create the VPN Connection


Create VNets and Subnets
Specify the DNS Server (optional)
Create the Gateway Subnet
Create the VPN Gateway
Create the Local Network Gateway
Configure the VPN Device
Create the VPN Connection

Once your VPN gateways are created, you can create the connection between them. If your VNets are in the same subscription, you can use the portal.



Home > All resources > TestVNet1GW - Connections > Add connection > Choose virtual network gateway

Add connection
TestVNet1GW

* Name
TestVNet1toTestVNet4

Connection type
VNet-to-VNet

* First virtual network gateway
TestVNet1GW

* Second virtual network gateway
Choose another virtual network...

* Shared key (PSK)
abc123

Choose virtual network gateway

To use a virtual network with a connection, it must be associated to a virtual network gateway. Learn more

TestVNet4GW
TestRG4

TestVNet1GW
TestRG1

TestVNet3GW
TestRG3

- **Name**. Enter a name for your connection.
- **Connection type**. Select VNet-to-VNet from the drop-down.
- **First virtual network gateway**. This field value is automatically filled in because you're creating this connection from the specified virtual network gateway.
- **Second virtual network gateway**. This field is the virtual network gateway of the VNet that you want to create a connection to.
- **Shared key (PSK)**. In this field, enter a shared key for your connection. You can generate or create this key yourself. In a site-to-site connection, the key you use is the same for your on-premises device and your virtual network gateway connection. The concept is similar here, except that rather than connecting to a VPN device, you're connecting to another virtual network gateway.

✓ If your VNets are in different subscriptions, you must use PowerShell to make the connection. You can use the New-AzVirtualNetworkGatewayConnection.

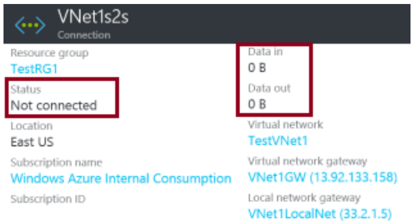
[5] https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-devices

Verify the VPN Connection

After you have configured all the Site-to-Site components it is time to verify that everything is working. You can verify the connections either in the portal, or by using PowerShell.

Portal

When you view your connection in the Azure portal the Status should be Succeeded or Connected. Also, you should have data flowing in the Data in and Data out information.



VNet1s2s
Connection

Resource group
TestRG1

Status
Not connected

Location
East US

Subscription name
Windows Azure Internal Consumption

Subscription ID

Data in
0 B

Data out
0 B

Virtual network
TestVNet1

Virtual network gateway
VNet1GW (13.92.133.158)

Local network gateway
VNet1LocalNet (33.2.1.5)

PowerShell

To verify your connection with PowerShell, use the Get-AzVirtualNetworkGatewayConnection cmdlet. For example,

```
Get-AzVirtualNetworkGatewayConnection -Name MyGWConnection -ResourceGroupName MyRG
```

After the cmdlet has finished, view the values. The connection status should show 'Connected' and you can see ingress and egress bytes.

```
"connectionStatus": "Connected",
"ingressBytesTransferred": 33509044,
"egressBytesTransferred": 4142431
```

Demonstration - VNet to VNet Connections

Note: This demonstration works best with two virtual networks with subnets. All the steps are in the portal.

Explore the Gateway subnet blade

1. For one of your virtual network, select the **Subnets** blade.
2. Select **+ Gateway subnet**.

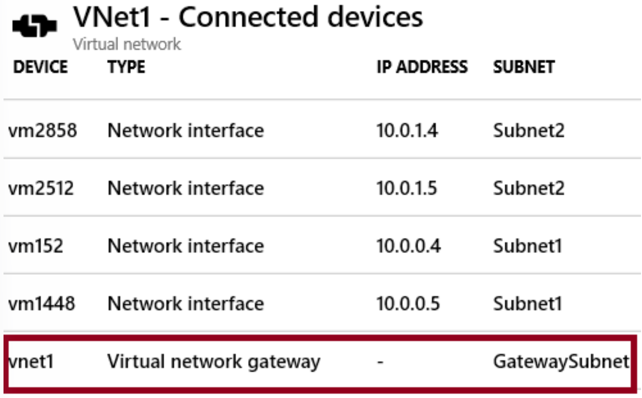
   Notice the name of the subnet cannot be changed.

   Notice the **address range** of the gateway subnet. The address must be contained by the address space of the virtual network.
3. Remember each virtual network needs a gateway subnet.
4. Close the Add gateway subnet page. You do not need to save your changes.

Explore the Connected Devices blade

1. For the virtual network, select the **Connected Devices** blade.

2. After a gateway subnet is deployed it will appear on the list of connected devices.



VNet1 - Connected devices
Virtual network

DEVICE	TYPE	IP ADDRESS	SUBNET
vm2858	Network interface	10.0.1.4	Subnet2
vm2512	Network interface	10.0.1.5	Subnet2
vm152	Network interface	10.0.0.4	Subnet1
vm1448	Network interface	10.0.0.5	Subnet1
vnet1	Virtual network gateway	-	GatewaySubnet

Explore adding a virtual network gateway

1. Search for **Virtual network gateways**.
2. Click **+ Add**.
3. Review each setting for the virtual netowrk gateway.
4. Use the Information icons to learn more about the settings.
5. Notice the **Gateway type**, **VPN type**, and **SKU**.
6. Notice the need for a **Public IP address**.
7. Remember each virtual network will need a virtual network gateway.
8. Close the Add virtual network gateway. You do not need to save your changes.

Explore adding a connection between the virtual networks

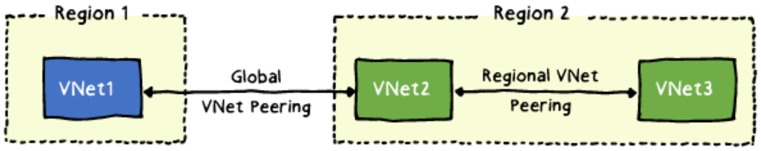
1. Search for **Connections**.
2. Click **+ Add**.
3. Notice the **Connection type** can be VNet-to-VNet, Site-to-Site (IPsec), or ExpressRoute.
4. Provide enough information, so you can click the **Ok** button.
5. On the **Settings** page, notice that you will need select the two different virtual networks.
6. Read the Help information on the **Establilsh bidirectional connnectivity** checkbox.
7. Notice the **Shared key (PSK)** information.
8. Close the Add connection page. You do not need to save your changes.

Virtual Network Peering

VNet Peering

Perhaps the simplest and quickest way to connect your VNets is to use VNet peering. Virtual network peering enables you to seamlessly connect two Azure **virtual networks**[6]. Once peered, the virtual networks appear as one, for connectivity purposes. There are two types of VNet peering.

- **Regional VNet peering** connects Azure virtual networks in the same region.
- **Global VNet peering** connects Azure virtual networks in different regions.



The benefits of using local or global virtual network peering, include:

- **Private**. Network traffic between peered virtual networks is private. Traffic between the virtual networks is kept on the Microsoft backbone network. No public Internet, gateways, or encryption is required in the communication between the virtual networks.
- **Performance**. A low-latency, high-bandwidth connection between resources in different virtual networks.
- **Communication**. The ability for resources in one virtual network to communicate with resources in a different virtual network, once the virtual networks are peered.
- **Seamless**. The ability to transfer data across Azure subscriptions, deployment models, and across Azure regions.
- **No disruption**. No downtime to resources in either virtual network when creating the peering, or after the peering is created.

✓ The default VNet peering configuration provides full connectivity. Can you see how network security groups could be applied to block or deny access to specific subnets or virtual machines?

For more information, you can see:

Virtual network peering - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-peering-overview**

Configure VNet Peering

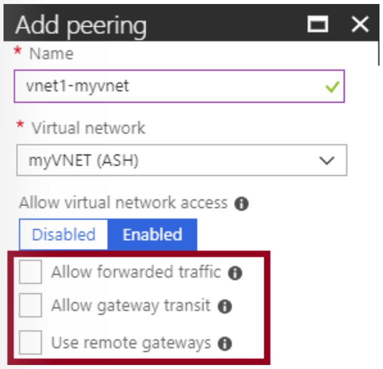
Here are the steps to configure VNet peering. Notice you will need two virtual networks. To test the peering, you will need a virtual machine in each network. Initially, the VMs will not be able to communicate (ping), but after configuration the communication will work. The step that is new is configuring the peering of the virtual networks.

1. Create two virtual networks.
2. **Peer the virtual networks**.

[6] https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview

3. Create virtual machines in each virtual network.
4. Test the communication between the virtual machines.

To configure the peering use the **Add peering** page. There are only a few optional configuration parameters to consider.



Add peering

* Name

vnet1-myvnet

* Virtual network

myVNET (ASH)

Allow virtual network access

Disabled Enabled

□ Allow forwarded traffic

□ Allow gateway transit

□ Use remote gateways

- **Allow forwarded traffic.** Allows traffic not originating from within the peer virtual network into your virtual network.
- **Allow gateway transit.** Allows the peer virtual network to use your virtual network gateway. The peer cannot already have a gateway configured.
- **Use remote gateways.** Use your peer's virtual gateway. Only one virtual network can have this enabled.

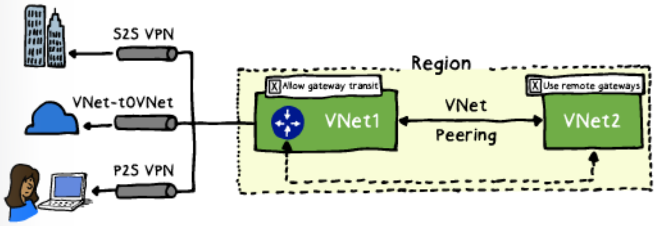
✓ You must configure peering on each virtual network. If you select 'allow gateway transit' on one virtual network; then you should select 'use remote gateways' on the other virtual network.

Gateway Transit

When you allow gateway transit the virtual network can communicate to resources outside the peering. For example, the subnet gateway could:

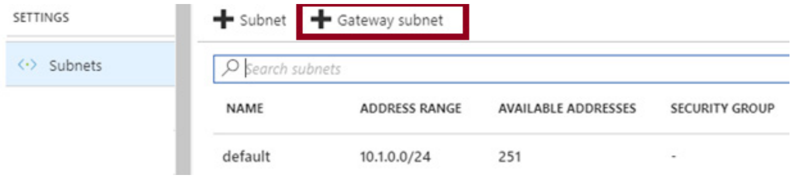
- Use a site-to-site VPN to connect to an on-premises network.
- Use a VNet-to-VNet connection to another virtual network.
- Use a point-to-site VPN to connect to a client.

In these scenarios, gateway transit allows peered virtual networks to share the gateway and get access to resources. This means you do not need to deploy a VPN gateway in the peer virtual network.



When you create your virtual network gateway, gateway VMs are deployed to the gateway subnet and configured with the required VPN gateway settings. You must never deploy anything else (for example, additional VMs) to the gateway subnet. The gateway subnet must be named 'GatewaySubnet'.

To deploy a gateway in your virtual network simply add a gateway subnet.



NAME	ADDRESS RANGE	AVAILABLE ADDRESSES	SECURITY GROUP
default	10.1.0.0/24	251	-

This architecture is often referred to as a hub-spoke topology in Azure. In the illustration at the beginning of this topic, VNet1 is the hub and acts as a central point of connectivity to external resources. VNet2 is the spoke that peers with the hub and can be used to isolate workloads.

✓ When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your VPN gateway to stop functioning as expected.

For more information, you can see:

Gateway transit - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-peering-gateway-transit?toc=%2fazure%2fvirtual-network%2ftoc.json**

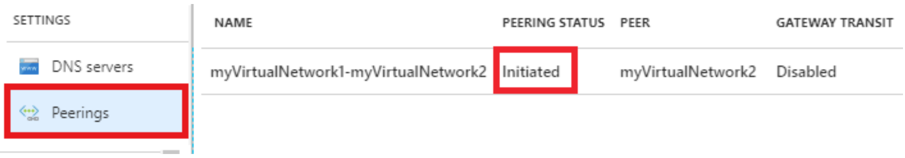
PowerShell Example - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-peering-gateway-transit?toc=%2fazure%2fvirtual-network%2ftoc.json#powershell-sample**[7]

Hub and spoke - **https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/hybrid-networking/hub-spoke**

Gateway Subnet - **https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings#gwsub**[8]

Global VNet Peering

Global VNet peering in is the ability to peer virtual networks across regions. You can check the status of VNet peering. The peering is not successfully established until the peering status for both virtual network peerings shows **Connected**.



NAME	PEERING STATUS	PEER	GATEWAY TRANSIT
myVirtualNetwork1-myVirtualNetwork2	Initiated	myVirtualNetwork2	Disabled

- **Initiated.** When you create the peering to the second virtual network from the first virtual network, the peering status is Initiated.
- **Connected.** When you create the peering from the second virtual network to the first virtual network, its peering status is Connected. If you view the peering status for the first virtual network, you see its status changed from Initiated to Connected.

[7] https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-peering-gateway-transit?toc=%2fazure%2fvirtual-network%2ftoc.json
[8] https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings

Requirements and constraints

The benefits and configuration steps are the same as for regional peering, but there are some special requirements.

- **Public clouds.** The virtual networks can exist in any Azure public cloud region, but not in Azure national clouds. National clouds are physical and logical network-isolated instances of Microsoft enterprise cloud services, which are confined within the geographic borders of specific countries and operated by local personnel. There are very specific customer requirements to using and operating national clouds.
- **Virtual network resources.** Resources in one virtual network cannot communicate with the IP address of an Azure internal load balancer in the peered virtual network. The load balancer and the resources that communicate with it must be in the same virtual network.
- **Gateway transit.** You should not configure 'use remote gateways' or 'allow gateway transit'. Gateway transit only applies to regional VNet peering.
- **Transitivity.** VNet global peerings are not transitive meaning downstream VNets in one region cannot talk with downstream VNets in another region. If you create peerings between VNet1-VNet2 and VNet2-VNet3, there is no implied peering between VNet1 and VNet3.
- **Virtual machines.** Peering **high performance compute**[9] and **GPU**[10] virtual machines is not supported. For example, H, NC, NV, NCv2, NCv3, and ND series VMs.

For more information, you can see:

How to setup Global VNet peering in Azure - **https://blogs.msdn.microsoft.com/azureedu/2018/04/24/how-to-setup-global-vnet-peering-in-azure/**

Requirements and constraints - **https://docs.microsoft.com/en-us/azure/virtual-network/virtual-network-manage-peering#requirements-and-constraints**

Service Chaining

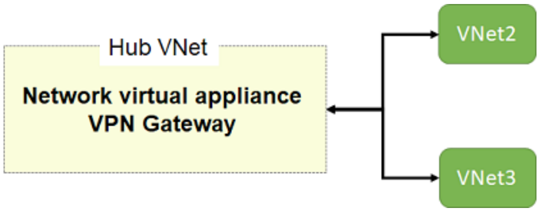
VNet Peering is nontransitive. This means that if you establish VNet Peering between VNet1 and VNet2 and between VNet2 and VNet3, VNet Peering capabilities do not apply between VNet1 and VNet3. However, you can leverage user-defined routes and service chaining to implement custom routing that will provide transitivity. This allows you to:

- Implement a multi-level hub and spoke architecture.
- Overcome the limit on the number of VNet Peerings per virtual network.

Hub and spoke architecture

You can deploy hub-and-spoke networks, where the hub virtual network can host infrastructure components such as a network virtual appliance or VPN gateway. All the spoke virtual networks can then peer with the hub virtual network. Traffic can flow through network virtual appliances or VPN gateways in the hub virtual network.

[9] https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes-hpc
[10] https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes-gpu



User-defined routes and service chaining

Virtual network peering enables the next hop in a user-defined route to be the IP address of a virtual machine in the peered virtual network, or a VPN gateway.

Service chaining enables you to direct traffic from one virtual network to a virtual appliance, or virtual network gateway, in a peered virtual network, through user-defined routes.

Review Questions

Module 4 Review Questions

Limitations on Virtual Networks

You manage the Azure environment for your organization. The environment supports 45 different applications.

Each application must use a separate resource group and must use a unique virtual network.

You need to add several new applications to the environment.

What issues might you face? How should you resolve the issues?

Suggested Answer ↓

By default, you can create up to 50 virtual networks per subscription per region, although you can increase this limit to 500 by contacting Azure support.

Multiple NICs in virtual machines

You manage the Azure environment for your organization. You deploy a new application server virtual machine (VM).

The application server must communicate with internal on-premises resources , and must also respond to communications initiated from external users.

How should you configure the VM? How can you deploy the VM with multiple NICs?

Suggested Answer ↓

A VM with single NIC cannot be configured to use multiple NICs (and vice-versa) once it is deployed. You must delete and recreate the VM. To deploy a VM that has multiple NICs you must use Azure PowerShell.

Av2, Dv2/Dv3 or DSv2/DSv3 series VM sizes all support multiple NICs

User-defined Routing

You manage the Azure environment for your organization. You deploy a new server application virtual machine (VM) in your environment. The application server provides services to internal and external users through a web portal.

When internal users navigate to the portal, they must be directed to an administrative sign in page. External users must be directed to an inventory page and must not be required to sign in.

What should you do?

Suggested Answer ↓

You can create user-defined routes in Azure to override Azure's default system routes, or to add additional routes to a subnet's route table. In Azure, you create a route table, then associate the route table to zero or more virtual network subnets.

For this specific VM you want to utilize multiple NICs, and route traffic using user-defined custom routes depending on the IP address that attempts to access the resource.

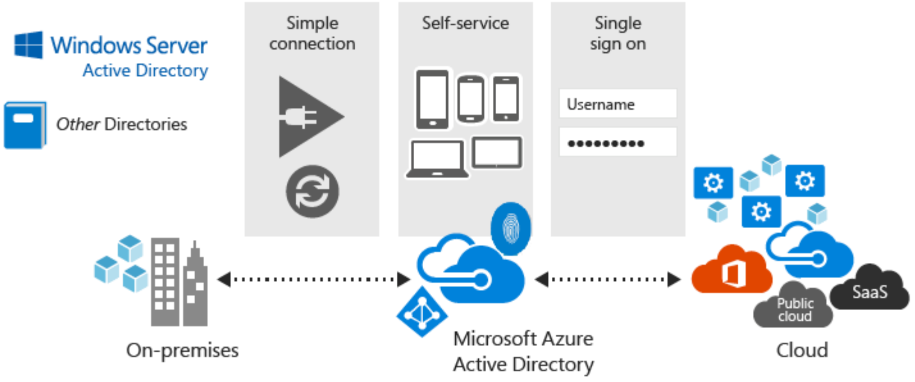
Module 5 Module Managing Identities

Managing Azure Active Directory (AAD)

Azure Active Directory

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.



Identity manage capabilities and integration

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 assure corporate compliance goals are met.

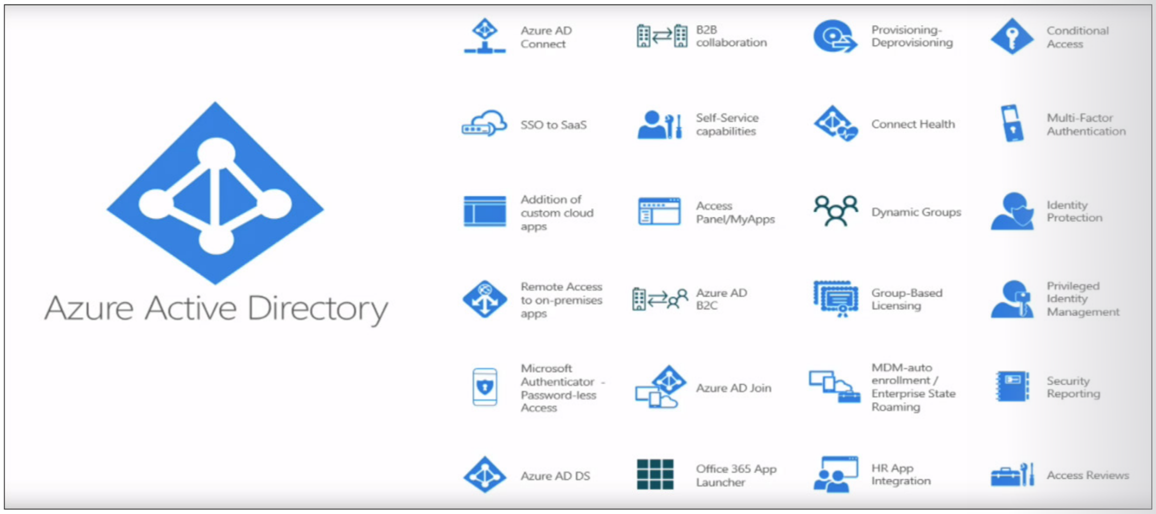
Additionally, Azure AD can be integrated with an existing Windows Server Active Directory, giving organizations the ability to leverage their existing on-premises identity investments to manage access to cloud based SaaS applications.

✓ 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 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.

For more information, you can see: What is Azure Active Directory? - **https://docs.microsoft.com/en-us/azure/active-directory/active-directory-whatis**

Azure Active Directory Benefits

- **Single sign-on to any cloud or on-premises web app.** Azure Active Directory provides secure single sign-on to cloud and on-premises applications including Microsoft Office 365 and thousands of SaaS applications such as Salesforce, Workday, DocuSign, ServiceNow, and Box.
- **Works with iOS, Mac OS X, Android, and Windows devices.** Users can launch applications from a personalized web-based access panel, mobile app, Office 365, or custom company portals using their existing work credentials—and have the same experience whether they're working on iOS, Mac OS X, Android, and Windows devices.
- **Protect on-premises web applications with secure remote access.** Access your on-premises web applications from everywhere and protect with multi-factor authentication, conditional access policies, and group-based access management. Users can access SaaS and on-premises web apps from the same portal.
- **Easily extend Active Directory to the cloud.** Connect Active Directory and other on-premises directories to Azure Active Directory in just a few clicks and maintain a consistent set of users, groups, passwords, and devices across both environments.
- **Protect sensitive data and applications.** Enhance application access security with unique identity protection capabilities that provide a consolidated view into suspicious sign-in activities and potential vulnerabilities. Take advantage of advanced security reports, notifications, remediation recommendations and risk-based policies to protect your business from current and future threats.
- **Reduce costs and enhance security with self-service capabilities.** Delegate important tasks such as resetting passwords and the creation and management of groups to your employees. Providing self-service application access and password management through verification steps can reduce helpdesk calls and enhance security.



✓ What reasons do you have for considering Azure Active Directory?

Active Directory Domain Services (AD DS)

AD DS is the traditional deployment of Windows Server-based Active Directory on a physical or virtual server. Although AD DS is commonly considered to be primarily a directory service, it is only one component of the Windows Active Directory suite of technologies, which also includes Active Directory Certificate Services (AD CS), Active Directory Lightweight Directory Services (AD LDS), Active Directory Federation Services (AD FS), and Active Directory Rights Management Services (AD RMS). Although you can deploy and manage AD DS in Azure virtual machines it's recommended you use Azure AD instead, unless you are targeting IaaS workloads that depend on AD DS specifically.

Azure AD is different from AD DS

Although Azure AD has many similarities to AD DS, there are also many differences. It is important to realize that using Azure AD is different from deploying an Active Directory domain controller on an Azure virtual machine and adding it to your on-premises domain. Here are some characteristics of Azure AD that make it different.

- **Identity solution.** Azure AD is primarily an identity solution, and it is designed for Internet-based applications by using HTTP and HTTPS communications.
- **REST API Querying.** Because Azure AD is HTTP/HTTPS based, it cannot be queried through LDAP. Instead, Azure AD uses the REST API over HTTP and HTTPS.
- **Communication Protocols.** Because Azure AD is HTTP/HTTPS based, it does not use Kerberos authentication. Instead, it uses HTTP and HTTPS protocols such as SAML, WS-Federation, and OpenID Connect for authentication (and OAuth for authorization).

- **Federation Services.** Azure AD includes federation services, and many third-party services (such as Facebook).
- **Flat structure.** Azure AD users and groups are created in a flat structure, and there are no Organizational Units (OUs) or Group Policy Objects (GPOs).

✓ Azure AD is a managed service. You only manage the users, groups, and policies. Deploying AD DS with virtual machines using Azure means that you manage the deployment, configuration, virtual machines, patching, and other backend tasks. Do you see the difference?

Azure Active Directory Editions

Azure Active Directory comes in four editions—**Free**, **Basic**, **Premium P1**, and **Premium P2**. The Free edition is included with an Azure subscription. The Azure Active Directory Basic, Premium P1, and Premium P2 editions are built on top of your existing free directory, providing enterprise class capabilities spanning self-service, enhanced monitoring, security reporting, Multi-Factor Authentication (MFA), and secure access for your mobile workforce.

Free
Basic
Premium P1
Premium P2

The **Azure Active Directory Pricing**[1] page has detailed information on what is included in each of the editions.

- **Azure Active Directory Free** – Designed to introduce system administrators to Azure Active Directory. This version includes common features such as directory objects, user/group management, single sign-on, self-service password change, on-premises connect, and security/usage reports.
- **Azure Active Directory Basic** - Designed for task workers with cloud-first needs, this edition provides cloud centric application access and self-service identity management solutions. With the Basic edition of Azure Active Directory, you get productivity enhancing and cost reducing features like group-based access management, self-service password reset for cloud applications, and Azure Active Directory Application Proxy (to publish on-premises web applications using Azure Active Directory), all backed by an enterprise-level SLA of 99.9 percent uptime.
- **Azure Active Directory Premium P1** - Designed to empower organizations with more demanding identity and access management needs, Azure Active Directory Premium edition adds feature-rich enterprise-level identity management capabilities and enables hybrid users to seamlessly access on-premises and cloud capabilities. This edition includes everything you need for information worker and identity administrators in hybrid environments across application access, self-service identity and access management (IAM), and security in the cloud.
- **Azure Active Directory Premium P2** - Azure Active Directory Premium P2 includes every feature of all other Azure Active Directory editions enhanced with advanced identity protection and privileged identity management capabilities.

✓ Did you look through the pricing list to determine which features your organization needs?

[1] https://azure.microsoft.com/en-us/pricing/details/active-directory/?wt.mc_id=DXLEX_EDX_AZURE204X

Choosing Between Azure AD and Azure AD DS

One of the main differences between Azure AD and Azure AD DS is the way devices are registered and joined.

Azure AD Domain Services provides a managed AD domain in an Azure virtual network. You can join machines to this managed domain using traditional domain-join mechanisms. Azure AD also enables you to manage the identity of devices used by your organization and control access to corporate resources from these devices. Azure AD joined devices give you the following benefits:

- Single-sign-on (SSO) to applications secured by Azure AD
- Enterprise policy-compliant roaming of user settings across devices.
- Access to the Windows Store for Business using your corporate credentials.
- Windows Hello for Business
- Restricted access to apps and resources from devices compliant with corporate policy.

Aspect	Course Content	Azure AD Domain Services
Device controlled by	Azure AD	Azure AD Domain Services managed domain
Representation in the directory	Device objects in the Azure AD directory.	Computer objects in the AAD-DS managed domain.
Authentication	OAuth/OpenID Connect based protocols	Kerberos, NTLM protocols
Management	Mobile Device Management (MDM) software like Intune	Group Policy
Networking	Works over the internet	Requires machines to be on the same virtual network as the managed domain.
Great for ...	End-user mobile or desktop devices	Server virtual machines deployed in Azure

For more information, you can see:

Choose between Azure Active Directory join and Azure Active Directory Domain Services - **https://docs.microsoft.com/en-us/azure/active-directory-domain-services/active-directory-ds-compare-with-azure-ad-join**

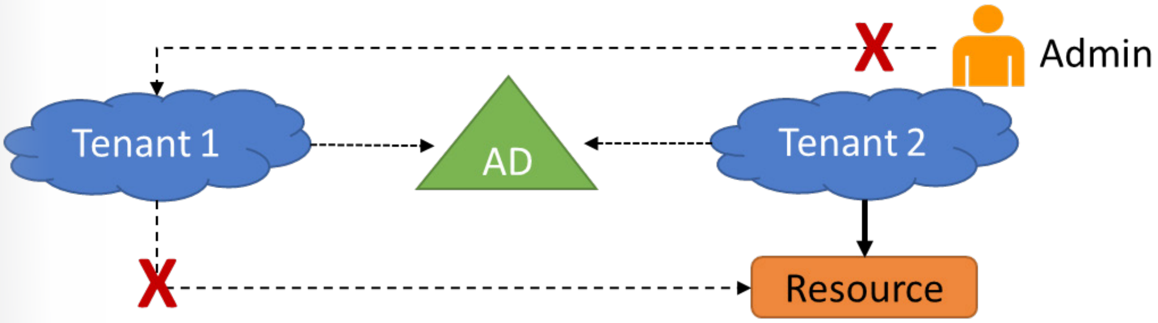
Azure AD Directories (Tenants)

A tenant is a dedicated instance of an Azure AD directory which is created whenever you sign up for a Microsoft cloud service, such as Office 365 or Azure. It is important to note; a tenant is not the same as a subscription. A subscription is typically tied to a credit card for billing, where a tenant is an instance of Active Directory. You can have multiple tenants in your organization, such as Contoso1.com and Contoso2.com .

Each tenant or Azure AD instance is separate and distinct from the other Azure AD directories in your organization. These different tenants could allow for different functions. For example: You could have a tenant for Office 365, another tenant a for testing environment, and then another tenant for Microsoft Intune. A tenant houses the users in a company and the information about them - their passwords, user profile data, permissions, and so on. It also contains groups, applications, and other information pertaining to an organization and its security.

Why would you need multiple tenants

Resource independence



- If you create or delete a resource in one tenant, it has no impact on any resource in another tenant, with the partial exception of external users.
- If you use one of your domain names with one tenant, it cannot be used with any other tenant.

Administrative independence

If a non-administrative user of tenant ‘Contoso’ creates a test tenant 'Test,' then:

- By default, the user who creates a tenant is added as an external user in that new tenant and assigned the global administrator role in that tenant.
- The administrators of tenant ‘Contoso’ have no direct administrative privileges to tenant 'Test,' unless an administrator of ‘Test’ specifically grants them these privileges.

Synchronization independence. You can configure each Azure AD tenant independently to get data synchronized from a single instance of either: The Azure AD Connect tool or the Forefront Identity Manager Azure Active Tenant Connector.

Built-in Roles

Azure AD provides many **built-in roles**[2] to cover the most common security scenarios. To understand how the roles work we will examine three roles that apply to all resource types:

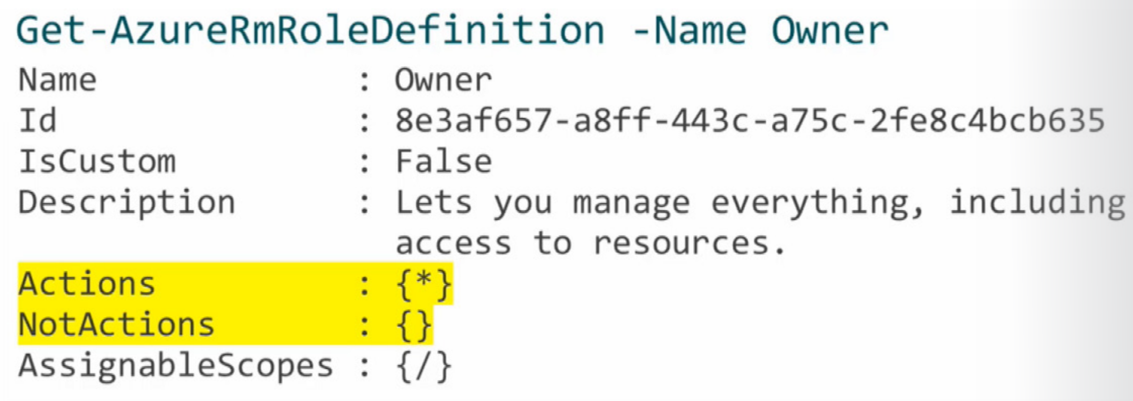
- **Owner** has full access to all resources including the right to delegate access to others.
- **Contributor** can create and manage all types of Azure resources but can’t grant access to others.
- **Reader** can view existing Azure resources.

Role definitions

Each role is a set of properties defined in a JSON file. This **role definition** includes Name, Id, and Description. It also includes the allowable permissions (Actions), denied permissions (NotActions), and scope (read access, etc.) for the role.

For the Owner role that means all (*) actions, no denied actions, and all (/) scopes. This information is available with the **Get-AzureRmRoleDefinition** cmdlet.

[2] https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-built-in-roles#roles-in-azure



```
Get-AzureRmRoleDefinition -Name Owner
Name             : Owner
Id               : 8e3af657-a8ff-443c-a75c-2fe8c4bcb635
IsCustom         : False
Description      : Lets you manage everything, including
                   access to resources.
Actions          : {*}
NotActions       : {}
AssignableScopes : {/}
```

✓ Take a minute to open the Azure Portal, open the Subscriptions or Resource Group blade, and click Access Control (IAM). Click **Add** and take a few minutes to review the built-in roles and see which role you would be most interested in using.

For more information, you can see:

Built-in roles in Azure - **https://docs.microsoft.com/en-us/azure/role-based-access-control/built-in-roles**

Create custom roles for Azure Role-Based Access Control - **https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-custom-roles**

Get-AzureRmRoleDefinition - **https://docs.microsoft.com/en-us/powershell/module/azurerm.resources/get-azurermroledefinition?view=azurermps-5.3.0**

Role Definitions

Actions and NotActions

The **Actions** and **NotActions** properties can be tailored to grant and deny the exact permissions you need. Review this table to see how Owner, Contributor, and Reader are defined.

Built-in Role	Action	NotActions
Owner (allow all actions)	*	
Contributor (allow all actions except writing or deleting role assignment)	*	Microsoft.Authorization/*/Delete, Microsoft.Authorization/*/Write, Microsoft.Authorization/elevate-Access/Action
Reader (allow all read actions)	*/read	

AssignableScopes

Defining the **Actions** and **NotActions** properties is not enough to fully implement a role. You must also properly scope your role.

The **AssignableScopes** property of the role specifies the scopes (subscriptions, resource groups, or resources) within which the custom role is available for assignment. You can make the custom role available for assignment in only the subscriptions or resource groups that require it, and not clutter user experience for the rest of the subscriptions or resource groups.

- /subscriptions/[subscription id]
- /subscriptions/[subscription id]/resourceGroups/[resource group name]
- /subscriptions/[subscription id]/resourceGroups/[resource group name]/[resource]

Example 1

Make a role available for assignment in two subscriptions.

“/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e”,

“/subscriptions/e91d47c4-76f3-4271-a796-21b4ecfe3624”

Example 2

Makes a role available for assignment only in the Network resource group.

“/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network”

✓ Take a minute to open the Azure Portal and use the Access Control blade to add a role and then assign it to a user. Can you see how for your organization which role assignments you would need?

For more information, you can see:

Custom roles access control - **https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-custom-roles#custom-roles-access-control**

Azure PowerShell and CLI

When you have large numbers of role assignments, you may prefer to use Azure PowerShell or the CLI.

```
#Role assignment properties
$roleName = “Contributor”
$assigneeName = josh@microsoft.com
$resourceGroupName = “contosoblue”
```

Azure PowerShell

```
New-AzureRmRoleAssignment -RoleDefinitionName $roleName -SignInName $assign-
eeName -ResourceGroupName $resourceGroupName
```

CLI

```
az role assignment create –role $roleName –assignee $assigneeName –re-
source-group $resourceGroupName
```

✓ If you have created a custom JSON role definition file you can use PowerShell or the CLI to create a new custom role definition. In the following examples the sysops.json file has the custom definition.

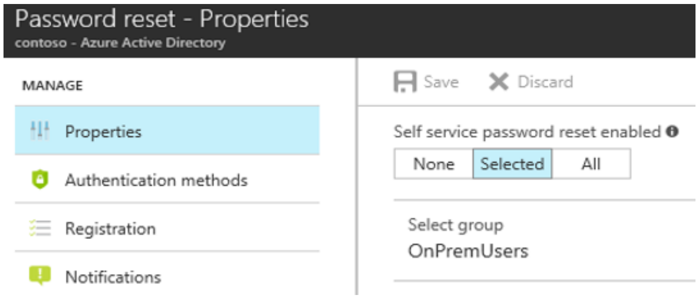
```
#PowerShell
New-AzureRmRoleDefinition -InputFile .\sysops.json
#CLI
az role definition create –role-definition “./sysops.json”
```

Configuring Self-Service Password Reset

Self-Service Password Reset

To configure self-service password reset, you first determine who will be enabled to use self-service password reset. From your existing Azure AD tenant, on the Azure Portal under **Azure Active Directory** select **Password reset.**

In the Password reset properties there are three options: **None**, **Selected**, and **All**.



Self service password reset enabled: None | [✓ Selected] | All

Select group: OnPremUsers

The **Selected** option is useful for creating specific groups who have self-service password reset enabled. The Azure documentation recommends creating a specific group for purposes of testing or proof of concept before deploying to a larger group within the Azure AD tenant. Once you are ready to deploy this functionality to all users with accounts in your AD Tenant, you can change the setting to **All**.

Important! Azure Administrator accounts will always be able to reset their passwords no matter what this option is set to.

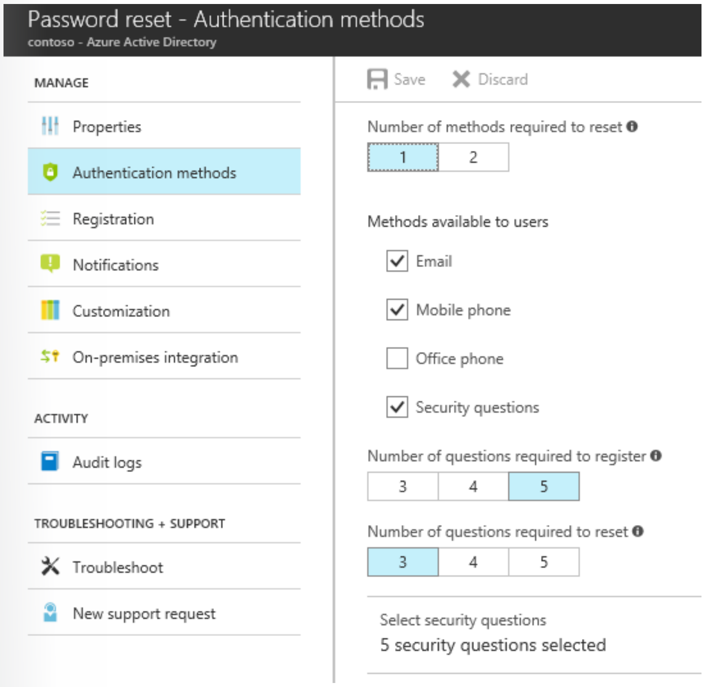
Authentication Methods for Password Reset

After enabling password reset for user and groups, you pick the number of authentication methods required to reset a password and the number of authentication methods available to users.

At least one authentication method is required to reset a password, but it is a good idea to have additional methods available. You can choose from email notification, a text or code sent to user’s mobile or office phone, or a set of security questions.

Regarding the security questions, these can be configured to require a certain number of questions to be registered for the users in your AD tenant. In addition, you must configure the number of correctly answered security question that are required for a successful password reset.

In the next demonstration, Corey walks through the process of self-service password reset.



Password reset - Authentication methods

contoso - Azure Active Directory

Number of methods required to reset: [✓ 1] 2

Methods available to users

☑ Email

☑ Mobile phone

□ Office phone

☑ Security questions

Number of questions required to register: 3 4 [✓ 5]

Number of questions required to reset: [✓ 3] 4 5

Select security questions

5 security questions selected

Password Writeback

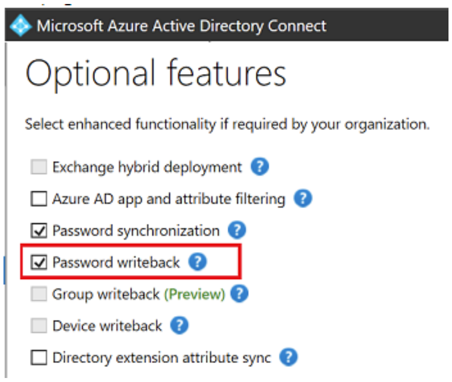
With password writeback, you can configure Azure Active Directory (Azure AD) to write passwords back to your on-premises Active Directory. Password writeback removes the need to set up and manage a complicated on-premises self-service password reset (SSPR) solution, and it provides a convenient cloud-based way for your users to reset their on-premises passwords wherever they are. Password writeback is a component of Azure Active Directory Connect that can be enabled and used by current subscribers of Premium Azure Active Directory editions. It's recommended that you use the auto-update feature of Azure AD Connect.

The following steps assume you have already configured Azure AD Connect in your environment by using the **Express**[3] or **Custom**[4] settings.

1. To configure and enable password writeback, sign in to your Azure AD Connect server and start the **Azure AD Connect** configuration wizard.
2. On the **Welcome** page, select **Configure**.
3. On the **Additional tasks** page, select **Customize synchronization options**, and then select **Next**.
4. On the **Connect to Azure AD** page, enter a global administrator credential, and then select **Next**.
5. On the **Connect directories** and **Domain/OU** filtering pages, select **Next**.
6. On the **Optional features** page, select the box next to **Password writeback** and select **Next**.

[3] https://docs.microsoft.com/en-us/azure/active-directory/connect/active-directory-aadconnect-get-started-express

[4] https://docs.microsoft.com/en-us/azure/active-directory/connect/active-directory-aadconnect-get-started-custom



Microsoft Azure Active Directory Connect

Optional features

Select enhanced functionality if required by your organization.

□ Exchange hybrid deployment
□ Azure AD app and attribute filtering
☑ Password synchronization
☑ Password writeback
□ Group writeback (Preview)
□ Device writeback
□ Directory extension attribute sync

1. On the **Ready to configure** page, select **Configure** and wait for the process to finish.
2. When you see the configuration finish, select **Exit**.

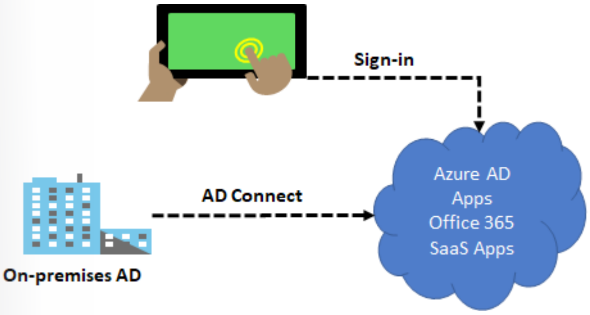
✓ Use the link below to read about the password writeback features. Which of the features are you most interested in?

For more information, you can see: Password writeback overview - **https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-sspr-writeback**

Implementing and Managing Hybrid Identities

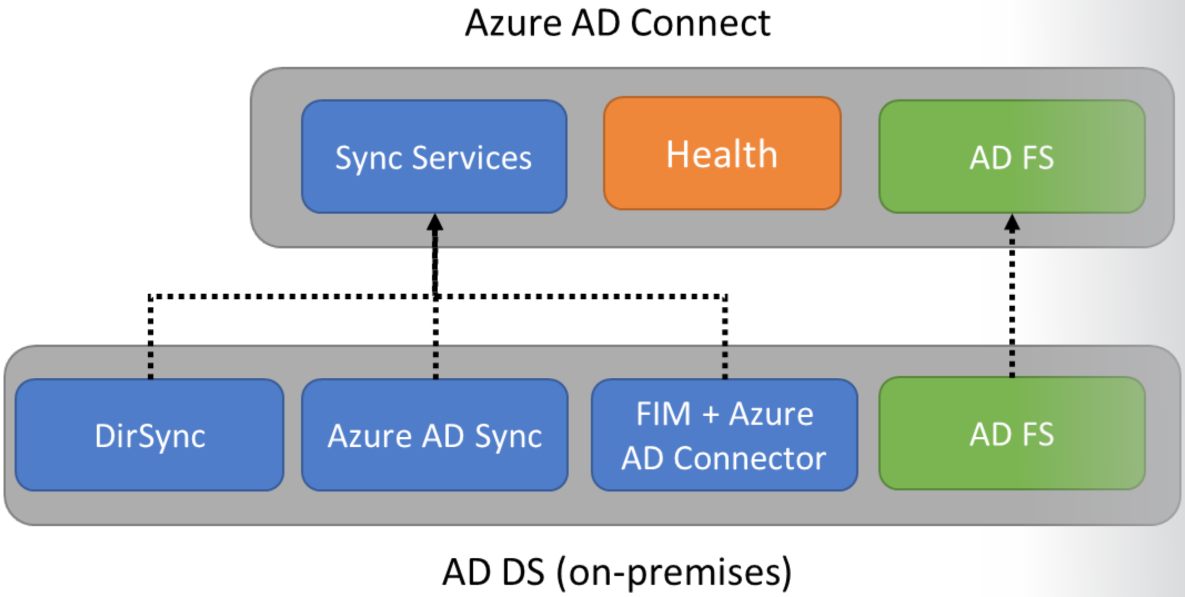
Azure AD Connect

Azure AD Connect will integrate your on-premises directories with Azure Active Directory. This allows you to provide a common identity for your users for Office 365, Azure, and SaaS applications integrated with Azure AD.



Azure AD Connect provides the following features:

- **Password hash synchronization**. A sign-in method that synchronizes a hash of a users on-premises AD password with Azure AD.
- **Pass-through authentication**. A sign-in method that allows users to use the same password on-premises and in the cloud, but doesn't require the additional infrastructure of a federated environment.
- **Federation integration**. Federation is an optional part of Azure AD Connect and can be used to configure a hybrid environment using an on-premises AD FS infrastructure. It also provides AD FS management capabilities such as certificate renewal and additional AD FS server deployments.
- **Synchronization**. Responsible for creating users, groups, and other objects. As well as, making sure identity information for your on-premises users and groups is matching the cloud. This synchronization also includes password hashes.
- **Health Monitoring**. Azure AD Connect Health can provide robust monitoring and provide a central location in the Azure portal to view this activity.



Sync Services. This component is responsible for creating users, groups, and other objects. It is also responsible for making sure identity information for your on-premises users and groups matches what's in the cloud.

Health Monitoring. Azure AD Connect Health can provide robust monitoring and provide a central location in the Azure portal to view this activity.

Active Directory Federation Services (AD FS). Federation is an optional part of Azure AD Connect and can be used to configure a hybrid environment using an on-premises AD FS infrastructure. Organizations can use this to address complex deployments, such as domain join SSO, enforcement of AD sign-in policy, and smart card or 3rd party MFA.

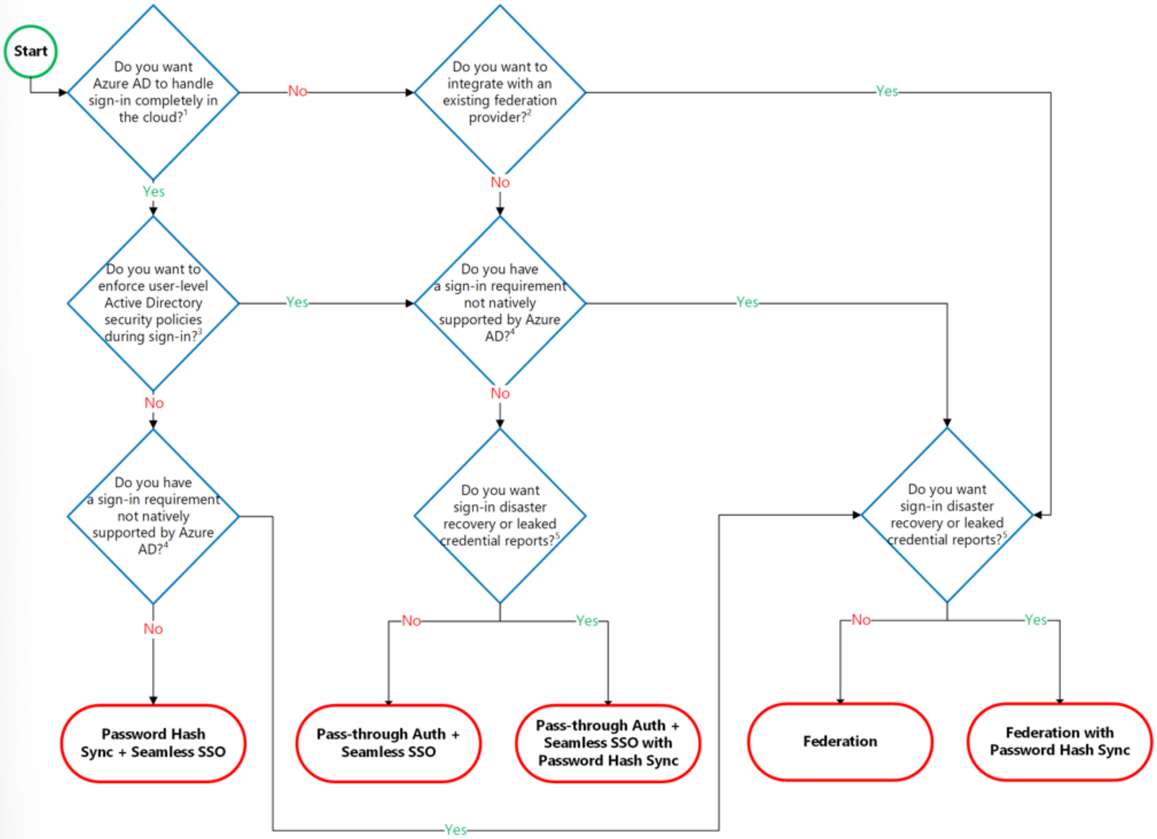
For more information, you can see:

Integrate your on-premises directories with Azure Active Directory - **https://docs.microsoft.com/en-us/azure/active-directory/connect/active-directory-aadconnect**

Authentication Options

Choosing an Azure AD Authentication method is important as it is one of the first important decisions when moving to the cloud as it will be the foundation of your cloud environment and is difficult to change at a later date.

You can choose cloud authentication which includes: Azure AD password hash synchronization and Azure AD Pass-through Authentication. You can also choose federated authentication where Azure AD hands off the authentication process to a separate trusted authentication system, such as on-premises Active Directory Federation Services (AD FS), to validate the user's password.


Start
Do you want Azure AD to handle sign-in completely in the cloud?[1]
No
Do you want to integrate with an existing federation provider?[2]
Yes
Yes
No
Do you want to enforce user-level Active Directory security policies during sign-in?[3]
Yes
Do you have a sign-in requirement not natively supported by Azure AD?[4]
Yes
No
No
Do you have a sign-in requirement not natively supported by Azure AD?[4]
Do you want sign-in disaster recovery or leaked credential reports?[5]
Do you want sign-in disaster recovery or leaked credential reports?[5]
No
No
Yes
No
Yes
Password Hash Sync + Seamless SSO
Pass-through Auth + Seamless SSO
Pass-through Auth + Seamless SSO with Password Hash Sync
Federation
Federation with Password Hash Sync
Yes

Summary

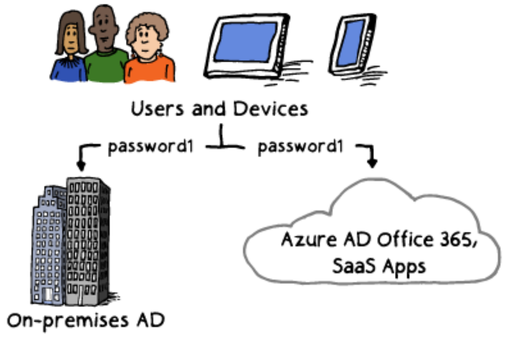
1. Do you need on-premises Active Directory integration? If the answer is No, then you would use Cloud-Only authentication.
2. If you do need on-premises Active Directory integration, then do you need to use cloud authentication, password protection, and your authentication requirements are natively supported by Azure AD? If the answer is Yes, Then you would use **Password Hash Sync** + Seamless SSO.
3. If you do need on-premises Active Directory integration, but you do not need to use cloud authentication, password protection, and your authentication requirements are natively supported by Azure AD, then you would use **Pass-through Authentication** Seamless SSO.
4. if you need on-premises Active Directory integration, have an existing federation provider and your authentication requirements are NOT natively supported by Azure AD, then you would use **Federation** authentication.

For more information, you can see:

Video - How to choose the right authentication option in Azure Active Directory - **https://www.youtube.com/watch?v=YtW2cmVqSEw**

Password Synchronization

The probability that you're blocked from getting your work done due to a forgotten password is related to the number of different passwords you need to remember. The more passwords you need to remember, the higher the probability to forget one. Questions and calls about password resets and other password-related issues demand the most helpdesk resources.



Password hash synchronization is a feature used to synchronize user passwords from an on-premises Active Directory instance to a cloud-based Azure AD instance. Use this feature to sign in to Azure AD services like Office 365, Microsoft Intune, CRM Online, and Azure Active Directory Domain Services (Azure AD DS). You sign in to the service by using the same password you use to sign in to your on-premises Active Directory instance. By reducing the number of passwords, your users need to maintain to just one. Password synchronization helps you to:

- Improve the productivity of your users.
- Reduce your helpdesk costs.

How does this work?

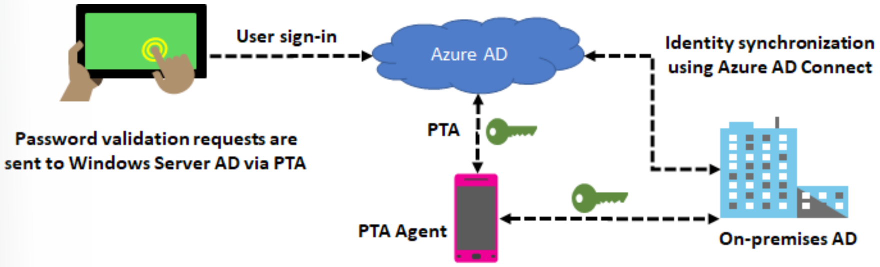
In the background, the password synchronization component takes the user's password hash from on-premises Active Directory, encrypts it, and passes it as a string to Azure. Azure decrypts the encrypted hash and stores the password hash as a user attribute in Azure AD.

When the user signs in to an Azure service, the sign-in challenge dialog box generates a hash of the user's password and passes that hash back to Azure. Azure then compares the hash with the one in that user's account. If the two hashes match, then the two passwords must also match and the user receives access to the resource. The dialog box provides the facility to save the credentials so that the next time the user accesses the Azure resource, the user will not be prompted.

✓ It is important to understand that this is **same sign-in**, not single sign-on. The user still authenticates against two separate directory services, albeit with the same user name and password. This solution provides a simple alternative to an AD FS implementation.

Pass-through Authentication

Azure AD Pass-through Authentication (PTA) is an alternative to Azure AD Password Hash Synchronization, and provides the same benefit of cloud authentication to organizations. PTA allows users to sign in to both on-premises and cloud-based applications using the same user account and passwords. When users sign-in using Azure AD, Pass-through authentication validates the users' passwords directly against an organizations on-premise Active Directory.


User sign-in
Azure AD
Identity synchronization using Azure AD Connect
Password validation requests are sent to Windows Server AD via PTA
PTA
PTA Agent
On-premises AD

Feature benefits

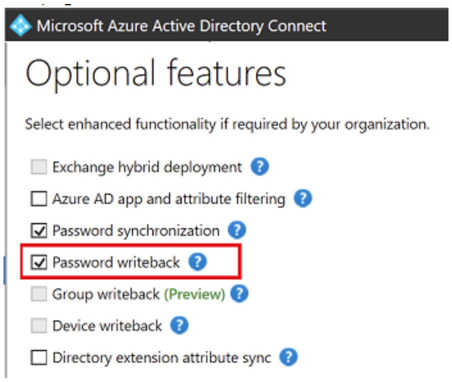
- Supports user sign-in into all web browser-based applications and into Microsoft Office client applications that use modern authentication.
- Sign-in usernames can be either the on-premises default username (userPrincipalName) or another attribute configured in Azure AD Connect (known as Alternate ID).
- Works seamlessly with conditional access features such as Multi-Factor Authentication to help secure your users.
- Integrated with cloud-based self-service password management, including password writeback to on-premises Active Directory and password protection by banning commonly used passwords.
- Multi-forest environments are supported if there are forest trusts between your AD forests and if name suffix routing is correctly configured.
- PTA is a free feature, and you don't need any paid editions of Azure AD to use it.
- PTA can be enabled via Azure AD Connect.
- PTA uses a lightweight on-premises agent that listens for and responds to password validation requests.
- Installing multiple agents provides high availability of sign-in requests.
- PTA protects your on-premises accounts against brute force password attacks in the cloud.

✓ This feature can be configured without using a federation service so that any organization, regardless of size, can implement a hybrid identity solution. Pass-through authentication is not only for user sign-in but allows an organization to use other Azure AD features, such as password management, role-based access control, published applications, and conditional access policies.

Password Writeback

Having a cloud-based password reset utility is great but most companies still have an on-premises directory where their users exist. How does Microsoft support keeping traditional on-premises Active Directory (AD) in sync with password changes in the cloud?

Password writeback is a feature enabled with Azure AD Connect that allows password changes in the cloud to be written back to an existing on-premises directory in real time.



Microsoft Azure Active Directory Connect

Optional features

Select enhanced functionality if required by your organization.

- □ Exchange hybrid deployment
- □ Azure AD app and attribute filtering
- ☑ Password synchronization
- ☑ Password writeback
- □ Group writeback (Preview)
- □ Device writeback
- □ Directory extension attribute sync

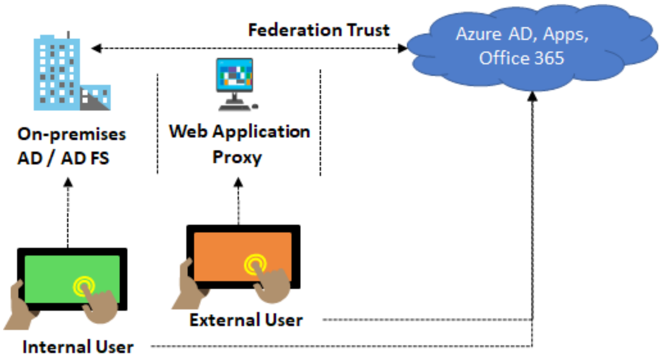
Password writeback provides:

- **Enforcement of on-premises Active Directory password policies**. When a user resets their password, it is checked to ensure it meets your on-premises Active Directory policy before committing it to that directory. This review includes checking the history, complexity, age, password filters, and any other password restrictions that you have defined in local Active Directory.
- **Zero-delay feedback**. Password writeback is a synchronous operation. Your users are notified immediately if their password did not meet the policy or could not be reset or changed for any reason.
- **Supports password changes from the access panel and Office 365**. When federated or password hash synchronized users come to change their expired or non-expired passwords, those passwords are written back to your local Active Directory environment.
- **Supports password writeback when an admin resets them from the Azure portal**. Whenever an admin resets a user's password in the Azure portal, if that user is federated or password hash synchronized, the password is written back to on-premises. This functionality is currently not supported in the Office admin portal.
- **Doesn't require any inbound firewall rules**. Password writeback uses an Azure Service Bus relay as an underlying communication channel. All communication is outbound over port 443.

✓ To use SSPR you must have already configured Azure AD Connect in your environment.

Federation with Azure AD

Federation is a collection of domains that have established trust. The level of trust may vary, but typically includes authentication and almost always includes authorization. A typical federation might include a number of organizations that have established trust for shared access to a set of resources.

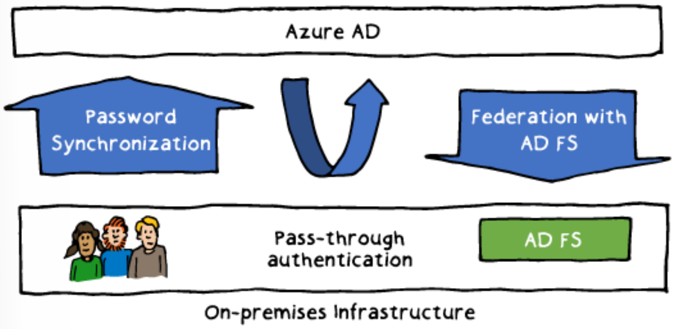


You can federate your on-premises environment with Azure AD and use this federation for authentication and authorization. This sign-in method ensures that all user authentication occurs on-premises. This method allows administrators to implement more rigorous levels of access control.

✓ If you decide to use Federation with Active Directory Federation Services (AD FS), you can optionally set up password hash synchronization as a backup in case your AD FS infrastructure fails.

Sign-On Methods

AD Connect provides several sign-on methods: **Password Synchronization**, **Pass-through authentication**, and **Federation with AD FS**. These methods are used to synchronize user accounts and, optionally, passwords from an on-premises Active Directory instance to a cloud-based Azure AD instance. Synchronization helps you to improve the productivity of your users and reduce your helpdesk costs.



Password Synchronization. This option can be used to synchronize an encrypted version of the password hash for user accounts. This ensures a user signing on to Azure uses the same password as the on-premises domain. The is sometimes referred to password hash synchronization.

For more information, you can see:

How password synchronization works - **https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-password-hash-synchronization#how-password-hash-synchronization-works**

Pass-through authentication (PTA). With this option the username and password are authenticated by the on-premises domain controllers. This is one of the newest authentication methods. Having a highly-available internet connection is highly recommended.

For more information, you can see:

User sign-in with Azure Active Directory Pass-through Authentication - **https://docs.microsoft.com/en-us/azure/active-directory/connect/active-directory-aadconnect-pass-through-authentication**

Federation with AD FS. AD FS is the Microsoft implementation of an identity federation solution that uses claims-based authentication. When AD FS has been configured, AD FS performs the validation through the on-premises AD DS environment. Azure AD Connect, discussed later in this module, can automate much of the AD FS configuration when integrating with Azure.

✓ If you are interested in the details of these authentication methods, check out the following deep dive video:

Deep-dive: Azure Active Directory Authentication and Single-Sign-On (video) - **https://channel9.msdn.com/events/Ignite/Microsoft-Ignite-Orlando-2017/BRK3015?term=Azure%20AD%20Pass-through%20Authentication%20and%20Seamless%20Single%20Sign-on**

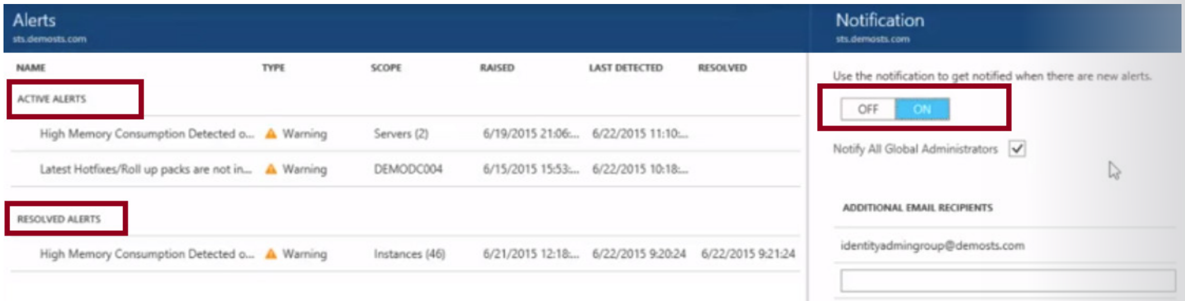
Azure AD Connect Health

When you integrate your on-premises directories with Azure AD, your users are more productive because there's a common identity to access both cloud and on-premises resources. However, this integration creates the challenge of ensuring that this environment is healthy so that users can reliably access resources both on premises and in the cloud from any device.

Azure AD Connect Health helps you:

- Monitor and gain insights into AD FS servers, Azure AD Connect, and AD domain controllers.
- Monitor and gain insights into the synchronizations that occur between your on-premises AD DS and Azure AD.
- Monitor and gain insights into your on-premises identity infrastructure that is used to access Office 365 or other Azure AD applications

With Azure AD Connect the key data you need is easily accessible. You can view and act on alerts, setup email notifications for critical alerts, and view performance data.



Alerts
sts.demosts.com

NAME	TYPE	SCOPE	RAISED	LAST DETECTED	RESOLVED
ACTIVE ALERTS					
High Memory Consumption Detected o...	⚠ Warning	Servers (2)	6/19/2015 21:06:...	6/22/2015 11:10:...	
Latest Hotfixes/Roll up packs are not in...	⚠ Warning	DEMODC004	6/15/2015 15:53:...	6/22/2015 10:18:...	
RESOLVED ALERTS					
High Memory Consumption Detected o...	⚠ Warning	Instances (46)	6/21/2015 12:18:...	6/22/2015 9:20:24	6/22/2015 9:21:24

Notification
sts.demosts.com

Use the notification to get notified when there are new alerts.

OFF [✓ ON]

Notify All Global Administrators ☑

ADDITIONAL EMAIL RECIPIENTS

identityadmingroup@demosts.com

✓ Using AD Connect Health works by installing an agent on each of your on-premises sync servers.

For more information, you can see:

Monitor your on-premises identity infrastructure and synchronization services in the cloud - **https://docs.microsoft.com/en-us/azure/active-directory/connect-health/active-directory-aadconnect-health**

Device Management

Azure Active Directory (Azure AD) enables single sign-on to devices, apps, and services from anywhere. The proliferation of devices - including Bring Your Own Device (BYOD) – empowers end users to be productive wherever and whenever. But, IT administrators must ensure corporate assets are protected and that devices meet standards for security and compliance.

To get a device under the control of Azure AD, you have two options:

- **Registering** a device to Azure AD enables you to manage a device’s identity. When a device is registered, Azure AD device registration provides the device with an identity that is used to authenticate the device when a user signs-in to Azure AD. You can use the identity to enable or disable a device.
- **Joining** a device is an extension to registering a device. This means, it provides you with all the benefits of registering a device and in addition to this, it also changes the local state of a device. Changing the local state enables your users to sign-in to a device using an organizational work or school account instead of a personal account.

✓ Registration combined with a mobile device management (MDM) solution such as Microsoft Intune, provides additional device attributes in Azure AD. This allows you to create conditional access rules that enforce access from devices to meet your standards for security and compliance.

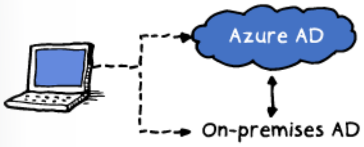
For more information, you can see:

Introduction to device management - **https://docs.microsoft.com/en-us/azure/active-directory/device-management-introduction**

Azure registered devices - **https://docs.microsoft.com/en-us/azure/active-directory/device-management-introduction#azure-ad-registered-devices**

Azure Joined Devices

If your environment has an on-premises AD footprint and you also want to benefit from the capabilities provided by Azure Active Directory, you can implement hybrid Azure AD joined devices. These are devices that are joined both to your on-premises Active Directory and your Azure Active Directory.



Joining devices to both directories allows:

- IT departments to manage work-owned devices from a central location.
- Users to sign in to their devices with their Active Directory work or school accounts.

Here is a comparison of Registered, AD Joined, and Hybrid AD Joined devices.

	Registered Devices	Azure AD Joined Devices	Hybrid AD Joined Devices
Device Type	Personal	Organization owned	Organization owned
Registration	Manual	Manual	Automatic
Operating System	Windows 10	Windows 10	Windows 7, 8, and 10

✓ Are you understanding the different types of joined devices? Which do you think your organization needs?

For more information, you can see:

Hybrid Azure AD joined devices - **https://docs.microsoft.com/en-us/azure/active-directory/device-management-introduction#hybrid-azure-ad-joined-devices**

Online Lab - Implementing User-Assigned Managed Identities for Azure Resources

Lab Steps

Online Lab: Implementing User-Assigned Managed Identities for Azure Resources

NOTE: For the most recent version of this online lab, see: **https://github.com/MicrosoftLearning/AZ-300-MicrosoftAzureArchitectTechnologies**

Scenario

Adatum Corporation wants to use manage identities to authenticate applications running in Azure VMs

Objectives

After completing this lab, you will be able to:

- Create and configure user-assigned managed identities
- Validate functionality of user-assigned managed identities

Lab Setup

Estimated Time: 30 minutes

User Name: **Student**

Password: **Pa55w.rd**

Exercise 1: Creating and configuring a user-assigned managed identity.

The main tasks for this exercise are as follows:

1. Deploy an Azure VM running Windows Server 2016 Datacenter
2. Create a user-assigned managed identity.
3. Assign the user-assigned managed identity to the Azure VM.
4. Grant RBAC-based permissions to the user-assigned managed identity.

Task 1: Deploy an Azure VM running Windows Server 2016 Datacenter

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. In the Azure portal, in the Microsoft Edge window, start a **Bash** session within the **Cloud Shell**.

3. If you are presented with the **You have no storage mounted** message, configure storage using the following settings:

    - Subscription: the name of the target Azure subscription
    - Cloud Shell region: the name of the Azure region that is available in your subscription and which is closest to the lab location
    - Resource group: **az3000500-LabRG**
    - Storage account: a name of a new storage account
    - File share: a name of a new file share

4. From the Cloud Shell pane, create a resource group by running (replace the `<Azure region>` placeholder with the name of the Azure region that is available in your subscription and which is closest to the lab location)

    ```
    az group create --resource-group az3000501-LabRG --location <Azure region>
    ```

5. From the Cloud Shell pane, upload the Azure Resource Manager template **\allfiles\AZ-300T01\Module_05\azuredeploy05.json** into the home directory.
6. From the Cloud Shell pane, upload the parameter file **\allfiles\AZ-300T01\Module_05\azuredeploy05.parameters.json** into the home directory.
7. From the Cloud Shell pane, deploy the two Azure VMs hosting Windows Server 2016 Datacenter into the first virtual network by running:

    ```
    az group deployment create --resource-group az3000501-LabRG --template-file
    azuredeploy05.json --parameters @azuredeploy05.parameters.json
    ```

8. **Note**: Wait for the deployment to complete. This might take about 5 minutes.

Task 2: Create a user-assigned managed identity and assign it to the Azure VM.

1. From the Cloud Shell pane, run the following to create a user-assigned managed identity:

    ```
    az identity create --resource-group az3000501-LabRG --name az3000501-mi
    ```

2. From the Cloud Shell pane, run the following to assign the user-assigned managed identity to the Azure VM:

    ```
    az vm identity assign --resource-group az3000501-LabRG --name az3000501-vm
    --identities az3000501-mi
    ```

Task 3: Configure RBAC referencing the user-assigned managed identity.

1. From the Cloud Shell pane, run the following to create a resource group (replace the `<Azure region>` placeholder with the name of the Azure region into which you deployed the Azure VM in this exercise):

```
az group create --resource-group az3000502-LabRG --location <Azure region>
```

2. In the Azure portal, navigate to the **az3000502-LabRG - Access control (IAM)** blade.
3. From the **az3000502-LabRG - Access control (IAM)** blade, assign the Owner role to the newly created user-assigned managed identity.

Result: After you completed this exercise, you have created and configured a user-assigned managed identity.

Exercise 2: Validating functionality of user-assigned managed identities

The main tasks for this exercise are as follows:

1. Configure an Azure VM for authenticating via user-assigned managed identity.
2. Validate functionality of user-assigned managed identity from the Azure VM.

Task 1: Configure an Azure VM for authenticating via user-assigned managed identity.

1. In the Azure portal, navigate to the **az3000501-vm** blade.
2. Connect to the Azure VM by using Remote Desktop and authenticate by providing the following credentials:
   - Username: **Student**
   - Password: **Pa55w.rd1234**
3. Once you establish a Remote Desktop session, you will be presented with an **Administrator: C:\Windows\system32\cmd.exe** window. To start a PowerShell session, at the command prompt, type `PowerShell` and press Enter.
4. From the PowerShell prompt, run the following to install the latest version of the PowerShellGet module (press Enter if prompted for confirmation):

   ```
   Install-Module -Name PowerShellGet -Force
   ```

5. From the PowerShell prompt, run the following to install the latest version of the Az module (press Enter if prompted for confirmation):

   ```
   Install-Module -Name Az -AllowClobber
   ```

6. Exit the current PowerShell session by typing `exit` and pressing Enter and then start it again by typing at the command prompt `PowerShell` and pressing Enter.
7. From the PowerShell prompt, run the following to install the the pre-release version of the PowerShellGet module:

   ```
   Install-Module -Name PowerShellGet -AllowPrerelease
   ```

8. From the PowerShell prompt, run the following to install the the pre-release version of the AzureRM.ManagedServiceIdentity module:

```
Install-Module -Name Az.ManagedServiceIdentity -AllowPrerelease
```

Task 2: Validate functionality of user-assigned managed identity from the Azure VM.

1. From the PowerShell prompt, run the following to sign-in as the user-assigned managed identity:

   ```
   Add-AzAccount -Identity
   ```

2. From the PowerShell prompt, run the following to attempt to retrieve the currently used managed identity:

   ```
   (Get-AzVM -ResourceGroupName az3000501-LabRG -Name az3000501-vm).Identity
   ```

3. Note the error message. As the message states, the current security context does not grant sufficent authorization to the target resource. To resolve this issue, switch to the Azure portal, navigate to the **az3000501-LabRG - Access control (IAM)** blade.
4. From the **az3000501-LabRG - Access control (IAM)** blade, assign the Reader role to the user-assigned managed identity **az3000501-mi**.
5. Switch back to the Remote Desktop session, and, from the PowerShell prompt, run the following to attempt to retrieve the currently used managed identity:

   ```
   (Get-AzVM -ResourceGroupName az3000501-LabRG -Name az3000501-vm).Identity
   ```

6. From the PowerShell prompt, run the following to store location in a variable:

   ```
   $location = (Get-AzResourceGroup -Name az3000502-LabRG).Location
   ```

7. From the PowerShell prompt, run the following to create a public IP address resource:

   ```
   New-AzPublicIpAddress -Name az3000502-pip -ResourceGroupName az3000502-
   LabRG -AllocationMethod Dynamic -Location $location
   ```

8. Verify that the command completed successfully.

Result: After you completed this exercise, you have validated the functionality of the user-defined managed identity.

Exercise 3: Remove lab resources

Task 1: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

   ```
   az group list --query "[?starts_with(name,'az30005')]".name --output tsv
   ```

3. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

Task 2: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

   ```
   az group list --query "[?starts_with(name,'az30005')]".name --output tsv |
   xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
   ```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

Result: In this exercise, you removed the resources used in this lab.

Review Questions

Module 5 Review Questions

Active Directory Domain Services (AD DS)

You manage the Azure subscription for an organization. You migrate an on-premises service to Azure. The service requires Kerberos for authentication.

You need to ensure that users can access and authenticate with the service.

What authentication options are available? Which option should you use and why?

Suggested Answer ↓

You can deploy and manage AD DS in Azure virtual machines it's recommended you use Azure AD instead, unless you are targeting IaaS workloads that depend on AD DS specifically.

Communication Protocols. Because Azure AD is HTTP/HTTPS based, it does not use Kerberos authentication. Instead, it uses HTTP and HTTPS protocols such as SAML, WS-Federation, and OpenID Connect for authentication (and OAuth for authorization).

Self-Service Password Reset

Users frequently become locked out of their accounts and may require password resets.

You decide to implement Azure Self-Service Password Reset.

What must you do to implement this service? Which authentication methods are supported?

Suggested Answer ↓

After enabling password reset for user and groups, you pick the number of authentication methods required to reset a password and the number of authentication methods available to users.

At least one authentication method is required to reset a password, but it is a good idea to have additional methods available. You can choose from email notification, a text or code sent to user's mobile or office phone, or a set of security questions.

Regarding the security questions, these can be configured to require a certain number of questions to be registered for the users in your AD tenant (3-5). In addition, you must configure the number of correctly answered security question that are required for a successful password reset (3-5).

Azure AD Connect Health

An organization uses a hybrid model for connecting to resources. The organization implement Azure AD in parallel with an on-premises Active Directory Domain Services (AD DS) environment. You configure Azure AD SSO.

You must implement Azure AD Connect Health to monitor the environment.

What are the benefits of using Azure AD Connect Health. How can you designate the servers to monitor?

Suggested Answer ↓

When you integrate your on-premises directories with Azure AD, your users are more productive because there is a common identity to access both cloud and on-premises resources. However, this integration creates the challenge of ensuring that this environment is healthy so that users can reliably access resources both on premises and in the cloud from any device.

Azure AD Connect Health helps you with the following activities:

- Monitor and gain insights into AD FS servers, Azure AD Connect, and AD domain controllers.
- Monitor and gain insights into the synchronizations that occur between your on-premises AD DS and Azure AD.
- Monitor and gain insights into your on-premises identity infrastructure that is used to access Office 365 or other Azure AD applications
- With Azure AD Connect the key data you need is easily accessible. You can view and act on alerts, setup email notifications for critical alerts, and view performance data.