Microsoft

Get started guide for Azure IT operators

Authors and Contributors

The following resources contributed to this version of this guide:

Author

Neil Peterson | Microsoft – Senior Content Developer

Contributors and Reference Content

Robin Shahan | Microsoft – Senior Content Developer

Michael Collier | Microsoft – Senior SDE

Microsoft Azure Essentials: Fundamentals of Azure, Second Edition

Summary

The purpose of this document is to provide information that will help quickly get started using Azure services. The target audience is those in an IT operator role.

Contents

This guide introduces core concepts related to the deployment and management of a Microsoft Azure infrastructure. If you are new to cloud computing, or Azure itself, this guide will help get you quickly started with concepts, deployment, and management details. Many sections of this guide discuss an operation such as deploying a virtual machine, and then provide a link for in-depth technical detail.

Introduction to cloud computing and Microsoft Azure

Cloud computing overview

Cloud computing provides a modern alternative to the traditional on-premises datacenter. Public cloud vendors provide and manage all computing infrastructure and the underlying management software. These vendors provide a wide variety of cloud services. A cloud service in this case might be a virtual machine, a web server, or cloud-hosted database engine. As a cloud provider customer, you lease these cloud services on an as-needed basis. In doing so, you convert the capital expense of hardware maintenance into an operational expense. A cloud service also provides these benefits:

- Rapid deployment of large compute environments
- Rapid deallocation of systems that are no longer required
- Easy deployment of traditionally complex systems like load balancers
- Ability to provide flexible compute capacity or scale when needed
- More cost-effective computing environments
- Access from anywhere with a web-based portal or programmatic automation
- Cloud-based services to meet most compute and application needs

With on-premises infrastructure, you have complete control over the hardware and software that is deployed. Historically, this has led to hardware procurement decisions that focus on scaling up. An example is purchasing a server with more cores to satisfy peak performance needs. Unfortunately, this infrastructure might be underutilized outside a demand window. With Azure, you can deploy only the infrastructure that you need, and adjust this up or down at any time. This leads to a focus on scaling out through the deployment of additional compute nodes to satisfy a performance need. Although this has consequences for the design of an appropriate software architecture, there is now ample proof that scaling out the commodity of cloud services is more cost-effective than scaling up through expensive hardware.

Microsoft has deployed many Azure datacenters around the globe, with more planned. Additionally, Microsoft is increasing sovereign clouds in regions like China and Germany. Only the largest global enterprises can deploy datacenters in this manner, so using Azure makes it easy for enterprises of any size to deploy their services close to their customers.

For small businesses, Azure allows for a low-cost entry point, with the ability to scale rapidly as demand for compute increases. This prevents a large up-front capital investment in infrastructure, and it provides the flexibility to architect and re-architect systems as needed. The use of cloud computing fits well with the scale-fast and fail-fast model of startup growth.

For more information on the available Azure regions, see Azure regions.

Types of cloud computing

Cloud computing is usually classified into three categories: SaaS, PaaS, and IaaS.

SaaS: Software as a service

SaaS is software that is centrally hosted and managed. It's usually based on a multitenant architecture—a single version of the application is used for all customers. It can be scaled out to multiple instances to ensure the best performance in all locations. SaaS software typically is licensed through a monthly or annual subscription.

Microsoft Office 365 is a prototypical model of a SaaS offering. Subscribers pay a monthly or annual subscription fee, and they get Microsoft Exchange as a service (online and/or desktop Microsoft Outlook), storage as a service (Microsoft OneDrive), and the rest of the Microsoft Office suite (online, the desktop version, or both). Subscribers always get the most recent version. So you can have an Exchange server without having to purchase a server and install and support Exchange—the Exchange server is managed for you. Compared to installing and upgrading Office every year, this is much less expensive and requires much less effort to keep updated.

PaaS: Platform as a service

With PaaS, you deploy your application into an application-hosting environment that the cloud service vendor provides. The developer provides the application, and the PaaS vendor provides the ability to deploy and run it. This frees developers from infrastructure management so they can focus on development.

Azure provides several PaaS compute offerings, including the Web Apps feature of Azure App Service and Azure Cloud Services (web and worker roles). In either case, developers have multiple ways to deploy their application without knowing anything about the nuts and bolts that support it. Developers don't have to create virtual machines (VMs), use Remote Desktop Protocol (RDP) to sign in to each one, or install the application. They just hit a button (or close to it), and the tools provided by Microsoft provision the VMs and then deploy and install the application on them.

IaaS: Infrastructure as a service

An IaaS cloud vendor runs and manages all physical compute resources and the required software to enable computer virtualization. A customer of this service deploys virtual machines in these hosted datacenters. Although the virtual machines are located in an offsite datacenter, the IaaS consumer has control over the configuration and management of them.

Azure includes several IaaS solutions, including Azure Virtual Machines, virtual machine scale sets, and related networking infrastructure. Azure Virtual Machines is a popular choice for initially migrating services to Azure because it enables a "lift and shift" migration model. You can configure a VM like the infrastructure currently running your services in your datacenter, and then migrate your software to the new VM. You might need to make configuration updates, such as URLs to other services or storage, but you can migrate many applications in this way.

Virtual machine scale sets are built on top of Azure Virtual Machines and provide an easy way to deploy clusters of identical VMs. Virtual machine scale sets also support autoscaling so that new VMs can be deployed automatically when required. This makes virtual machine scale sets an ideal platform to host higher-level microservice compute clusters, such as Azure Service Fabric and Azure Container Service.

Azure services

Azure offers many services in its cloud computing platform. These services include the following.

Compute services

Services for hosting and running application workload:

- Azure Virtual Machines—both Linux and Windows
- App Services (Web Apps, Mobile Apps, Logic Apps, API Apps, and Function Apps)
- Azure Batch (for large-scale parallel and batch compute jobs)
- Azure RemoteApp
- Azure Service Fabric
- Azure Container Service

Data services

Services for storing and managing data:

- Azure Storage (comprises the Azure Blob, Queue, Table, and File services)
- Azure SQL Database
- Azure DocumentDB
- Microsoft Azure StorSimple
- Azure Redis Cache

Application services

Services for building and operating applications:

- Azure Active Directory (Azure AD)
- Azure Service Bus for connecting distributed systems
- Azure HDInsight for processing big data
- Azure Scheduler
- Azure Media Services

Network services

Services for networking both within Azure and between Azure and on-premises datacenters:

- Azure Virtual Network
- Azure ExpressRoute
- Azure-provided DNS
- Azure Traffic Manager
- Azure Content Delivery Network

For detailed documentation on each of these services, as well as other Azure services, see Azure service documentation.

Azure key concepts

Datacenters and regions

Azure is a global cloud platform that is generally available in many regions around the world. When you provision a service, application, or VM in Azure, you are asked to select a region. The selected region represents a specific datacenter where your application runs. For more information, see Azure regions.

One of the benefits of using Azure is that you can deploy your applications into a variety of datacenters around the globe. The region you choose can affect the performance of your application. It's optimal to choose a region that is closer to most your customers, to reduce latency in network requests. You might also select a region to meet the legal requirements for distributing your app in certain countries.

Azure portal

The Azure portal is a web-based application that can be used to create, manage, and remove Azure resources and services. The Azure portal is located at https://portal.azure.com. It includes a customizable dashboard and tooling for managing Azure resources. It also provides billing and subscription information. For more information, see Microsoft Azure portal overview.

Resources

Azure resources are individual compute, networking, data, or app hosting services that have been deployed into an Azure subscription. Some common resources are a virtual machines, storage accounts, or SQL databases. Azure services often consist of several related Azure resources. For instance, an Azure virtual machine might include a VM, storage account, network adapter, and public IP address. All of these are individual resources. Each resource can be created, managed, and deleted individually or as a group. Azure resources are covered in more detail later in this guide.

Resource groups

An Azure resource group is a container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only resources that you want to manage as a group. Azure resource groups are covered in more detail later in this guide.

Resource Manager templates

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines one or more resources to deploy to a resource group. It also defines the dependencies between deployed resources. Resource Manager templates are covered in more detail later in this guide.

Automation

In addition to creating, managing, and deleting resources by using the Azure portal, you can automate these activities by using PowerShell or the Azure command-line interface (CLI).

Azure PowerShell

Azure PowerShell is a set of modules that provide cmdlets to manage Azure. You can use the cmdlets to create, manage, and remove Azure services. In most cases, you can use the cmdlets for the same tasks that you perform in the Azure portal. The cmdlets can help you can achieve consistent, repeatable, and hands-off deployments. For more information, see How to install and configure Azure PowerShell.

Azure command-line interface

The Azure command-line interface is a tool that you can use to create, manage, and remove Azure resources from the command line. The Azure CLI is available for Linux, Mac OS X, and Windows. For more information and technical details, see Install the Azure CLI.

REST APIs

Azure is built on a set of REST APIs that support the Azure portal UI. Most of these REST APIs are also supported to let you programmatically provision and manage your Azure resources and apps from any Internet-enabled device. For more information, see the Azure REST SDK Reference.

Getting started with Azure subscriptions

A subscription is a logical grouping of Azure services that is linked to an Azure account. A singe Azure account can contain multiple subscriptions. Billing for Azure services is done on a per-subscription basis. Azure subscriptions have an account administrator, who has full control over the subscription, and a service administrator, who has control over all services in the subscription. In addition to administrators, individual accounts can be granted detailed control of Azure resources through RBAC.

Select and enable an Azure subscription

Before you can work with Azure services, you need a subscription. Several subscription types are available.

Free accounts: The link to sign up for a free account is on the Azure website. This gives you a $200 credit over the course of 30 days to try any combination of resources in Azure. If you exceed your credit amount, your account will be suspended. At the end of the trial, your services will be decommissioned and will no longer work. You can upgrade this to a pay-as-you-go subscription at any time.

MSDN subscriptions: If you have an MSDN subscription, you get a specific amount in Azure credit each month. For example, if you have a Microsoft Visual Studio Enterprise with MSDN subscription, you get $150 per month in Azure credit.

If you exceed the credit amount, your service will be disabled until the next month starts. You can turn off the spending limit and add a credit card to be used for the additional costs. Some of these costs are discounted for MSDN accounts. For example, you pay the Linux price for VMs running Windows Server, and there is no additional charge for Microsoft servers such as Microsoft SQL Server. This makes MSDN accounts ideal for development and test scenarios.

BizSpark accounts: The Microsoft BizSpark program provides a lot of benefits to startups. One of those benefits is access to all the Microsoft software for development and test environments for up to five MSDN accounts. You get $150 in Azure credit for each of those five MSDN accounts, and you pay reduced rates for several of the Azure services, such as Virtual Machines.

Pay-as-you-go: With this subscription, you pay for what you use by attaching a credit card or debit card to the account. If you are an organization, you can also be approved for invoicing.

Enterprise agreements: With an enterprise agreement, you commit to using a certain amount of services in Azure over the next year, and you pay that amount ahead of time. The commitment that you make is consumed throughout the year. If you exceed the commitment amount, you can pay the overage in arrears. Depending on the amount of the commitment, you get a discount on the services in Azure.

For more information and to create an Azure subscription, see How to sign up, purchase, upgrade, or activate Azure.

Grant administrative access to an Azure subscription

Multiple account administrator roles are available and can be changed at any time. Two key roles are:

- **Service administrator**: This role is authorized to manage Azure services. By default, it's granted access to the same account as the account administrator.
- **Co-administrator**: This role has the same access as the service administrator. However, this role cannot change the association of a subscription to Azure directories.

For more information, see How to add or change Azure administrator roles.

View billing information in the Azure portal

An important component of using Azure is the ability to view billing information. The Azure portal provides detailed insight into Azure billing information.

For more information, see How to download your Azure billing invoice and daily usage data.

Get billing information from billing APIs

In addition to viewing the billing in the portal, you can access the billing information by using a script or program through the Azure Billing REST APIs:

- You can use the Azure Usage API to retrieve your usage data. You can fine-tune the billing usage information by tagging related Azure resources. For example, you can tag each of the resources in a resource group with a department name or project name, and then track the costs specifically for that one tag.
- You can use the Azure Rate Card API to list all the available resources, along with the metadata and pricing information about each of those resources.

For more information, see Gain insights into your Microsoft Azure resource consumption.

Forecast cost with the pricing calculator

The pricing for each service in Azure is different. Many Azure services provide Basic, Standard, and Premium tiers. Usually, each tier has several price and performance levels. By using the online pricing calculator, you can create pricing estimates. The calculator includes flexibility to estimate cost on a single resource or a group of resources.

Set up billing alerts

After you have deployed your application or solution on Azure, you can create alerts that send you email when you approach spending limits defined in the alert. For more information, see Set up billing alerts for your Microsoft Azure subscriptions.

Azure Resource Manager

Azure Resource Manager is a deployment, management, and organization mechanism for Azure resources. By using Resource Manager, you can put many individual resources together in a resource group.

Resource Manager also includes deployment capabilities that allow for customizable deployment and configuration of related resources. For instance, by using Resource Manager, you can deploy an application that consists of multiple virtual machines, a load balancer, and a SQL database as a single unit. You develop these deployments by using a Resource Manager template.

Resource Manager provides several benefits:

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

Tips for creating resource groups

When you're making decisions about your resource groups, consider these tips:

- All the resources in a resource group should have the same lifecycle.
- You can assign a resource to only one group at a time.
- You can add or remove a resource from a resource group at any time. Every resource must belong to a resource group. So if you remove a resource from one group, you must add it to another.
- You can move most types of resources to a different resource group at any time.
- The resources in a resource group can be in different regions.
- You can use a resource group to control access for the resources in it.

Building Resource Manager templates

Resource Manager templates declaratively define the resources and resource configurations that will be deployed into a single resource group. You can use Resource Manager templates to orchestrate complex deployments without the need for excess scripting or manual configuration. After you develop a template, you can deploy it multiple times—each time with an identical outcome.

A Resource Manager template consists of four sections:

- **Parameters**: These are inputs to the deployment. Parameter values can be provided by a human or an automated process. An example parameter might be an admin user name and password for a Windows VM. The parameter values will be used throughout the deployment when they're specified.
- **Variables**: These are used to hold values that are used throughout the deployment. Unlike parameters, a variable value is not provided at deployment time. Instead, it's hard coded or dynamically generated.
- **Resources**: This section of the template defines the resources to be deployed, such as virtual machines, storage accounts, and virtual networks.
- **Output**: After a deployment has finished, Resource Manager can return data such as dynamically generated connection strings.

The following mechanisms are available for deployment automation:

- **Functions**: You can use several functions in Resource Manager templates. These include operations such as converting a string to lowercase, deploying multiple instances of a defined resource, and dynamically returning the target resource group. Resource Manager functions help build dynamic deployments.
- **Resource dependencies**: When you're deploying multiple resources, some resources will have a dependency on others. To facilitate deployment, you can use a dependency declaration so that dependent resources are deployed before the others.
- **Template linking**: From within one Resource Manager template, you can link to another template. This allows deployment decomposition into a set of targeted, purpose-specific templates.

You can build Resource Manager templates in any text editor. However, the Azure SDK for Visual Studio includes tooling to assist in the creation process. By using Visual Studio, you can add resources to the template through a wizard. You can then deploy and debug the template directly from the Visual Studio IDE. For more information, see Authoring Azure Resource Manager templates.

Finally, you can convert existing resource groups into a reusable template from the Azure portal. This can be helpful if you want to create a deployable template of an existing resource group, or you just want to examine the underlying JSON. To export a resource group, select the **Automation Script** button from the resource group's settings.

Security of Azure resources (RBAC)

You can grant operational access to user accounts at a specified scope: subscription, resource group, or individual resource. This means you can deploy a set of resources into a resource group, such as a virtual machine and all related resources, and grant permissions to a specific user or group. This approach limits access to only the resources that belong to the target resource group. You can also grant access to a single resource, such as a virtual machine or a virtual network.

To grant access, you assign a role to the user or user group. There are many predefined roles. You can also define your own custom roles.

Here are a few example roles built into Azure:

- **Owner**: A user with this role can manage everything, including access.
- **Reader**: A user with this role can read resources of all types (except secrets) but can't make changes.
- **Virtual machine contributor**: A user with this role can manage virtual machines but can't manage the virtual network to which they are connected or the storage account where the VHD file resides.
- **SQL DB contributor**: A user with this role can manage SQL databases but not their security-related policies.
- **SQL security manager**: A user with this role can manage the security-related policies of SQL servers and databases.
- **Storage account contributor**: A user with this role can manage storage accounts but cannot manage access to the storage accounts.

For more information, see Use role assignments to manage access to your Azure subscription resources.

Azure Virtual Machines

Azure Virtual Machines is one of the central IaaS services in Azure. Azure Virtual Machines supports the deployment of Windows or Linux virtual machines in a Microsoft Azure datacenter. With Azure Virtual Machines, you have total control over the VM configuration and are responsible for all software installation, configuration, and maintenance.

When you're deploying an Azure VM, you can select an image from the Azure Marketplace, or you can provide you own generalized image. This image will be used to apply the operating system and initial configuration. During the deployment, Resource Manager will handle some configuration settings, such as assigning the computer name, administrative credentials, and network configuration. You can use Azure virtual machine extensions to further automate configurations such as software installation, antivirus configuration, and monitoring solutions.

You can create virtual machines in many different sizes. The size of virtual machine dictates resource allocation such as processing, memory, and storage capacity. In some cases, specific features such as RDMA-enabled network adapters and SSD disks are available only with certain VM sizes. For a complete list of VM sizes and capabilities, see "Sizes for virtual machines in Azure" for Windows and Linux.

Use cases

Because Azure virtual machines offer complete control over configuration, they are ideal for a wide range of server workloads that do not fit into a PaaS model. Server workloads such as database servers (SQL Server, Oracle, or MongoDB), Windows Server Active Directory, Microsoft SharePoint, and many more become possible to run on the Microsoft Azure platform. If desired, you can move such workloads from an on-premises datacenter to one or more Azure regions, without a large amount of reconfiguration.

Deployment of virtual machines

You can deploy Azure virtual machines by using the Azure portal, by using automation with the Azure PowerShell module, or by using automation with the cross-platform CLI.

Portal

Deploying a virtual machine by using the Azure portal requires only an active Azure subscription and access to a web browser. You can select many different operating system images with varying configurations. All storage and networking requirements are configured during the deployment. For more information, see "Create your first Windows virtual machine in the Azure portal" for Windows and Linux.

In addition to deploying a virtual machine from the Azure portal, you can deploy an Azure Resource Manager template from the portal. This will deploy and configure all resources as defined in the template. For more information, see Deploy resources with Resource Manager templates and Azure portal.

PowerShell

Deploying an Azure virtual machine by using PowerShell allows for complete deployment automation of all related virtual machine resources, including storage and networking. For more information, see Create a Windows VM using Resource Manager and PowerShell.

In addition to deploying Azure compute resources individually, you can use the Azure PowerShell module to deploy an Azure Resource Manager template. This provides automation to start the deployment action while retaining all benefits of modeling a deployment by using Resource Manager templates. For more information, see Deploy resources with Resource Manager templates and Azure PowerShell.

Command-line interface

As with the PowerShell module, the Azure command-line interface provides deployment automation and can be used on Windows, OS X, or Linux systems. When you're using the Azure CLI **vm quick-create** command, all related virtual machine resources (including storage and networking) and the virtual machine itself are deployed. For more information, see Create a Linux VM in Azure by using the CLI.

Likewise, you can use the Azure CLI to deploy an Azure Resource Manager template. This provides automation to start the deployment action while retaining all benefits of modeling a deployment by using Resource Manager templates. For more information, see Deploy resources with Resource Manager templates and Azure CLI.

Access and security for virtual machines

Accessing a virtual machine from the Internet requires the associated network interface, or load balancer if applicable, to be configured with a public IP address. The public IP address includes a DNS name that will resolve to the virtual machine or load balancer. For more information, see IP addresses in Azure.

You manage access to the virtual machine over the public IP address by using a network security group (NSG) resource. An NSG acts like a firewall and allows or denies traffic across the network interface or subnet on a set of defined ports. For instance, to create a Remote Desktop session with an Azure VM, you need to configure the NSG to allow inbound traffic on port 3389. For more information, see Opening ports to a VM in Azure using the Azure portal.

Finally, as with the management of any computer system, you should provide security for an Azure virtual machine at the operating system by using security credentials and software firewalls.

Azure Storage

Azure Storage is a Microsoft-managed service that provides durable, scalable, and redundant storage. You can add an Azure storage account as a resource to any resource group by using any resource deployment method. Azure includes four storage types: Blob storage, File Storage, Table storage, and Queue storage. When deploying a storage account, two account types are available, General-purpose and Blob storage. A General purpose storage account gives you access to all four storage types. Blob storage accounts are similar to general-purpose accounts, but contain specialized blobs that include hot and cold access tiers. For more information on Blob storage, see Azure Blob storage.

Azure storage accounts can be configured with different levels of redundancy:

- **Locally redundant storage** provides high availability by ensuring that three copies of all data are made synchronously before a write is deemed successful. These copies are stored in a single facility in a single region. The replicas reside in separate fault domains and upgrade domains. This means the data is available even if a storage node that's holding your data fails or is taken offline to be updated.
- **Geo-redundant storage** makes three synchronous copies of the data in the primary region for high availability, and then asynchronously makes three replicas in a paired region for disaster recovery.
- **Read-access geo-redundant storage** is geo-redundant storage plus the ability to read the data in the secondary region. This ability makes it suitable for partial disaster recovery. If there's a problem with the primary region, you can change your application to have read-only access to the paired region.

Use cases

Each storage type has a different use case.

Blob storage

The word *blob* is an acronym for *binary large object*. Blobs are unstructured files like those that you store on your computer. Blob storage can store any type of text or binary data, such as a document, media file, or application installer. Blob storage is also referred to as object storage. Azure Blob storage also holds Azure Virtual Machines data disks.

Azure Storage supports three kinds of blobs:

- **Block blobs** are used to hold ordinary files up to 195 GB in size (4 MB × 50,000 blocks). The primary use case for block blobs is the storage of files that are read from beginning to end, such as media files or image files for websites. They are named block blobs because files larger than 64 MB must be uploaded as small blocks. These blocks are then consolidated (or committed) into the final blob.
- **Page blobs** are used to hold random-access files up to 1 TB in size. Page blobs are used primarily as the backing storage for the VHDs that provide durable disks for Azure Virtual Machines, the IaaS compute service in Azure. They are named page blobs because they provide random read/write access to 512-byte pages.
- **Append blobs** consist of blocks like block blobs, but they are optimized for append operations. These are frequently used for logging information from one or more sources to the same blob. For example, you might write all of your trace logging to the same append blob for an application that's running on multiple VMs. A single append blob can be up to 195 GB.

For more information, see Get started with Azure Blob storage.

File storage

Azure File storage is a service that offers file shares in the cloud by using the standard Server Message Block (SMB) protocol. The service supports both SMB 2.1 and SMB 3.0. With Azure File storage, you can migrate applications that rely on file shares to Azure quickly and without costly rewrites. Applications running on Azure virtual machines, in cloud services, or from on-premises clients can mount a file share in the cloud. This is similar to how a desktop application mounts a typical SMB share. Any number of application components can then mount and access the File storage share simultaneously.

Because a File storage share is a standard SMB file share, applications running in Azure can access data in the share via file system I/O APIs. Developers can therefore use their existing code and skills to migrate existing applications. IT pros can use PowerShell cmdlets to create, mount, and manage File storage shares as part of the administration of Azure applications.

For more information, see Get started with Azure File storage.

Table storage

Azure Table storage is a service that stores structured NoSQL data in the cloud. Table storage is a key/attribute store with a schema-less design. Because Table storage is schema-less, it's easy to adapt your data as the needs of your application evolve. Access to data is fast and cost-effective for all kinds of applications. Table storage is typically significantly lower in cost than traditional SQL for similar volumes of data.

You can use Table storage to store flexible datasets, such as user data for web applications, address books, device information, and any other type of metadata that your service requires. You can store any number of entities in a table. A storage account can contain any number of tables, up to the capacity limit of the storage account.

For more information, see Get started with Azure Table storage.

Queue storage

Azure Queue storage provides cloud messaging between application components. In designing applications for scale, application components are often decoupled so that they can scale independently. Queue storage delivers asynchronous messaging for communication between application components, whether they are running in the cloud, on the desktop, on an on-premises server, or on a mobile device. Queue storage also supports managing asynchronous tasks and building process workflows.

For more information, see Get started with Azure Queue storage.

Deploying a storage account

Portal

Deploying a storage account by using the Azure portal requires only an active Azure subscription and access to a web browser. You can deploy a new storage account into a new or existing resource group. After you've created the storage account, you can create a blob container or file share by using the portal. You can create Table and Queue storage entities programmatically.

In addition to deploying a storage account from the Azure portal, you can deploy an Azure Resource Manager template from the portal. This will deploy and configure all resources as defined in the template, including any storage accounts. For more information, see Deploy resources with Resource Manager templates and Azure portal.

PowerShell

Deploying an Azure storage account by using PowerShell allows for complete deployment automation of the storage account. For more information, see Using Azure PowerShell with Azure Storage.

In addition to deploying Azure resources individually, you can use the Azure PowerShell module to deploy an Azure Resource Manager template. This provides automation to start the deployment action while retaining all benefits of modeling a deployment by using Resource Manager templates. For more information, see Deploy resources with Resource Manager templates and Azure PowerShell.

Command-line interface

As with the PowerShell module, the Azure command-line Interface provides deployment automation and can be used on Windows, OS X, or Linux systems. You can use the Azure CLI **storage account create** command to create a storage account. For more information, see Using the Azure CLI with Azure Storage.

Likewise, you can use the Azure CLI to deploy an Azure Resource Manager template. This provides automation to start the deployment action while retaining all benefits of modeling a deployment by using Resource Manager templates. For more information, see Deploy resources with Resource Manager templates and Azure CLI.

Access and security for Azure Storage

Azure Storage is accessed in a variety of ways, including though the Azure portal, during VM creation and operation, and from Storage client libraries. This section will detail a few of these.

Virtual machine disks

When you're deploying a virtual machine, you also need to create a storage account to hold the virtual machine operating system disk and any additional data disks. You can select an existing storage account or create a new one. Because the maximum size of a blob is 1,024 GB, a single VM disk has a maximum size of 1,023 GB. To configure a larger data disk, you can present multiple data disks to the virtual machine and pool them together as a single logical disk. For more information, see "Storage infrastructure guidelines" for Windows and Linux.

Storage tools

Azure storage accounts can be accessed through many different storage explorers, such as Visual Studio Cloud Explorer. These tools provide the ability to browse through storage accounts and data. For more information and a list of available storage explorers, see Azure Storage client tools.

Storage API

Storage resources can be accessed by any language that can make HTTP/HTTPS requests. Additionally, Azure Storage offers programming libraries for several popular languages. These libraries simplify many aspects of working with Azure Storage by handling details such as synchronous and asynchronous invocation, batching of operations, exception management, automatic retries, and operational behavior. For more information, see Azure Storage service REST API reference.

Storage access keys

Each storage account has two authentication keys, a primary and a secondary. Either of these can be used for storage access operations. These storage keys are used to help secure a storage account and are required for programmatically accessing data. There are two keys to allow occasional rollover of the keys to enhance security. It is critical to keep these secure because their possession, along with the account name, allows unlimited access to any data in the storage account.

Shared access signatures

If you need to allow users to have controlled access to your storage resources, you can create a shared access signature. A shared access signature is a token that can be appended to a URL that enables delegated access to a storage resource. Anyone who possesses the token can access the resource that it points to with the permissions that it specifies, for the period of time that it's valid. For more information, see Using shared access signatures.

Azure Virtual Network

Virtual networks are necessary to support communications between virtual machines. You can define subnets, custom IP address, DNS settings, security filtering, and load balancing. By using a VPN gateway or an ExpressRoute circuit, you can connect Azure virtual networks to your on-premises networks.

Use cases

Cloud-only virtual networks

An Azure virtual network, by default, is accessible only to resources stored in Azure. Resources connected to the same virtual network can communicate with each other. You can associate virtual machine network interfaces and load balancers with a public IP address to make the virtual machine accessible over the Internet. You can help secure access to the publicly exposed resources by using a network security group.

Cross-premises virtual networks

You can connect an on-premises network to an Azure virtual network by using ExpressRoute or a site-to-site VPN connection. In this configuration, the Azure virtual network is essentially a cloud-based extension of your on-premises network.

Because the Azure virtual network is connected to your on-premises network, cross-premises virtual networks must use a unique portion of the address space that your organization uses. In the same way that different corporate locations are assigned a specific IP subnet, Azure becomes another location as you extend your network.

Deploying a virtual network

Portal

Deploying an Azure virtual network by using the Azure portal requires only an active Azure subscription and access to a web browser. You can deploy a new virtual network into a new or existing resource group. When you're creating a new virtual machine from the portal, you can select an existing virtual network or a create a new one. For more information, see Create a virtual network using the Azure portal.

In addition to deploying an Azure virtual network from the Azure portal, you can deploy an Azure Resource Manager template from the portal. This will deploy and configure all resources as defined in the template, including any virtual network resources. For more information, see Deploy resources with Resource Manager templates and Azure portal.

PowerShell

Deploying an Azure virtual network by using PowerShell allows for complete deployment automation of the storage account. For more information, see Create a virtual network by using PowerShell.

In addition to deploying Azure resources individually, you can use the Azure PowerShell module to deploy an Azure Resource Manager template. This provides automation to start the deployment action while retaining all benefits of modeling a deployment by using Resource Manager templates. For more information, see Deploy resources with Resource Manager templates and Azure PowerShell.

Command-line interface

As with the PowerShell module, the Azure command-line interface provides deployment automation and can be used on Windows, OS X, or Linux systems. You can use the Azure CLI **network vnet create** command to create a virtual network. For more information, see Create a virtual network by using the Azure CLI.

Likewise, you can use the Azure CLI to deploy an Azure Resource Manager template. This provides automation to start the deployment action while retaining all benefits of modeling a deployment by using Resource Manager templates. For more information, see Deploy resources with Resource Manager templates and Azure CLI.

Access and security for virtual networks

You can help secure Azure virtual networks by using a network security group. NSGs contain a list of access control list (ACL) rules that allow or deny network traffic to your VM instances in a virtual network. You can associate NSGs with either subnets or individual VM instances within that subnet. When you associate an NSG with a subnet, the ACL rules apply to all the VM instances in that subnet. In addition, you can further restrict traffic to an individual VM by associating an NSG directly with that VM. For more information, see What is a Network Security Group?.