

Introduction to the program

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

The cloud is enabling dramatic changes in the way we live our lives.

[MUSIC]

Play video starting at ::6 and follow transcript0:06

From the health care sector.

>> To the entertainment industry.

>> The cloud allows small businesses to go global,
developing countries to expand into new markets and language barriers to fade.

Play video starting at ::19 and follow transcript0:19

Doctors can remotely monitor patients in their own home.

>> Scientists can predict hurricanes.

>> And education can now be distributed to more people than ever before.

>> From universities to refugee camps.

>> We can use the Cloud AI and

machine learning to help us quickly search through large amounts of data,
helping us make important decisions with the right kind of information.

So instead of guessing and making decisions based on only part of the story,
we are helping scientists, doctors and many other people make better decisions.

With the right knowledge and skills, you could be part of this change.

>> Cloud computing is rapidly expanding to all businesses,
creating new career opportunities.

Career opportunities in cloud computing cover a broad range of roles,
from developers and architects to security professionals and data scientists.

Given the constantly evolving nature of the cloud,
working in the cloud requires continuously updating your knowledge and skills.

>> However, maybe you don't have that specific university degree,

the right certifications and hands on experience.

Or maybe the cost is just too high.

>> For these reasons, Microsoft and Coursera have partner to develop the first in a series of programs to prepare you for a career in the cloud.

This program consists of four courses that will act as a bedrock of fundamental knowledge to prepare you for the 8900 certification exam.

>> The AZ 900 certification is designed to give you the fundamental knowledge, skills and confidence to begin your Azure certification journey.

Play video starting at :1:56 and follow transcript1:56

We've assembled a great team of instructors to prepare you for this journey.

>> I'm Rachel.

In course one introduction toe Azure core concepts and services.

You'll learn the basics of cloud computing, its advantages on how to determine whether Azure is the right solution for your business needs.

You learn about several of the database and big data services that are available on Microsoft Azure.

You'll also learn how to take advantage off several virtualization services in Azure compute.

Which can help your applications scale out quickly on efficiently to meet increasing demands.

Finally, you learn about the different storage and virtual network options available in Azure.

>> I'm Barry, in course, two Azure management tools and security solutions, you learn about AI and software development tools and services from Microsoft Azure.

You learn about monitoring and management tools and services from Azure.

You'll then look at the serverless computing technology and Azure IoT service that best addresses different business scenarios.

Finally, you learn how Azure can help you protect the workloads that you run both in the cloud and in your on premises data center.

>> I'm Anita, in course, three.

Azure services and life cycles, you'll learn how, Azure active directory provides identity and access management.

Then you'll learn how to make organizational decisions about your cloud

environment.

By using the cloud adoption framework for Azure.

You learn how to control and

audit how your resource is air created by using Azure policy and enable governance

at scale across multiple Azure subscriptions by using Azure blueprints.

You'll use the total cost of ownership calculator to compare your current at
a center costs to running the same workloads on Azure.

>> And finally, in course four preparing for the AZ 900 Azure fundamentals exam.

You'll get a more detailed overview of the Microsoft certification program and
where you can go next in your career.

You'll get tips and tricks testing strategies, useful resource is and
information on how to sign up for the AZ 900 exam.

You also get a recap of the key topics and concepts covered in each course,
along with a practice exam.

Finally, you'll get to take the practice exam that tests all the main
topics covered in the AZ 900 proctor's exam,
ensuring you're well prepared for certification success.

Introduction to Azure Core Concepts & Services

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Hello and welcome to the first course in this program,
introduction to Azure core concepts and services.

Before we start diving into the detail,
let's take a look at an overview
of the program and how you

can benefit from Microsoft certification in this area.

Becoming AZ- 900 Microsoft Azure Fundamental certified can be the launch path for your learning journey into Cloud computing and Azure Technologies.

This certification is a good fit if you're beginning to work with Cloud-based solutions and services or are new to Azure.

Acquiring the Azure Fundamental certification is also an opportunity for you to prove your knowledge of client concepts and Azure's main features.

These include, Azure services, Azure workloads, Azure security and privacy, and finally, Azure pricing and support.

This course will help you to develop these skills and prepare you to pass the official Azure AZ-900 exam.

To get the most out of this course, it helps if you are familiar with the general technology concepts, including concepts of networking, storage, compute, application support, and application development.

Microsoft certifications provide globally recognized and industry endorsed evidence of mastering technical skills.

Microsoft certification provides you with the pathway to upgrade your skills, validate your abilities, enhance your professional performance, and develop your career.

Microsoft certifications validate your skills and capabilities, and leads you to success.

Achieving certification shows employers

that you have drive an initiative,
if you get hired in a new role
or promoted or change your career,
your certification speaks volumes
about you and what you know.

In this course, you will explore
various modules related to
Azure core concepts and services.

You'll start off with
an introduction to Azure Fundamentals.

Here, you'll explore basic Cloud concepts,
get a streamlined overview of many Azure services,
and be able to access hands-on exercises to
deploy your very first services for free.

As you navigate this module,
you'll become familiar with Cloud concepts,
Cloud models and platforms
such as infrastructure as a service,
platform as a Service,
and software as a service.

You'll also cover all the core things
you need to know about Cloud computing,
such as elasticity, scalability, and agility.

Next, you'll work through Azure
fundamental concepts and architectural components.

In this module, you'll
learn about the advantages of using
Cloud computing services and how to
differentiate between the categories
and types of Cloud computing.

You'll also examine the various concepts, resources,
and terminology that are
necessary to work with the Azure Architecture.

As you dive deeper,
you'll explore Azure Database Analytics

and Compute Services.

In this module, you will identify several of the database services that are available on Microsoft Azure, such as Azure Cosmos DB, Azure SQL Database, Azure SQL Managed Instance, Azure Database for MySQL, and Azure Database for PostgreSQL.

In addition, you'll learn about several of the big data and analysis services in Azure.

You'll also learn how to take advantage of several virtualization services in Azure Compute, which can help your applications scale out quickly and efficiently to meet increasing demands.

As things become even more exciting, you will navigate the different storage options that are available in Azure Storage and Networking Services.

As you complete the individual units in this module, you'll learn about Azure Blob Storage, Azure Disk Storage, Azure Files, and Blob Access Tiers.

You'll also take a look at several of the core networking resources that are available in Azure.

You'll learn about Azure Virtual Network, which can configure into a customized network environment that meets your company's needs.

You'll also learn how you can use Azure VPN Gateway and Azure Express Right to create secure communication tunnels between your company's different locations.

Throughout this course, you will have an opportunity to get your hands-on experience with

Azure through interactive exercises, practice quizzes, and practice exams. The interactive exercises offer opportunities to practice and implement what you are learning. As an example, when you learn about creating a SQL database, you'll work in a temporary Azure environment called the Sandbox. The beauty about this, is that you will be working with real technology, but in a controlled environment, which allows you to apply what you learn and at your own pace. As you explore the concepts and services that are available through Azure, you'll be given a case study to apply what you're learning to real-world examples. In the case study, you'll assume the role of an IT specialist and address the technology challenges of Tailwind Traders so that you can help them conduct business more efficiently. Using real-world examples helps you to reinforce concepts, prepare you for the exam, and gives you confidence in your approach.

Careers in cloud computing

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using

arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Earning Microsoft certifications is a great option as it shows that you are keeping pace with today's technical roles and requirements.

The AZ-900 Azure Fundamentals forms part of the fundamental level of the Azure journey.

Each Certification pathway starts with AZ-900 as the fundamental level.

Although it is optional for those who have experience, it is recommended as a foundation for next level certifications.

Depending on your goals, there are many certifications and certification paths to choose from.

Let's look at a few examples so that you can get some ideas on how to build your career.

John started his career in an IT support department at the help desk.

He was brand new to Cloud services.

He decided to learn the fundamentals and then explore further opportunities.

He enrolled for the AZ-900 exam as a starting point.

These exams do not assume any prior knowledge and this was a great place for him to get his feet wet.

John's plan for his career was to become an Azure Administrator.

The Azure Administrator Certification, AZ-104 is an associate level certification.

Candidates for the Azure Administrator associate certification should have subject matter expertise implementing, managing, and monitoring an organization's Microsoft Azure environment.

John successfully completed the AZ-900 examination and was promoted into an Azure support role.

He went on to gain six months of experience administering his company's Azure environment.

He needed this experience as a prerequisite to register for the Azure Administrator Certification.

Today, John is a successful Azure Administrator.

He keeps the Azure environment running and responds to daily business expectations.

Ladiesha has always believed that it's important to upscale in a competitive work environment and as made plans to advance her existing career.

She decided to add new credentials to her portfolio and to step beyond software development.

She has more than two years experience in her existing field and

supports the solutions architect with her company's evolving Azure Infrastructure.

Ladiesha decided that she could achieve her goal by completing the Developer Certification.

Developers implement applications and services by partnering with solution architects and customers.

Responsibilities for this role include participating in all phases of Cloud development, from requirements, definition and design to development, deployment and maintenance, performance, tuning and monitoring.

Ladiesha started off with the AZ-900 exam, as she believed this would be a good foundation for success.

Ladiesha has become an Azure Developer at her company and successfully completed the developers Certification Path, which includes Azure Fundamentals AZ-900,

and Azure Developer Associate AZ-204.

Jose lost his job during the COVID crisis.

He had graduated with

a Bachelor's degree in History a few years ago.

Jose gathered two-years experience

as an Azure Administrator,

but he did not have

any certification to validate his knowledge and skills.

Jose has always wanted to be

a DevOps Engineer and he decided that

achieving Certification would open

new career opportunities for him.

Jose was an ideal candidate for

DevOps Engineer certification given

the experience he had gathered as an Azure Administrator.

He completed the AZ-900

Fundamentals exam as a foundation for

further studies and went on to

successfully complete

the DevOps Engineer certification pathway.

After Certification, Jose was able to

secure a role as an Azure Administrator

with a large company and will be considered for

a DevOps Engineer position in the near future.

Microsoft DevOps professionals

combined people, processes,

and technologies to continuously

deliver valuable products

and services that meet user

needs and business objectives.

Based on these scenarios,

you can see that Microsoft provides you with

the ability to start a specific Certification path that

matches your job role while providing you

with the opportunity to improve

your skills at the same time.

There are different certification paths for you to choose from, depending on your expertise and background. Certifications are structured into three expertise levels.

The fundamental certifications are targeted towards those just starting out with the technologies covered or looking to change careers.

The Associate certifications are targeted towards professionals that already have at least two years of practical experience working with the technologies covered.

The expert certifications are targeted towards professionals that have a minimum of five years advanced levels of practical experience and skills with the technologies covered.

We have provided the link to the Microsoft certification at the end of this lesson so that you can explore them in more detail.

Course syllabus

Course Syllabus

This course is the first of a series that aims to help you learn more about Azure and prepares you for the Azure 900 Exam. When you pass the AZ-900 exam, you earn the Microsoft Certified Azure Fundamentals certification.

Module 1: Introduction to Azure Fundamentals

In this module, you are introduced to Azure fundamentals. You'll learn basic cloud concepts, get a streamlined overview of many Azure services, and be able to access hands-on exercises to deploy your very first services for free. After completing this module, you will be able to: 1) Describe the basic concepts of cloud computing; 2) Determine whether Azure is the right solution for your business needs; 3) Differentiate between the different methods of creating an Azure subscription.

Module 2: Azure Fundamental Concepts & Architectural Components

In this module, you'll learn about the advantages of using cloud computing services and how to differentiate between the categories and types of cloud computing. You'll also examine the various concepts, resources, and terminology that are necessary to work with Azure architecture. After completing this module, you will be able to: 1) Identify the benefits and considerations of using cloud services; 2) Describe the differences between categories of cloud services; 3) Describe the differences between types of cloud computing; 4) Understand Azure subscriptions and management groups; 5) Evaluate Azure resources, resource groups, and Azure Resource Manager; 6) Identify Azure regions, region pairs, and availability zones.

Module 3: Azure Database, Analytics, & Compute Services

In this module, you'll learn about several of the database services that are available on Microsoft Azure, such as Azure Cosmos DB, Azure SQL Database, Azure SQL Managed Instance, Azure Database for MySQL, and Azure Database for PostgreSQL. In addition, you'll learn about several of the big data and analysis services in Azure. You'll also learn how to take advantage of several virtualization services in Azure compute, which can help your applications scale out quickly and efficiently to meet increasing demands. After completing this module, you will be able to: 1) Describe Azure Cosmos DB; 2) Describe Azure SQL Database and SQL Managed Instance; 3) Describe Azure Database for MySQL and PostgreSQL 4) Explore big data and analytics; 5) Explore Azure virtual machines and app services; 6) Explain Azure container instances and Kubernetes services; 7) Explore Azure functions and Windows virtual desktop.

Module 4: Azure Storage & Networking Services

In this module, you'll learn about some of the different storage options that are available in Azure Storage services, and the scenarios in which each storage option is appropriate. As you complete the individual units in this module, you'll learn about Azure Blob Storage, Azure Disk Storage, Azure Files, and Blob access tiers. You'll also take a look at several of the core networking resources that are available in Azure. You'll learn about Azure Virtual Network, which you can configure into a customized network environment that meets your company's needs. You'll also learn how you can use Azure VPN Gateway and Azure ExpressRoute to create secure communication tunnels between your company's different locations. After completing this module, you will be able to: 1) Describe Azure Blob Storage, Azure Disk Storage; 2) Understand the benefit of Azure Files; 3) Describe Azure Blob access tiers; 4) Describe the core networking resources that are available in Azure; 5) Describe the benefits and usage of Virtual Network; 6) VPN Gateway, and ExpressRoute.

By the end of this course, you'll be able to:

Understand the benefits of taking the AZ-900 exam and becoming Microsoft Certified

- Explain cloud concepts such as high availability, scalability, elasticity, agility, and disaster recovery
- Compare Azure's database services such as Azure Cosmos DB, Azure SQL, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure's big data and analysis services

- Examine Azure networking resources such as Virtual Networks, VPN Gateways, and Azure ExpressRoute
- Summarize Azure storage services such as Azure Blob Storage, Azure Disk Storage, and Azure File Storage
- Describe core Azure architecture components such as subscriptions, management groups, resources, and resource groups
- Summarize geographic distribution concepts such as Azure regions, region pairs, and availability zones
- Understand the breadth of services available in Azure including compute, network, storage, and database
- Identify virtualization services such as Azure Virtual Machines, Azure Container Instances, Azure Kubernetes Service, and Windows Virtual Desktop
- Identify the benefits of cloud computing in Azure and how it can save you time and money

The benefits of cloud computing

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Hello and welcome to Azure fundamental concepts
in architectural components.

In this module, you'll learn about the main concepts of
Cloud computing and how Azure implements these concepts.

We will also see in our case study how Tailwind Traders
can benefit from migrating
to a Cloud computing environment.

More specifically, after completing this lesson,
you will identify benefits and
considerations of using Cloud services.

You will describe the differences
between categories of Cloud services,
and you will describe
the differences between types of Cloud Computing.

Let's dive right into our case study.

You work in the IT department for Tailwind Traders, which has decided to migrate its applications and data to Microsoft Azure.

You're aware that Cloud computing will save your company time and money by migrating from your existing on-premises physical hardware to a Cloud solution.

With this new solution, you'll only need to pay for the resources and computing time that you use.

However, some of the Cloud Computing Concepts are new to many members of your IT staff.

They've been asking some specific questions about what Cloud computing can do for them.

For example, the team that manages Tailwind Traders website wants to know how Azure improves the site's availability and scalability.

The team that handles the deployment of new hardware is curious to see how Cloud computing can make their deployment processes faster.

In addition, your developer team wants to learn about the different options available to them as they are designing new applications.

For example, is there a way to run their applications in a hybrid configuration where part of their application runs on-premises and the rest of the application runs in the Cloud?

There are several benefits that a Cloud environment has over a physical environment. For example, Cloud-based applications employ a myriad of related strategies.

High availability.

Depending on the service level agreement that you choose, your Cloud-based applications can provide a continuous user-experience with no apparent downtime, even when things go wrong.

Scalability. Applications in the Cloud can be scaled in two ways.

Vertically, computing capacity can be increased by adding RAM or CPUs to a virtual machine. Horizontally, computing capacity can be increased by adding instances of a resource, such as adding more virtual machines to your configuration.

Elasticity. Cloud-based applications can be configured to take advantage of auto-scaling so your applications will always have the resources they need.

Agility. Cloud-based resources can be deployed and configured quickly as your application requirements change.

Geo-distribution.

Applications and data can be deployed to regional data centers around the globe so your customers always have the best performance in their region.

Disaster recovery.

By taking advantage of Cloud-based backup services, data replication, and geo-distribution, you can deploy your applications with the confidence that comes from knowing that your data is safe in the event that disaster should occur. Cloud service providers operate

on a consumption-based model,
which means that end users
only pay for the resources that they use.

Whatever they use is what they pay for.

A consumption-based model has many benefits,
including no upfront costs,
no need to purchase and manage
costly infrastructure that users
might not use to its fullest,
the ability to pay for
additional resources when they are needed,
the ability to stop paying for
resources that are no longer needed.

When analyzing the benefits of Cloud computing,
there are two different types of
expenses that you should consider;
capital expenditure or CapEX,
operational expenditure or OpEX.

Capital expenditure or CapEx is
the upfront spending of money on physical infrastructure,
and then deducting that upfront expense over time.

The upfront cost from CapEx
has a value that reduces over time.

Operational expenditure or OpEX is where you spend money
on products or services and are
build for them at the moment of use.

You can think of these as
the day-to-day expenses that are paid for immediately.

In other words, when
Tailwind Traders owns its infrastructure,
it buys equipment that goes
onto its balance sheets as assets.
Because a capital investment was made,
accountants categorize this transaction as
a CapEx over time to

account for the assets limited useful lifespan, assets are depreciated or amortized. Cloud services, on the other hand, are categorized as an OpEx because of their consumption model. There's no asset for Tailwind Traders to amortize and it's Cloud service provider, Azure, manages the costs that are associated with the purchase and lifespan of the physical equipment. As a result, OpEx has a direct impact on net profit, taxable income, and the associated expenses on the balance sheet. To summarize, CapEx requires significant upfront financial costs, as well as ongoing maintenance and support expenditures. By contrast, OpEx is a consumption-based model. Tailwind Traders is only responsible for the cost of the computing resources that it uses.

Cloud service models

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

If you've been around cloud computing for a while, you've probably seen the terms infrastructure as a service or IaaS.

Platform as a service or PaaS and software as a service or SaaS for the different Cloud Service models.

These models defined the different level of shared responsibility that a cloud

provider and cloud tenant are responsible for.

IaaS is the most flexible category of cloud services, it aims to give you complete control over the hardware that runs your application instead of buying hardware with IaaS you rent it.

PaaS provides the same benefits and considerations as IaaS but there are some additional benefits.

SaaS is software that centrally hosted and managed for you on your users or customers.

Usually, one version of the application is used for all customers and its licensed through a monthly or annual subscription.

Play video starting at ::53 and follow transcript 0:53

SaaS provides the same benefits as IaaS but again there are some additional benefits.

You can see how, when you move from IaaS to PaaS to SaaS that shared responsibilities reduces for the client and increases for the provider.

This chart illustrates the various levels of responsibility between a cloud provider and a cloud tenant.

As you move from on premises through IaaS and PaaS to SaaS the responsibility for maintaining infrastructure, platform and software responsibilities progressively transferred to the provider from the tenant.

Describe the different categories of cloud services

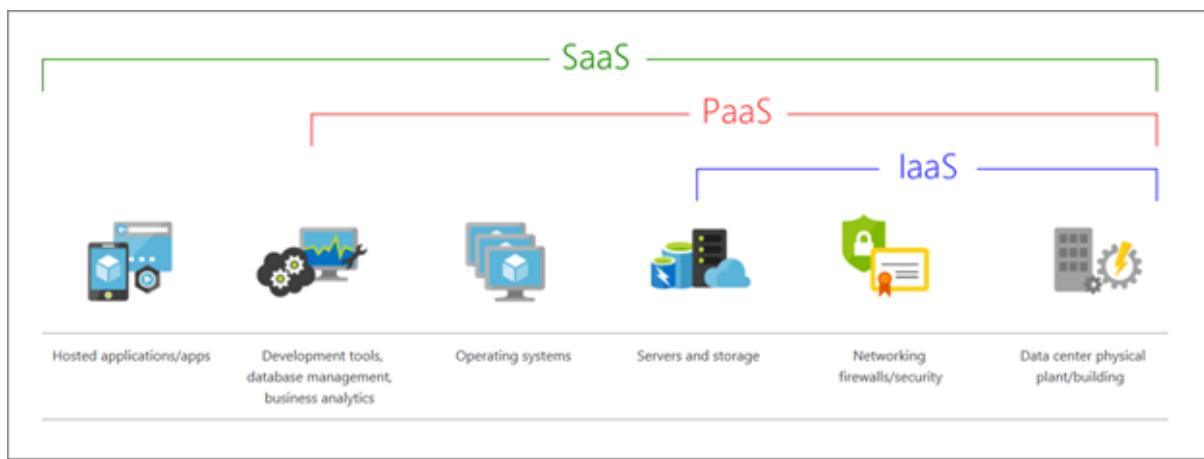
What are cloud service models?

If you've been around cloud computing for a while, you've probably seen the *PaaS*, *IaaS*, and *SaaS* acronyms for the different *cloud service models*. These models define the different levels of shared responsibility that a cloud provider and cloud tenant are responsible for.

	Model Definition	Description
IaaS	<i>Infrastructure-as-a-Service</i>	This cloud service model is the closest to managing physical servers; a cloud provider will keep the hardware up-to-date, but operating system maintenance and network configuration is up to you as the cloud tenant. For example, Azure virtual machines are fully operational virtual compute devices running in Microsoft datacenters. An advantage of this cloud service model is rapid deployment of new compute devices. Setting up a new virtual machine is considerably faster than procuring, installing, and configuring a physical server.
PaaS	<i>Platform-as-a-Service</i>	This cloud service model is a managed hosting environment. The cloud provider manages the virtual machines and networking resources, and the cloud tenant deploys their applications into

	Model Definition	Description
SaaS	<i>Software-as-a-Service</i>	<p>the managed hosting environment. For example, Azure App Services provides a managed hosting environment where developers can upload their web applications, without having to worry about the physical hardware and software requirements.</p> <p>In this cloud service model, the cloud provider manages all aspects of the application environment, such as virtual machines, networking resources, data storage, and applications. The cloud tenant only needs to provide their data to the application managed by the cloud provider. For example, Microsoft Office 365 provides a fully working version of Microsoft Office that runs in the cloud. All you need to do is create your content, and Office 365 takes care of everything else.</p>

The following illustration demonstrates the services that might run in each of the cloud service models:



IaaS

IaaS is the most flexible category of cloud services. It aims to give you complete control over the hardware that runs your application. Instead of buying hardware, with IaaS, you rent it.

Advantages:

- **No CapEx.** Users have no up-front costs.
- **Agility.** Applications can be made accessible quickly and deprovisioned whenever needed.
- **Management.** The shared responsibility model applies; the user manages and maintains the services they have provisioned, and the cloud provider manages and maintains the cloud infrastructure.
- **Consumption-based model.** Organizations pay only for what they use and operate under an Operational Expenditure (OpEx) model.
- **Skills.** No deep technical skills are required to deploy, use, and gain the benefits of a public cloud. Organizations can use the skills and expertise of the cloud provider to ensure workloads are secure, safe, and highly available.
- **Cloud benefits.** Organizations can use the skills and expertise of the cloud provider to ensure workloads are made secure and highly available.
- **Flexibility.** IaaS is the most flexible cloud service because you have control to configure and manage the hardware running your application.

PaaS

Advantages:

- **No CapEx.** Users have no up-front costs.
- **Agility.** PaaS is more agile than IaaS, and users don't need to configure servers for running applications.
- **Consumption-based model.** Users pay only for what they use, and operate under an OpEx model.
- **Skills.** No deep technical skills are required to deploy, use, and gain the benefits of PaaS.
- **Cloud benefits.** Users can take advantage of the skills and expertise of the cloud provider to ensure that their workloads are made secure and highly available. In addition, users can gain access to more cutting-edge development tools. They can then apply these tools across an application's lifecycle.
- **Productivity.** Users can focus on application development only, because the cloud provider handles all platform management. Working with distributed teams as services is easier because the platform is accessed over the internet. You can make the platform available globally more easily.
- **Disadvantage**
- **Platform limitations.** There can be some limitations to a cloud platform that might affect how an application runs. When you're evaluating which PaaS platform is best suited for a workload, be sure to consider any limitations in this area.

SaaS

Advantages:

- **No CapEx.** Users have no up-front costs.
- **Agility.** Users can provide staff with access to the latest software quickly and easily.
- **Pay-as-you-go pricing model.** Users pay for the software they use on a subscription model, typically monthly or yearly, regardless of how much they use the software.
- **Skills.** No deep technical skills are required to deploy, use, and gain the benefits of SaaS.
- **Flexibility.** Users can access the same application data from anywhere.
- **Disadvantage**
- **Software limitations.** There can be some limitations to a software application that might affect how users work. Because you're using as-is software, you don't have direct control of features. When you're evaluating which SaaS platform is best suited for a workload, be sure to consider any business needs and software limitations.

What is serverless computing?

Like PaaS, *serverless computing* enables developers to build applications faster by eliminating the need for them to manage infrastructure. With serverless applications, the cloud service provider automatically provisions, scales, and manages the infrastructure required to run the code. Serverless architectures are highly scalable and event-driven, only using resources when a specific function or trigger occurs.

It's important to note that servers are still running the code. The "serverless" name comes from the fact that the tasks associated with infrastructure provisioning and management are invisible to the developer. This approach enables developers to increase their focus on the business logic, and deliver more value to the core of the business. Serverless computing helps teams increase their productivity and bring products to market faster, and it allows organizations to better optimize resources and stay focused on innovation.

Types of cloud service models

Save note

TranscriptNotesDownloadsDiscuss

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Like PaaS, Serverless Computing
enables developers to build
applications faster by eliminating
the need for them to manage infrastructure.

With Serverless applications,
the Cloud Service Provider
automatically provisions scales,
and manages the infrastructure required to run the code.
Serverless architectures are
highly scalable and event-driven,
only using resources when
a specific function or trigger occurs.

It's important to note that
servers are still running the code.

The serverless name comes from the fact that the tasks
associated with infrastructure provisioning and
management are invisible to the developer.

This approach enables developers
to increase their focus on
the business logic and deliver
more value to the core of the business.

Serverless Computing helps teams increase
their productivity and bring products to market faster,
and it allows organizations to better optimize
resources and stay focused on innovation.

What are public, private, and hybrid clouds?

Save note

TranscriptNotesDownloadsDiscuss

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

There are three deployment models for cloud computing.

Public cloud, private cloud, and hybrid cloud.

Each deployment model has
different aspects that you should
consider as you migrate to the cloud.

In a public cloud, services are offered over
the public internet and
available to anyone who wants to purchase them.

Cloud resources, such as servers and
storage are owned and operated
by a third-party cloud service provider
and delivered over the Internet.

A private cloud consists of computing resources used
exclusively by users from one business or organization.

A private cloud can be physically located at your
organization's on-site or on-premises data center.

It can also be hosted by a third-party service provider.

In the third model,
hybrid cloud computing environments
combine a public cloud and
a private cloud by allowing
data and applications to be shared between them.

This image illustrates several of
the cloud computing concepts

that are presented in this unit.

In this example, several factors

are demonstrated when you were considering

where to deploy a database server

in a hybrid cloud environment.

As your resources moved from on-premises to off-premises,

your costs are reduced and

your administration requirements decrease.

What is a private cloud?

The private cloud is defined as computing services offered either over the Internet or a private internal network and only to select users instead of the general public. Also called an internal or corporate cloud, private cloud computing gives businesses many of the benefits of a [public cloud](#) - including self-service, scalability, and elasticity - with the additional control and customization available from dedicated resources over a computing infrastructure hosted on-premises. In addition, private clouds deliver a higher level of security and privacy through both company firewalls and internal hosting to ensure operations and sensitive data are not accessible to third-party providers. One drawback is that the company's IT department is held responsible for the cost and accountability of managing the private cloud. So private clouds require the same staffing, management, and maintenance expenses as traditional datacenter ownership.

Two models for cloud services can be delivered in a private cloud. The first is [infrastructure as a service \(IaaS\)](#) that allows a company to use infrastructure resources such as compute, network, and storage as a service. The second is [platform as a service \(PaaS\)](#) that lets a company deliver everything from simple cloud-based applications to sophisticated-enabled enterprise applications. Private clouds can also be combined with public clouds to create a [hybrid cloud](#), allowing the business to take advantage of [cloud bursting](#) to free up more space and scale computing services to the public cloud when computing demand increases.

What is a hybrid cloud?

A hybrid cloud—sometimes called a cloud hybrid—is a computing environment that combines an on-premises datacenter (also called a [private cloud](#)) with a [public cloud](#), allowing data and applications to be shared between them. Some people define hybrid cloud infrastructure to

include "multicloud" configurations where an organization uses more than one public cloud in addition to their on-premises datacenter.

What are the benefits of hybrid cloud computing?

No matter which definition of hybrid cloud you use, the benefits are the same: When computing and processing demand increases beyond an on-premises datacenter's capabilities, businesses can use the cloud to instantly scale capacity up or down to handle excess capacity. It also allows businesses to avoid the time and cost of purchasing, installing, and maintaining new servers that they may not always need.

Are there regulatory issues with using a hybrid cloud model?

For industries that work with highly sensitive data, such as banking, finance, government, and healthcare, using a hybrid cloud model may be their best option. For example, some regulated industries require certain types of data to be stored on-premises while allowing less sensitive data to be [stored on the cloud](#). In this kind of [hybrid cloud architecture](#), organizations gain the flexibility of the public cloud for less regulated computing tasks, while still meeting their industry requirements.

Are there security issues with hybrid cloud infrastructure?

Organizations that use a hybrid cloud platform are able to take advantage of many of the same security measures that they use in their existing on-premises infrastructure—including security information and event management (SIEM) capabilities. In fact, some organizations find [cloud hybrid security](#) to be superior to that of their on-premises datacenter because of capabilities such as automated data redundancy, high availability, disaster recovery, and other robust cybersecurity features.

What is a public cloud?

The public cloud is defined as computing services offered by third-party providers over the public Internet, making them available to anyone who wants to use or purchase them. They may be free or sold on-demand, allowing customers to pay only per usage for the CPU cycles, storage, or bandwidth they consume.

Unlike [private clouds](#), public clouds can save companies from the expensive costs of having to purchase, manage, and maintain on-premises hardware and application infrastructure - the [cloud service provider](#) is held responsible for all management and maintenance of the system. Public clouds can also be deployed faster than on-premises infrastructures and with an almost infinitely scalable platform. Every employee of a company can use the same application from any office or branch

using their device of choice as long as they can access the Internet. While security concerns have been raised over public cloud environments, when implemented correctly, the public cloud can be as secure as the most effectively managed private cloud implementation if the provider uses proper security methods, such as intrusion detection and prevention systems (IDPS).

Lesson introduction (ARCHITECTURE)

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Hello and welcome to this lesson on Azure Architecture.

In this lesson, you'll examine
the various concepts, resources,
and terminology that are
necessary to work with Azure Architecture.

After completing this lesson,
you'll be able to describe the benefits and
usage of Azure subscriptions and management groups.

Azure resources, resource groups,
and Azure Resource Manager.

Azure regions, region pairs, and availability zones.

In this lesson, we will
continue working on our case study as
Tailwind Traders plans the adoption of
Azure for their Cloud computing platform.

Let's say that you work as
a developer for Tailwind Traders,
a successful hardware and manufacturing company.
Your company's Chief Technology Officer

recently decided to adopt

Azure as the Cloud computing platform.

You're currently in the planning

stages for the migration.

Before you begin the migration process,

you decide to study Azure concepts, resources,

and terminology to ensure your migration is a success.

Overview of Azure subscriptions, management groups, resources, and regions

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Let's dive right into our case study.

As part of your research for Tailwind traders,

you need to learn the organizing structure

for resources in Azure,

which has four levels, resources,

resource groups, subscriptions, and management groups.

Having seen the top-down hierarchy of organization,

let's describe each of those levels from the bottom up.

Resources. Resources are instances of

services that you create like Virtual Machines,

Storage, or Sequel databases.

Resource groups, resources are

combined into resource groups

which act as a logical container

into which Azure resources like web apps,

databases, and storage accounts are deployed and managed.

Subscriptions.

A subscription groups together user accounts

and the resources that had been

created by those user accounts.

For each subscription, there are limits or

quotas on the amount of

resources that you can create and use.

Organizations can use subscriptions to manage

costs and the resources that are created by users,

teams, or projects.

Management groups.

These groups help you manage access,

policy, and compliance for multiple subscriptions.

All subscriptions in a management group

automatically inherit the conditions

applied to the management group.

You'll examine each of

these four organizational levels in detail.

Azure subscriptions & management groups

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

As Tailwind traders get started with Azure,

one of your first steps will be to

create at least one Azure subscription.

You'll use it to create

your cloud-based resources in Azure.

Azure resource is a manageable item

that's available through Azure.

Virtual Machines, VMs, Storage Accounts,

web apps, Databases, and

virtual networks are all examples of resources.

Using Azure requires an Azure subscription.

A subscription provides you with authenticated and

authorized access to Azure products and services.

It also allows you to provision resources.

An Azure subscription is a logical unit of

Azure services that links to an Azure account.

Which is an identity in Azure Active Directory,

also called Azure AD,

or in a directory that Azure AD trusts.

There are two types of

subscription boundaries that you

can use. Billing boundary.

This subscription type determines

how an Azure account is build for using Azure.

You can create multiple subscriptions

for different types of billing requirements.

Azure generates separate billing reports and invoices for

each subscription so that you

can organize and manage costs.

Access control boundary.

Azure applies access management policies

at the subscription level.

You can create separate subscriptions to

reflect different organizational structures.

An example is that within a business,

you have different departments to which you apply

distinct Azure subscription policies.

This billing model allows you

to manage and control access to

the resources that users provision with specific subscriptions.

You might want to create additional subscriptions for resource of billing management purposes.

For example, you might choose to create additional subscriptions to separate environments.

When managing your resources, you can choose to create subscriptions to set up separate environments for development and testing security, or to isolate data for compliance reasons.

This design is particularly useful because resource access control occurs at the subscription level.

Organizational structures.

You can create subscriptions to reflect different organizational structures.

For example, you could limit a team to lower cost resources while allowing the IT department a full range.

This design allows you to manage and control access to the resources that users provision within each subscription.

Billing. You might want to also create additional subscriptions for billing purposes.

Because costs are first aggregated at the subscription level.

You might want to create subscriptions to manage and track costs based on your needs.

For instance, you might want to create one subscription for your production workloads and another subscription for your development and testing workloads.

You might also need

additional subscriptions because of subscription limits.

Subscriptions are bound to some hard limitations.

For example, the maximum number of Azure express ride circuits per subscription is 10.

Those limits should be considered as you create subscriptions on your account.

If there's a need to go over those limits in particular scenarios, you might need additional subscriptions.

The diagram you will see next shows an overview of how Billing is structured.

If you've previously signed up for Azure or if your organization has an enterprise agreement, your billing might be set up differently.

If you have multiple subscriptions, you can organize them into invoice sections.

Each invoice section is a line item on the invoice that shows the charges incurred that month.

For example, you might need a single invoice for your organization, but want to organize charges by department, team, or project.

Depending on your needs, you can set up multiple invoices within the same billing account.

To do this, create additional billing profiles.

Each billing profile has its own monthly invoice and payment method.

If your organization has many subscriptions, you might need a way to efficiently manage access policies and compliance for their subscriptions.

Azure management groups provide a level of scope above subscriptions.

You organize subscriptions into containers called management groups.

Apply your governance conditions

to the management groups.

All subscriptions within a management group,

automatically inherit the conditions

applied to the management group.

Management groups give you

Enterprise grade management at a large scale,

no matter what type of subscriptions you might have.

All subscriptions within a single management group

must trust the same Azure AD [inaudible].

For example, you can apply policies to

a management group that limits

the regions available for 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.

You can build a flexible structure of

management groups and subscriptions to organize

your resources into a hierarchy for

unified policy and access management.

The following diagram shows an example of

creating a hierarchy for

governance by using management groups.

You can create a hierarchy that applies a policy.

For example, you could limit VM locations to

the US West region in a group called Production.

This policy will inherit onto

all the enterprise agreement subscriptions that are

descendants of that management group and will

apply to all VMs under those subscriptions.

The security policy can't be

altered by the resource or subscription owner,

which allows for improved governance.

Another scenario where you would use management groups

is to provide user access to multiple subscriptions.

By moving multiple subscriptions under that management group, you can create one Role-Based Access Control RBAC assignment on the management group, which will inherit that access to all the subscriptions. One assignment on the management group can enable users to have access to everything they need, instead of scripting RBAC over different subscriptions. We will cover RBAC in more detail later in the course.

There are some important facts about management groups that you should keep in mind. Up to 10,000 management groups can be supported in a single directory.

A management group tree can support up to six levels of depth. This limit doesn't include the root level or at the subscription level.

Each management group and subscription can support only one parent.

Each management group can have many children. All subscriptions and management groups are within a single hierarchy in each directory.

Azure resources and Azure Resource Manager

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

After you've created a subscription for Tailwind traders,
you're ready to start creating resources on storing them in resource groups.
With that in mind, it's important to define those terms.

Resource, a manageable item that's available through Azure
virtual machines or VM storage accounts,
Web APs Databases on virtual networks are examples of Resources.

Resource Group, a container that holds related resources for an Azure solution.
The resource group includes resources that you want to manage as a group.
You decide which resources belong in a resource group based on what makes
the most sense for your organization.

Play video starting at ::40 and follow transcript0:40

Resource groups are a fundamental element off the Azure platform.
Our resource group is a logical container for resources deployed on Azure.
These resources are anything you create in an Azure subscription like
VM Azure application gateway instances and Azure cosmos DB Instances.
All resources must be in a resource group on a resource can only be a member
of a single resource group.

Many resources can be moved between resource groups,
with some services having specific limitations or requirements to move.

Resource groups can't be nested before any resource could be provisioned you
need a resource group for it to be placed in.

Resource groups exist to help manage and organize your Azure resources by
placing resources of similar usage, type or location in a resource group,
you can provide order and organization to resources you create in Azure.

Logical grouping is the aspect that you're most interested in here because
resources can become disordered.

It is a good idea to organize your resources by life cycle in non-production
environments.

If you delete a resource group,
all resources contained within it are also deleted.

Organizing resources by life cycle could be useful in non-production environments
where you might try an experiment and then dispose of it.

Resource groups make it easy to remove a set of resources all at once.
You can also apply permissions to ease administration on limit access.

Resource groups are also a scope for applying role based access control permissions. By applying our back permissions to a resource group, you can ease administration and limit access to allow only what's needed.

Azure Resource Manager commonly referred to his ARM, is the deployment and management service for Azure.

It provides a management layer that enables you to create, update and delete resources in your Azure account.

You use management features like access control locks and tags to secure and organize your resources after deployment.

When a user sends a request from any of the Azure tools APIs or SDKs Resource Manager receives the request.

Play video starting at :2:54 and follow transcript2:54

It authenticates and authorizes the request.

Resource Manager sends the request to the Azure service, which takes the requested action because all requests are handled through the same API, UC consistent results and capabilities in all the different tools.

All capabilities that are available in the azure portal are also available through PowerShell, the Azure CLI,

REST APIs on client SDKs functionality, initially released through APIs will be represented in the portal within 180 days of initial release.

Azure resource manager brings many benefits.

With RM,

you can manage your infrastructure through decorative templates rather than scripts, and RM template is adjacent on file that defines what you want to deploy toe Azure, deploy, manage and monitor all the resources for your solution as a group.

Rather than handling these resources individually, redeploy your solution throughout the development life cycle on have confidence your resources are deployed in a consistent state defined the dependencies between resources, so they're deployed in the correct order.

Apply access control toe all services because our back is natively integrated into the management platform.

Apply tags to resources to logically organize all the resources in your subscription.

Clarify your organization's billing by viewing costs for

a group of resources that share the same tag.

Azure regions and availability zones

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

By now, you should have learned
about Azure resources and resource groups.

Resources are created in regions which are
different geographical locations around
the globe that contain Azure data centers.

Azure is made up of data centers
located around the globe.

When you use a service or create a resource,
such as a SQL database or Virtual Machine,
you're using physical equipment
in one or more of these locations.

The specific data centers
aren't exposed to users directly.

Instead, Azure organizes them into regions.

As you'll see later in this lesson,
some of these regions offer Availability Zones
which are different Azure
data centers within that region.

A region is a geographical area
on the planet that contains
at least one but potentially multiple data centers
that are nearby a network
together with a low latency network.

Azure intelligently assigns and controls the resources within each region to ensure workloads are appropriately balanced.

When you deploy resource in Azure, you'll often need to choose the region where you want your resource deployed.

It's important to note that some services or VM features are only available in certain regions, such as specific VM sizes or Storage types.

There are also some global Azure services that don't require you to select a particular region, such as Azure Active Directory, Azure Traffic Manager, and Azure DNS.

Azure has more global regions than any other Cloud provider.

These regions gives you the flexibility to bring applications closer to your users no matter where they are.

Global regions provide better scalability and redundancy.

They also preserve data residency for your services.

A few examples of regions are West US, Canada Central, West Europe, Australia East, and Japan west.

Here's a view of all the available regions as of June 2020.

Azure has specialized regions that you might want to use when you build out your applications for compliance or legal purposes.

A few examples include US Department of Defense Central, US government, Virginia, US government Iowa, and more.

These regions are physical and logical network isolated instances of Azure for US government agencies and partners.

These data centers are operated by

screened US personnel and include additional compliance certifications. China East, China North, and more. These regions are available through a unique partnership between Microsoft and 21 Vionnet, whereby Microsoft doesn't directly maintain the Data Centers.

Regions are what you use to identify the location for your resources.

There are two other terms you should also be aware of, geographies and Availability Zones.

You want to ensure your services and data are redundant, so you can protect your information in case of failure.

When you host your Infrastructure, setting up your own redundancy requires that you create duplicate hardware environments.

Azure can help make your app highly available through Availability Zones.

Availability Zones are physically separate data centers within an Azure region.

Each availability zone is made up of one or more data centers equipped with independent power, cooling and networking.

An availability zone is set up to be an isolation boundary.

If one zone goes down, the other continues working.

Availability Zones are connected through high-speed private fiber optic networks.

Not every region has support for Availability Zones.

For an updated list, check your documentation for Azure services that support Availability Zones.

You can use Availability Zones to run mission critical applications and built

high availability into your application architecture by co-locating your compute, storage, networking, and data resources within a zone and replicating in other zones.

Keep in mind that there could be a cost to duplicating your services and transferring data between zones.

Availability Zones are primarily for VMs, managed disks, load balances and SQL databases.

Azure services that support Availability Zones fall into two categories: Zonal services, you pin the resource to a specific zone.

For example, VMs, managed disks and IP addresses. Zone-redundant services.

The Platform replicates automatically across zones.

For example, zone-redundant storage, SQL database.

Check the documentation to determine which elements of your architecture you can associate with an Availability Zone.

Availability Zones are created by using one or more data centers.

There's a minimum of three zones within a single region.

It's possible that a large disaster could cause an outage big enough to affect even two data centers.

That's why Azure also creates region pairs.

Each Azure region is always paired with another region within the same geography, such as US, Europe, or Asia, at least 300 miles away.

This approach allows for the replication of resources such as VM storage across a geography that helps reduce the likelihood of interruptions because of events such as natural disasters, civil unrest, power outages,

or physical network outages

that affect both regions at once.

If a region in a pair was

affected by a natural disaster, for instance,

services would automatically fail

over to the other region in its region pair.

An example of a region pair in

Azure is West US paired with East US.

Similarly, in Asia,

Southeast Asia is paired with East Asia.

Because the pair of regions is directly connected

and far enough apart to be

isolated from regional disasters,

you can use them to provide

reliable services on data redundancy.

Some services offer

automatic geo-redundant storage by using

region pairs and there are

some additional advantages of region pairs.

If an extensive Azure outage occurs,

one region out of every pair is prioritized to make sure

at least one is restored as quickly as

possible for applications hosted in that region pair.

Planned Azure updates are rolled out to paired regions,

one region at a time to minimize

downtime and risk of application outage.

Data continues to reside within

the same geography as its pair except for Brazil,

SCIF for tax and law enforcement jurisdiction purposes.

Having a broadly distributed set of data centers

allows Azure to provide a high guarantee of availability.

Introduction (ADDITIONAL NOTES)

Completed 100 XP

- 2 minutes

The term *governance* describes the general process of establishing rules and policies and ensuring that those rules and policies are enforced.

When running in the cloud, a good governance strategy helps you maintain control over the applications and resources that you manage in the cloud. Maintaining control over your environment ensures that you stay compliant with:

- Industry standards, like [PCI DSS](#).
- Corporate or organizational standards, such as ensuring that network data is encrypted.

Governance is most beneficial when you have:

- Multiple engineering teams working in Azure.
- Multiple subscriptions to manage.
- Regulatory requirements that must be enforced.
- Standards that must be followed for all cloud resources.

Meet Tailwind Traders

[Tailwind Traders](#) is a fictitious home improvement retailer. It operates retail hardware stores across the globe and online.



Tailwind Traders specializes in competitive pricing, fast shipping, and a large range of items. It's looking at cloud technologies to improve business operations and support growth into new markets. By moving to the cloud, the company plans to enhance its shopping experience to further differentiate itself from competitors.

How will Tailwind Traders improve agility while maintaining control?

Tailwind Traders is continuing its migration to the cloud. For its existing datacenter, development and test teams must submit support tickets to request access to virtual

machines, storage, and networking components. It can take IT staff anywhere from two weeks to two months to purchase, provision, and configure these components.

By working in the cloud, you essentially have immediate access to compute, storage, and networking components. Many kinds of groups and users, including people from development, test, operations, and security teams, can potentially have direct access to cloud resources.

Going forward, Tailwind Traders could enforce similar processes that prevent teams from directly creating or configuring resources on Azure, similar to its existing approach where central IT provisions infrastructure. But the company knows that these restrictions reduce team agility and the ability to innovate. How can they enable innovation while still maintaining control?

In this module, you'll help the company explore ways it can enforce standards while still enabling teams to create and manage the cloud resources they need.

Learning objectives

After completing this module, you'll be able to:

- Make organizational decisions about your cloud environment by using the Cloud Adoption Framework for Azure.
- Define who can access cloud resources by using Azure role-based access control.
- Apply a resource lock to prevent accidental deletion of your Azure resources.
- Apply tags to your Azure resources to help describe their purpose.
- Control and audit how your resources are created by using Azure Policy.
- Enable governance at scale across multiple Azure subscriptions by using Azure Blueprints.

Control access to cloud resources by using Azure role-based access control

Completed 100 XP

- 4 minutes

When you have multiple IT and engineering teams, how can you control what access they have to the resources in your cloud environment? It's a good security practice to grant users only the rights they need to perform their job, and only to the relevant resources.

Instead of defining the detailed access requirements for each individual, and then updating access requirements when new resources are created, Azure enables you to control access through [Azure role-based access control](#) (Azure RBAC).

Azure provides built-in roles that describe common access rules for cloud resources. You can also define your own roles. Each role has an associated set of access permissions that relate to that role. When you assign individuals or groups to one or more roles, they receive all of the associated access permissions.

How is role-based access control applied to resources?

Role-based access control is applied to a *scope*, which is a resource or set of resources that this access applies to.

Here's a diagram that shows the relationship between roles and scopes.

		Role				
		Reader	Resource-specific	Custom	Contributor	Owner
Scope	Management group					
	Subscription	Observers		Users managing resources		Admins
	Resource group					
	Resource		Automated processes			

Scopes include:

- A management group (a collection of multiple subscriptions).
- A single subscription.
- A resource group.
- A single resource.

Observers, *Users managing resources*, *Admins*, and *Automated processes* illustrate the kinds of users or accounts that would typically be assigned each of the various roles.

When you grant access at a parent scope, those permissions are inherited by all child scopes. For example:

- When you assign the [Owner](#) role to a user at the management group scope, that user can manage everything in all subscriptions within the management group.
- When you assign the [Reader](#) role to a group at the subscription scope, the members of that group can view every resource group and resource within the subscription.
- When you assign the [Contributor](#) role to an application at the resource group scope, the application can manage resources of all types within that resource group, but not other resource groups within the subscription.

When should I use Azure RBAC?

Use Azure RBAC when you need to:

- Allow one user to manage VMs in a subscription and another user to manage virtual networks.
- Allow a database administrator group to manage SQL databases in a subscription.
- Allow a user to manage all resources in a resource group, such as virtual machines, websites, and subnets.
- Allow an application to access all resources in a resource group.

These are just a few examples. You'll find the complete list of built-in roles at the end of this module.

How is Azure RBAC enforced?

Azure RBAC is enforced on any action that's initiated against an Azure resource that passes through Azure Resource Manager. Resource Manager is a management service that provides a way to organize and secure your cloud resources.

You typically access Resource Manager from the Azure portal, Azure Cloud Shell, Azure PowerShell, and the Azure CLI. Azure RBAC doesn't enforce access permissions at the application or data level. Application security must be handled by your application.

RBAC uses an *allow model*. When you're assigned a role, RBAC *allows* you to perform certain actions, such as read, write, or delete. If one role assignment grants you read permissions to a resource group and a different role assignment grants you write permissions to the same resource group, you have both read and write permissions on that resource group.

Who does Azure RBAC apply to?

You can apply Azure RBAC to an individual person or to a group. You can also apply Azure RBAC to other special identity types, such as service principals and managed identities. These identity types are used by applications and services to automate access to Azure resources.

Tailwind Traders has the following teams with an interest in some part of their overall IT environment:

- **IT Administrators** This team has ultimate ownership of technology assets, both on-premises and in the cloud. The team requires full control of all resources.
- **Backup and Disaster Recovery** This team is responsible for managing the health of regular backups and invoking any data or system recoveries.
- **Cost and Billing** People in this team track and report on technology-related spend. They also manage the organization's internal budgets.
- **Security Operations** This team monitors and responds to any technology-related security incidents. The team requires ongoing access to log files and security alerts.

How do I manage Azure RBAC permissions?

You manage access permissions on the **Access control (IAM)** pane in the Azure portal. This pane shows who has access to what scope and what roles apply. You can also grant or remove access from this pane.

The following screenshot shows an example of the **Access control (IAM)** pane for a resource group. In this example, Alain Charon has been assigned the **Backup Operator** role for this resource group.

The screenshot shows the Azure portal interface for managing access control. The top navigation bar includes 'Home', 'Resource groups', 'sales-projectforecast', and 'Access Control - Role assignment'. On the left, a sidebar lists 'Overview', 'Activity log', 'Access control (IAM)' (which is selected and highlighted with a red box), 'Tags', 'Events', 'Settings', and 'Quickstart'. The main content area is titled 'Access Control - Role assignment' for 'sales-projectforecast'. It features search and filter controls for 'Name', 'Role', 'Type', and 'Scope'. Below these, a summary states '8 items (5 Users, 1 Groups, 2 Service Principals)'. A table lists the assignments, with one entry for 'Alain Charon' highlighted with a red box. The table columns are 'NAME', 'TYPE', 'ROLE', and 'SCOPE'. The 'ROLE' column for Alain Charon shows 'BACKUP OPERATOR'. The 'SCOPE' column shows 'This resource'.

NAME	TYPE	ROLE	SCOPE
Alain Charon alain@tailwindtraders.com	User	Backup Operator	This resource

Prevent accidental changes by using resource locks

Completed 100 XP

- 3 minutes

A [resource lock](#) prevents resources from being accidentally deleted or changed.

Even with Azure role-based access control (Azure RBAC) policies in place, there's still a risk that people with the right level of access could delete critical cloud resources. Think of a resource lock as a warning system that reminds you that a resource should not be deleted or changed.

For example, at Tailwind Traders, an IT administrator was performing routine cleanup of unused resources in Azure. The admin accidentally deleted resources that appeared to be unused. But these resources were critical to an application that's used for seasonal promotions. How can resource locks help prevent this kind of incident from happening in the future?

How do I manage resource locks?

You can manage resource locks from the Azure portal, PowerShell, the Azure CLI, or from an Azure Resource Manager template.

To view, add, or delete locks in the Azure portal, go to the **Settings** section of any resource's **Locks** pane in the Azure portal.

Here's an example that shows how to add a resource lock from the Azure portal. You'll apply a similar resource lock in the next part.

The screenshot shows the Azure portal interface for a resource group named 'my-test-rg'. The top navigation bar includes a search bar and an 'Add' button. On the left, there's a sidebar with various options like 'Events', 'Settings', 'Quickstart', 'Deployments', 'Policies', 'Properties', and 'Locks'. The 'Locks' option is highlighted with a red box. Below the sidebar, there are two sections: 'Lock name' and 'This resource'. The 'Lock name' section has a 'Lock name' input field and a 'Lock type' dropdown set to 'CanNotDelete'. The 'This resource' section shows a list of resources with locks applied, including 'my-test-rg' with a lock type of 'CanNotDelete'.

What levels of locking are available?

You can apply locks to a subscription, a resource group, or an individual resource. You can set the lock level to **CanNotDelete** or **ReadOnly**.

- **CanNotDelete** means authorized people can still read and modify a resource, but they can't delete the resource without first removing the lock.
- **ReadOnly** means authorized people can read a resource, but they can't delete or change the resource. Applying this lock is like restricting all authorized users to the permissions granted by the **Reader** role in Azure RBAC.

How do I delete or change a locked resource?

Although locking helps prevent accidental changes, you can still make changes by following a two-step process.

To modify a locked resource, you must first remove the lock. After you remove the lock, you can apply any action you have permissions to perform. This additional step allows the action to be taken, but it helps protect your administrators from doing something they might not have intended to do.

Resource locks apply regardless of RBAC permissions. Even if you're an owner of the resource, you must still remove the lock before you can perform the blocked activity.

Combine resource locks with Azure Blueprints

What if a cloud administrator accidentally deletes a resource lock? If the resource lock is removed, its associated resources can be changed or deleted.

To make the protection process more robust, you can combine resource locks with Azure Blueprints. Azure Blueprints enables you to define the set of standard Azure resources that your organization requires. For example, you can define a blueprint that specifies that a certain resource lock must exist. Azure Blueprints can automatically replace the resource lock if that lock is removed.

You'll learn more about Azure Blueprints later in this module.

Organize your Azure resources by using tags

Completed 100 XP

- 3 minutes

As your cloud usage grows, it's increasingly important to stay organized. A good organization strategy helps you understand your cloud usage and can help you manage costs.

For example, as Tailwind Traders prototypes new ways to deploy its applications on Azure, it needs a way to mark its test environments so that it can easily identify and delete resources in these environments when they're no longer needed.

One way to organize related resources is to place them in their own subscriptions. You can also use resource groups to manage related resources. Resource *tags* are another way to organize resources. Tags provide extra information, or metadata, about your resources. This metadata is useful for:

- **Resource management** Tags enable you to locate and act on resources that are associated with specific workloads, environments, business units, and owners.
- **Cost management and optimization** Tags enable you to group resources so that you can report on costs, allocate internal cost centers, track budgets, and forecast estimated cost.
- **Operations management** Tags enable you to group resources according to how critical their availability is to your business. This grouping helps you formulate service-level agreements (SLAs). An SLA is an uptime or performance guarantee between you and your users.

- **Security** Tags enable you to classify data by its security level, such as *public* or *confidential*.
- **Governance and regulatory compliance** Tags enable you to identify resources that align with governance or regulatory compliance requirements, such as ISO 27001. Tags can also be part of your standards enforcement efforts. For example, you might require that all resources be tagged with an owner or department name.
- **Workload optimization and automation** Tags can help you visualize all of the resources that participate in complex deployments. For example, you might tag a resource with its associated workload or application name and use software such as Azure DevOps to perform automated tasks on those resources.

How do I manage resource tags?

You can add, modify, or delete resource tags through PowerShell, the Azure CLI, Azure Resource Manager templates, the REST API, or the Azure portal.

You can also manage tags by using Azure Policy. For example, you can apply tags to a resource group, but those tags aren't automatically applied to the resources within that resource group. You can use Azure Policy to ensure that a resource inherits the same tags as its parent resource group. You'll learn more about Azure Policy later in this module.

You can also use Azure Policy to enforce tagging rules and conventions. For example, you can require that certain tags be added to new resources as they're provisioned. You can also define rules that reapply tags that have been removed.

An example tagging structure

A resource tag consists of a name and a value. You can assign one or more tags to each Azure resource.

After reviewing its business requirements, Tailwind Traders decides on the following tags.

Name

Value

AppName

The name of the application that the resource is part of.

CostCenter

The internal cost center code.

Owner

The name of the business owner who's responsible for the resource.

Environment

An environment name, such as "Prod," "Dev," or "Test."

Impact

How important the resource is to business operations, such as "Mission-critical," "High-impact," or "Low-impact."

Here's an example that shows these tags as they're applied to a virtual machine during provisioning.

Name ⓘ	Value ⓘ	Resource
AppName	: SpecialOrders	Virtual machine
CostCenter	: 0224 - Infrastructure R&D	Virtual machine
Owner	: tim@tailwindtraders.com	Virtual machine
Environment	: Test	Virtual machine
Impact	: High-impact	Virtual machine

The Tailwind Traders team can run queries, for example, from PowerShell or the Azure CLI, to list all resources that contain these tags.

Keep in mind that you don't need to enforce that a specific tag is present on all of your resources. For example, you might decide that only mission-critical resources have the **Impact** tag. All non-tagged resources would then not be considered as mission-critical.

Control and audit your resources by using Azure Policy

Completed 100 XP

- 5 minutes

In a previous exercise in this module, you identified your governance and business requirements. How do you ensure that your resources *stay* compliant? Can you be alerted if a resource's configuration has changed?

[Azure Policy](#) is a service in Azure that enables you to create, assign, and manage policies that control or audit your resources. These policies enforce different rules across all of your resource configurations so that those configurations stay compliant with corporate standards.

How does Azure Policy define policies?

Azure Policy enables you to define both individual policies and *groups* of related policies, known as *initiatives*. Azure Policy evaluates your resources and highlights resources that aren't compliant with the policies you've created. Azure Policy can also prevent noncompliant resources from being created.

Azure Policy comes with built-in policy and initiative definitions for Storage, Networking, Compute, Security Center, and Monitoring. For example, if you define a policy that allows only a certain SKU (stock-keeping unit) size for the virtual machines (VMs) to be used in your environment, that policy is invoked when you create a new VM and whenever you resize existing VMs. Azure Policy also evaluates and monitors all current VMs in your environment.

In some cases, Azure Policy can automatically remediate noncompliant resources and configurations to ensure the integrity of the state of the resources. For example, if all resources in a certain resource group should be tagged with **AppName** tag and a value of "SpecialOrders," Azure Policy will automatically reapply that tag if it was missing.

Azure Policy also integrates with Azure DevOps by applying any continuous integration and delivery pipeline policies that pertain to the pre-deployment and post-deployment phases of your applications.

Azure Policy in action

Implementing a policy in Azure Policy involves three tasks:

1. Create a policy definition.
2. Assign the definition to resources.
3. Review the evaluation results.

Let's examine each step in more detail.

Task 1. Create a policy definition

A policy definition expresses what to evaluate and what action to take. For example, you could prevent VMs from being deployed in certain Azure regions. You also could audit your storage accounts to verify that they only accept connections from allowed networks.

Every policy definition has conditions under which it's enforced. A policy definition also has an accompanying effect that takes place when the conditions are met. Here are some example policy definitions:

- **Allowed virtual machine SKUs** This policy enables you to specify a set of VM SKUs that your organization can deploy.
- **Allowed locations** This policy enables you to restrict the locations that your organization can specify when it deploys resources. Its effect is used to enforce your geographic compliance requirements.
- **MFA should be enabled on accounts with write permissions on your subscription** This policy requires that multifactor authentication (MFA) be enabled for all subscription accounts with write privileges to prevent a breach of accounts or resources.
- **CORS should not allow every resource to access your web applications** Cross-origin resource sharing (CORS) is an HTTP feature that enables a web application running under one domain to access resources in another domain. For security reasons, modern web browsers restrict cross-site scripting by default. This policy allows only required domains to interact with your web app.
- **System updates should be installed on your machines** This policy enables Azure Security Center to recommend missing security system updates on your servers.

Task 2. Assign the definition to resources

To implement your policy definitions, you assign definitions to resources. A *policy assignment* is a policy definition that takes place within a specific scope. This scope could be a management group (a collection of multiple subscriptions), a single subscription, or a resource group.

Policy assignments are inherited by all child resources within that scope. If a policy is applied to a resource group, that policy is applied to all resources within that resource group. You can exclude a subscope from the policy assignment if there are specific child resources you need to be exempt from the policy assignment.

Task 3. Review the evaluation results

When a condition is evaluated against your existing resources, each resource is marked as compliant or noncompliant. You can review the noncompliant policy results and take any action that's needed.

Policy evaluation happens about once per hour. If you make changes to your policy definition and create a policy assignment, that policy is evaluated over your resources within the hour.

What are Azure Policy initiatives?

An Azure Policy initiative is a way of grouping related policies together. The initiative definition contains all of the policy definitions to help track your compliance state for a larger goal.

For example, Azure Policy includes an initiative named **Enable Monitoring in Azure Security Center**. Its goal is to monitor all of the available security recommendations for all Azure resource types in Azure Security Center.

Under this initiative, the following policy definitions are included:

- **Monitor unencrypted SQL Database in Security Center** This policy monitors for unencrypted SQL databases and servers.
- **Monitor OS vulnerabilities in Security Center** This policy monitors servers that don't satisfy the configured OS vulnerability baseline.
- **Monitor missing Endpoint Protection in Security Center** This policy monitors for servers that don't have an installed endpoint protection agent.

In fact, the **Enable Monitoring in Azure Security Center** initiative contains over 100 separate policy definitions.

Azure Policy also includes initiatives that support regulatory compliance standards, such as HIPAA and ISO 27001.

How do I define an initiative?

You define initiatives by using the Azure portal or command-line tools. From the Azure portal, you can search the list of built-in initiatives that are built into Azure. You also can create your own custom policy definition.

The following image shows a few example Azure Policy initiatives in the Azure portal.

Name	Definition location	Policies	Type
azuresecuritypack...	Non Production	3	Custom
azuresecuritypack...	Non Production	3	Custom
audit ssh auth_1.3	Non Production	4	Custom
audit ssh auth_1.1	Non Production	2	Custom
azuresecuritypack...	5e116433-8b65-49e...	3	Custom
azuresecuritypack...	5e116433-8b65-49e...	3	Custom
audit ssh auth_1.1	5e116433-8b65-49e...	2	Custom
audit ssh auth_1.1	Demonstration	2	Custom
Audit Windows V...		2	Built-in
Audit Windows V...		2	Built-in

How do I assign an initiative?

Like a policy assignment, an initiative assignment is an initiative definition that's assigned to a specific scope of a management group, a subscription, or a resource group.

Even if you have only a single policy, an initiative enables you to increase the number of policies over time. Because the associated initiative remains assigned, it's easier to add and remove policies without the need to change the policy assignment for your resources.

Govern multiple subscriptions by using Azure Blueprints

Completed 100 XP

- 4 minutes

So far, you've explored a number of Azure features that can help you implement your governance decisions, monitor the compliance of your cloud resources, and control access and protect critical resources from accidental deletion.

What happens when your cloud environment starts to grow beyond just one subscription? How can you scale the configuration of these features, knowing they need to be enforced for resources in new subscriptions?

Instead of having to configure features like Azure Policy for each new subscription, with [Azure Blueprints](#) you can define a repeatable set of governance tools and standard Azure resources that your organization requires. In this way, development teams can rapidly build and deploy new environments with the knowledge that they're building within organizational compliance with a set of built-in components that speed the development and deployment phases.

Azure Blueprints orchestrates the deployment of various resource templates and other artifacts, such as:

- Role assignments
- Policy assignments
- Azure Resource Manager templates
- Resource groups

Azure Blueprints in action

When you form a cloud center of excellence team or a cloud custodian team, that team can use Azure Blueprints to scale their governance practices throughout the organization.

Implementing a blueprint in Azure Blueprints involves these three steps:

1. Create an Azure blueprint.
2. Assign the blueprint.
3. Track the blueprint assignments.

With Azure Blueprints, the relationship between the blueprint definition (what should be deployed) and the blueprint assignment (what was deployed) is preserved. In other words, Azure creates a record that associates a resource with the blueprint that defines it. This connection helps you track and audit your deployments.

Blueprints are also versioned. Versioning enables you to track and comment on changes to your blueprint.

What are blueprint artifacts?

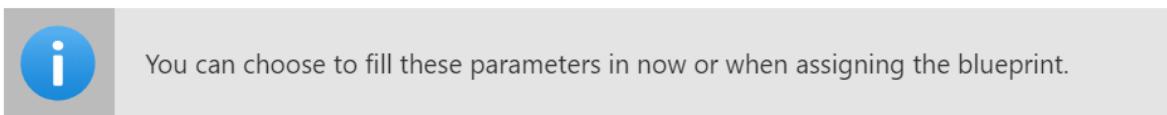
Each component in the blueprint definition is known as an *artifact*.

It is possible for artifacts to have no additional parameters (configurations). An example is the **Deploy threat detection on SQL servers** policy, which requires no additional configuration.

Artifacts can also contain one or more parameters that you can configure. The following screenshot shows the **Allowed locations** policy. This policy includes a parameter that specifies the allowed locations.

Allowed locations

This policy enables you to restrict the locations your organization can specify when deploying resources. Use to enforce your geo-compliance requirements. Excludes resource groups, Microsoft.AzureActiveDirectory/b2cDirectories, and resources that use the 'global' region.



Allowed locations

0 selected



This value should be specified when the blueprint is assigned

You can specify a parameter's value when you create the blueprint definition or when you assign the blueprint definition to a scope. In this way, you can maintain one standard blueprint but have the flexibility to specify the relevant configuration parameters at each scope where the definition is assigned.

How will Tailwind Traders use Azure Blueprints for ISO 27001 compliance?

[ISO 27001](#) is a standard that applies to the security of IT systems, published by the International Organization for Standardization. As part of its quality process, Tailwind Traders wants to certify that it complies with this standard. Azure Blueprints has several built-in blueprint definitions that relate to ISO 27001.

As an IT administrator, you decide to investigate the **ISO 27001: Shared Services Blueprint** definition. Here's an outline of your plan.

1. Define a management group that's named **PROD-MG**. Recall that a management group manages access, policies, and compliance across multiple Azure subscriptions. Every new Azure subscription is added to this management group when the subscription is created.

2. Create a blueprint definition that's based on the **ISO 27001: Shared Services Blueprint** template. Then publish the blueprint.
3. Assign the blueprint to your **PROD-MG** management group.

The following image shows artifacts that are created when you run an ISO 27001 blueprint from a template.

Create blueprint

<input checked="" type="checkbox"/> Enforce encryption on Data Lake Store accounts	Policy assignment	None
<input checked="" type="checkbox"/> Require blob encryption for storage accounts	Policy assignment	None
+ Add artifact...		
✓ <input checked="" type="checkbox"/> Log Analytics resource group	Resource group	2 out of 2 parameters populated
<input checked="" type="checkbox"/> Log Analytics template	Azure Resource Manager te...	0 out of 4 parameters populated
+ Add artifact...		
✓ <input checked="" type="checkbox"/> Network resource group	Resource group	2 out of 2 parameters populated
<input checked="" type="checkbox"/> Azure Firewall template	Azure Resource Manager te...	0 out of 3 parameters populated
<input checked="" type="checkbox"/> Virtual Network and Route Table template	Azure Resource Manager te...	0 out of 9 parameters populated

You see that the blueprint template contains policy assignments, Resource Manager templates, and resource groups. The blueprint deploys these artifacts to any existing subscriptions within the **PROD-MG** management group. The blueprint also deploys these artifacts to any new subscriptions as they're created and added to the management group.

Accelerate your cloud adoption journey by using the Cloud Adoption Framework for Azure

Completed 100 XP

- 3 minutes

The [Cloud Adoption Framework for Azure](#) provides you with proven guidance to help with your cloud adoption journey. The Cloud Adoption Framework helps you create and implement the business and technology strategies needed to succeed in the cloud.

Tailwind Traders needs to control its cloud environment so that it complies with several industry standards, but it's not sure where to start. It has existing business requirements, and it understands how these requirements relate to its on-premises

workloads. These requirements also must be met by any workloads it runs in the cloud.

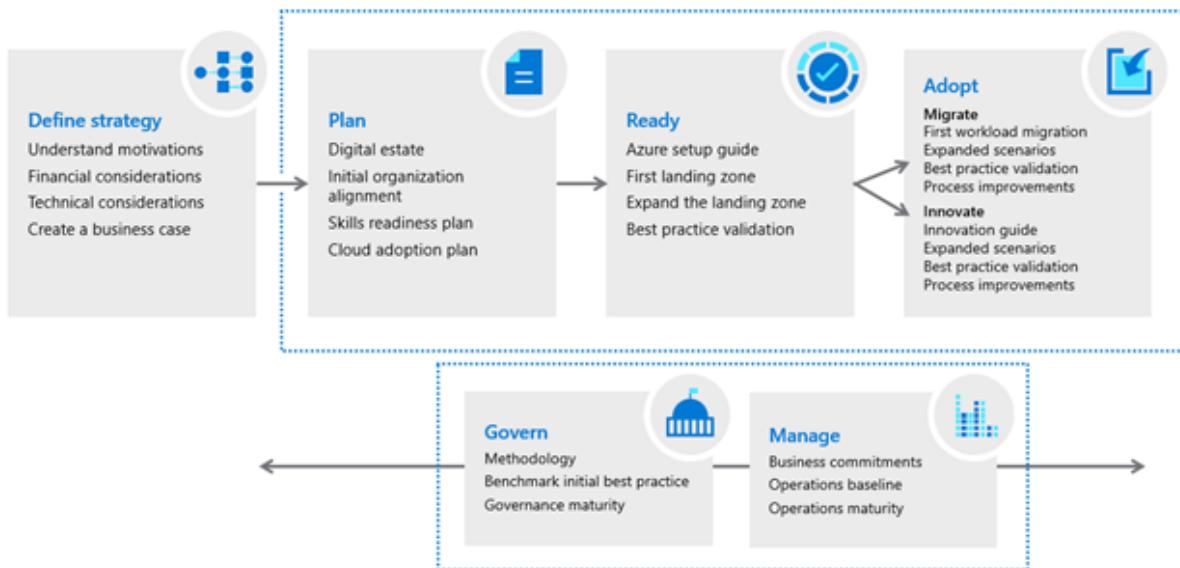
You've been tasked with investigating what's available on Azure and to define and implement the governance strategy for Tailwind Traders. You decide to start with the Cloud Adoption Framework.

What's in the Cloud Adoption Framework?

As mentioned in the video, Cloud Adoption Framework consists of tools, documentation, and proven practices. The Cloud Adoption Framework includes these stages:

1. Define your strategy.
2. Make a plan.
3. Ready your organization.
4. Adopt the cloud.
5. Govern and manage your cloud environments.

Microsoft Cloud Adoption Framework for Azure



The govern stage focuses on cloud governance. You can refer back to the Cloud Adoption Framework for recommended guidance as you build your cloud governance strategy.

To help build your adoption strategy, the Cloud Adoption Framework breaks out each stage into further exercises and steps. Let's take a brief look at each stage.

Define your strategy

Here, you answer why you're moving to the cloud and what you want to get out of cloud migration. Do you need to scale to meet demand or reach new markets? Will it reduce costs or increase business agility? When you define your cloud business strategy, you should understand [cloud economics](#) and consider business impact, turnaround time, global reach, performance, and more.

Here are the steps in this stage.



Define and document your motivations: Meeting with stakeholders and leadership can help you answer why you're moving to the cloud.



Document business outcomes: Meet with leadership from your finance, marketing, sales, and human resource groups to help you document your goals.



Evaluate financial considerations: Measure objectives and identify the return expected from a specific investment.



Understand technical considerations: Evaluate those technical considerations through the selection and completion of your first technical project.

Make a plan

Here, you build a plan that maps your aspirational goals to specific actions. A good plan helps ensure that your efforts map to the desired business outcomes.

Here are the steps in this stage.

Digital estate: Create an inventory of the existing digital assets and workloads that you plan to migrate to the cloud.

A light gray circle with a thin blue border, containing the black number '1' in the center.

1

Initial organizational alignment: Ensure that the right people are involved in your migration efforts, both from a technical standpoint as well as from a cloud governance standpoint.

A light gray circle with a thin blue border, containing the black number '2' in the center.

2

Skills readiness plan: Build a plan that helps individuals build the skills they need to operate in the cloud.

A light gray circle with a thin blue border, containing the black number '3' in the center.

3

Cloud adoption plan: Build a comprehensive plan that brings together the development, operations, and business teams toward a shared cloud adoption goal.

A light gray circle with a thin blue border, containing the black number '4' in the center.

4

Ready your organization

Here, you create a *landing zone*, or an environment in the cloud to begin hosting your workloads.

Here are the steps in this stage.

A light gray circle with a thin blue border, containing the black number '1' in the center.

1

Azure setup guide: Review the Azure setup guide to become familiar with the tools and approaches you need to use to create a landing zone.



2

Azure landing zone: Begin to build out the Azure subscriptions that support each of the major areas of your business. A landing zone includes cloud infrastructure as well as governance, accounting, and security capabilities.



3

Expand the landing zone: Refine your landing zone to ensure that it meets your operations, governance, and security needs.



4

Best practices: Start with recommended and proven practices to help ensure that your cloud migration efforts are scalable and maintainable.

Adopt the cloud

Here, you begin to migrate your applications to the cloud. Along the way, you might find ways to modernize your applications and build innovative solutions that use cloud services.

The Cloud Adoption Framework breaks this stage into two parts: migrate and innovate.

Migrate: Here are the steps in the migrate part of this stage.

Migrate your first workload: Use the Azure migration guide to deploy your first project to the cloud.



1

Migration scenarios: Use additional in-depth guides to explore more complex migration scenarios.

2

Best practices: Check in with the Azure cloud migration best practices checklist to verify that you're following recommended practices.

3

Process improvements: Identify ways to make the migration process scale while requiring less effort.

4

Innovate: Here are the steps in the innovate part of this stage.

1

Business value consensus: Verify that investments in new innovations add value to the business and meet customer needs.

2

Azure innovation guide: Use this guide to accelerate development and build a minimum viable product (MVP) for your idea.

3

Best practices: Verify that your progress maps to recommended practices before you move forward.



4

Feedback loops: Check in frequently with your customers to verify that you're building what they need.

Govern and manage your cloud environments

Here, you begin to form your cloud governance and cloud management strategies. As the cloud estate changes over time, so do cloud governance processes and policies. You need to create resilient solutions that are constantly optimized.

Govern: Here are the steps in the govern part of this stage.

Methodology: Consider your end state solution. Then define a methodology that incrementally takes you from your first steps all the way to full cloud governance.



1

Benchmark: Use the [governance benchmark tool](#) to assess your current state and future state to establish a vision for applying the framework.



2

Initial governance foundation: Create an MVP that captures the first steps of your governance plan.



3

Improve the initial governance foundation: Iteratively add governance controls that address tangible risks as you progress toward your end state solution.



4

Manage: Here are the steps in the manage part of this stage.



Establish a management baseline: Define your minimum commitment to operations management. A management baseline is the minimum set of tools and processes that should be applied to every asset in an environment.



Define business commitments: Document supported workloads to establish operational commitments with the business and agree on cloud management investments for each workload.



Expand the management baseline: Apply recommended best practices to iterate on your initial management baseline.



Advanced operations and design principles: For workloads that require a higher level of business commitment, perform a deeper architecture review to deliver on your resiliency and reliability commitments.

Create a subscription governance strategy

Completed 100 XP

- 3 minutes

At the beginning of any cloud governance implementation, you identify a cloud organization structure that meets your business needs. This step often involves forming a *cloud center of excellence team* (also called a *cloud enablement team* or a *cloud custodian team*). This team is empowered to implement governance practices from a centralized location for the entire organization.

Teams often start their Azure governance strategy at the subscription level. There are three main aspects to consider when you create and manage subscriptions: billing, access control, and subscription limits.

Let's look at each of these aspects in more detail.

Billing

You can create one billing report per subscription. If you have multiple departments and need to do a "chargeback" of cloud costs, one possible solution is to organize subscriptions by department or by project.

Resource tags can also help. You'll explore tags later in this module. When you define how many subscriptions you need and what to name them, take into account your internal billing requirements.

Access control

A subscription is a deployment boundary for Azure resources. Every subscription is associated with an Azure Active Directory tenant. Each tenant provides administrators the ability to set granular access through defined roles by using Azure role-based access control.

When you design your subscription architecture, consider the deployment boundary factor. For example, do you need separate subscriptions for development and for production environments? With separate subscriptions, you can control access to each one separately and isolate their resources from one another.

Subscription limits

Subscriptions also have some resource limitations. For example, the maximum number of network Azure ExpressRoute circuits per subscription is 10. Those limits should be considered during your design phase. If you'll need to exceed those limits, you might need to add more subscriptions. If you hit a hard limit maximum, there's no flexibility to increase it.

Management groups are also available to assist with managing subscriptions. A management group manages access, policies, and compliance across multiple Azure subscriptions. You'll learn more about management groups later in this module.

What are Azure management groups?

- Article
- 04/21/2023
- 16 contributors

Feedback

In this article

1. [Hierarchy of management groups and subscriptions](#)
2. [Root management group for each directory](#)
3. [Initial setup of management groups](#)
4. [Management group access](#)

Show 4 more

If your organization has many Azure subscriptions, you may need a way to efficiently manage access, policies, and compliance for those subscriptions. *Management groups* provide a governance scope above subscriptions. You organize subscriptions into management groups; the governance conditions you apply cascade by inheritance to all associated subscriptions.

Management groups give you enterprise-grade management at scale no matter what type of subscriptions you might have. However, all subscriptions within a single management group must trust the same Azure Active Directory (Azure AD) tenant.

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 nested management groups, subscriptions, and resources, and allow VM creation only in authorized regions.

Hierarchy of management groups and subscriptions

You can build a flexible structure of management groups and subscriptions to organize your resources into a hierarchy for unified policy and access management. The following diagram shows an example of creating a hierarchy for governance using management groups.

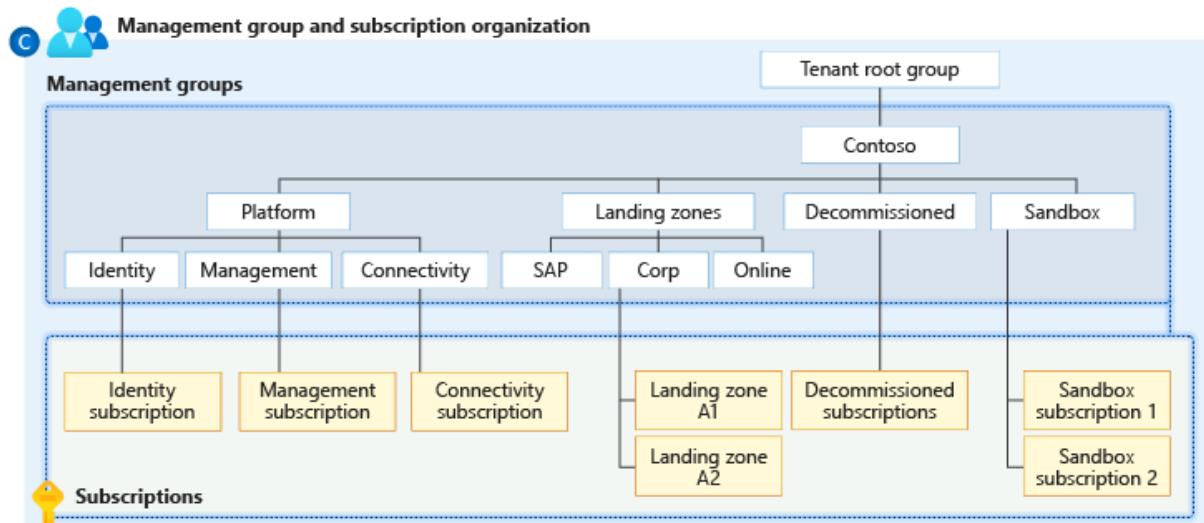


Diagram of a root management group holding both management groups and subscriptions. Some child management groups hold management groups, some hold subscriptions, and some hold both. One of the examples in the sample hierarchy is four levels of management groups with the child level being all subscriptions.

You can create a hierarchy that applies a policy, for example, which limits VM locations to the West US region in the management group called "Corp". This policy will inherit onto all the Enterprise Agreement (EA) subscriptions that are descendants of that management group and will apply to all VMs under those subscriptions. This security policy cannot be altered by the resource or subscription owner allowing for improved governance.

Note

Management groups aren't currently supported in Cost Management features for Microsoft Customer Agreement (MCA) subscriptions.

Another scenario where you would use management groups is to provide user access to multiple subscriptions. By moving multiple subscriptions under that management group, you can create one [Azure role assignment](#) on the management group, which will inherit that access to all the subscriptions. One assignment on the management group can enable users to have access to everything they need instead of scripting Azure RBAC over different subscriptions.

Important facts about management groups

- 10,000 management groups can be supported in a single directory.
- A management group tree can support up to six levels of depth.
 - This limit doesn't include the Root level or the subscription level.
- Each management group and subscription can only support one parent.

- Each management group can have many children.
- All subscriptions and management groups are within a single hierarchy in each directory. See [Important facts about the Root management group](#).

Root management group for each directory

Each directory is given a single top-level management group called the **root** management group. The root management group is built into the hierarchy to have all management groups and subscriptions fold up to it. This root management group allows for global policies and Azure role assignments to be applied at the directory level. The [Azure AD Global Administrator needs to elevate themselves](#) to the User Access Administrator role of this root group initially. After elevating access, the administrator can assign any Azure role to other directory users or groups to manage the hierarchy. As administrator, you can assign your own account as owner of the root management group.

Important facts about the root management group

- By default, the root management group's display name is **Tenant root group** and operates itself as a management group. The ID is the same value as the Azure Active Directory (Azure AD) tenant ID.
- To change the display name, your account must be assigned the **Owner** or **Contributor** role on the root management group. See [Change the name of a management group](#) to update the name of a management group.
- The root management group can't be moved or deleted, unlike other management groups.
- All subscriptions and management groups fold up to the one root management group within the directory.
 - All resources in the directory fold up to the root management group for global management.
 - New subscriptions are automatically defaulted to the root management group when created.
- All Azure customers can see the root management group, but not all customers have access to manage that root management group.
 - Everyone who has access to a subscription can see the context of where that subscription is in the hierarchy.
 - No one is given default access to the root management group. Azure AD Global Administrators are the only users that can elevate themselves to gain access. Once they have access to the root management group, the global administrators can assign any Azure role to other users to manage it.

Important

Any assignment of user access or policy on the root management group **applies to all resources within the directory**. Because of this, all customers should evaluate the need to have items defined on this scope. User access and policy assignments should be "Must Have" only at this scope.

Initial setup of management groups

When any user starts using management groups, there's an initial setup process that happens. The first step is the root management group is created in the directory. Once this group is created, all existing subscriptions that exist in the directory are made children of the root management group. The reason for this process is to make sure there's only one management group hierarchy within a directory. The single hierarchy within the directory allows administrative customers to apply global access and policies that other customers within the directory can't bypass. Anything assigned on the root will apply to the entire hierarchy, which includes all management groups, subscriptions, resource groups, and resources within that Azure AD tenant.

Management group access

Azure management groups support [Azure role-based access control \(Azure RBAC\)](#) for all resource accesses and role definitions. These permissions are inherited to child resources that exist in the hierarchy. Any Azure role can be assigned to a management group that will inherit down the hierarchy to the resources. For example, the Azure role VM contributor can be assigned to a management group. This role has no action on the management group, but will inherit to all VMs under that management group.

The following chart shows the list of roles and the supported actions on management groups.

Azure Role Name	Create	Rename	Move**	Delete	Assign Access	Assign Policy	Read
Owner	X	X	X	X	X	X	X
Contributor	X	X	X	X			X
MG Contributor*	X	X	X	X			X
Reader							X
MG Reader*							X
Resource Policy Contributor						X	

Azure Role Name	Create	Rename	Move**	Delete	Assign Access	Assign Policy	Read
User Access Administrator			X		X		

*: The **Management Group Contributor** and **Management Group Reader** roles allow users to perform those actions only on the management group scope.

**: Role assignments on the root management group aren't required to move a subscription or management group to and from it.

See [Manage your resources with management groups](#) for details on moving items within the hierarchy.

Azure custom role definition and assignment

You can define a management group as an assignable scope in an Azure custom role definition. The Azure custom role will then be available for assignment on that management group and any management group, subscription, resource group, or resource under it. The custom role will inherit down the hierarchy like any built-in role. For information about the limitations with custom roles and management groups, see [Limitations](#).

Example definition

[Defining and creating a custom role](#) doesn't change with the inclusion of management groups. Use the full path to define the management group **/providers/Microsoft.Management/managementgroups/{groupId}**.

Use the management group's ID and not the management group's display name. This common error happens since both are custom-defined fields when creating a management group.

JSONCopy

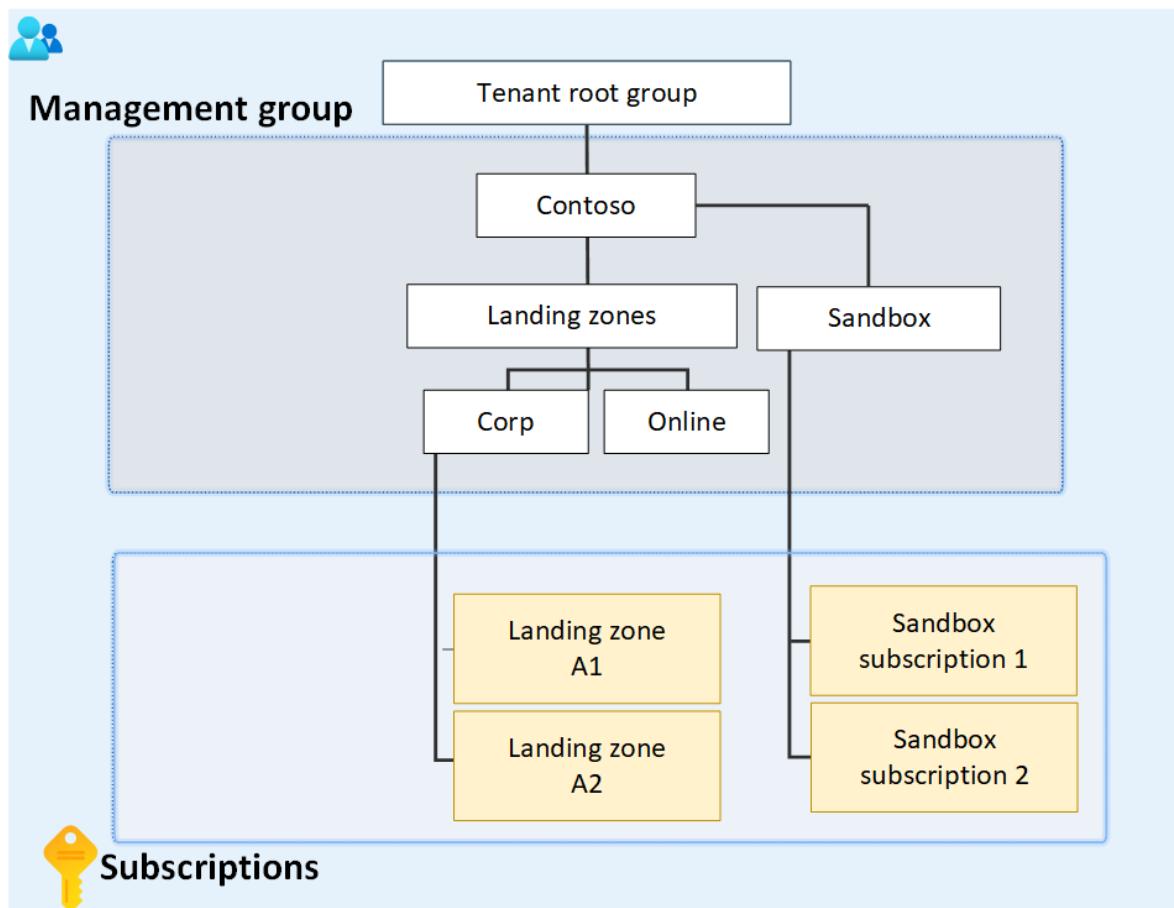
```
...
{
  "Name": "MG Test Custom Role",
  "Id": "id",
  "IsCustom": true,
  "Description": "This role provides members understand custom roles.",
  "Actions": [
    "Microsoft.Management/managementgroups/delete",
    "Microsoft.Management/managementgroups/read",
    "Microsoft.Management/managementgroup/write",
    "Microsoft.Management/managementgroup/subscriptions/delete",
    "Microsoft.Management/managementgroup/subscriptions/write",
    "Microsoft.resources/subscriptions/read",
```

```
"Microsoft.Authorization/policyAssignments/*",
"Microsoft.Authorization/policyDefinitions/*",
"Microsoft.Authorization/policySetDefinitions/*",
"Microsoft.PolicyInsights/*",
"Microsoft.Authorization/roleAssignments/*",
"Microsoft.Authorization/roleDefinitions/*"
],
"NotActions": [],
"DataActions": [],
"NotDataActions": [],
"AssignableScopes": [
    "/providers/microsoft.management/managementGroups/ContosoCorporate"
]
}
...
...
```

Issues with breaking the role definition and assignment hierarchy path

Role definitions are assignable scope anywhere within the management group hierarchy. A role definition can be defined on a parent management group while the actual role assignment exists on the child subscription. Since there's a relationship between the two items, you'll receive an error when trying to separate the assignment from its definition.

For example, let's look at a small section of a hierarchy for a visual.



The diagram focuses on the root management group with child Landing zones and Sandbox management groups. The Landing zones management group has two child management groups named Corp and Online while the Sandbox management group has two child subscriptions.

Let's say there's a custom role defined on the Sandbox management group. That custom role is then assigned on the two Sandbox subscriptions.

If we try to move one of those subscriptions to be a child of the Corp management group, this move would break the path from subscription role assignment to the Sandbox management group role definition. In this scenario, you'll receive an error saying the move isn't allowed since it will break this relationship.

There are a couple different options to fix this scenario:

- Remove the role assignment from the subscription before moving the subscription to a new parent MG.
- Add the subscription to the role definition's assignable scope.
- Change the assignable scope within the role definition. In the above example, you can update the assignable scopes from Sandbox to the root management group so that the definition can be reached by both branches of the hierarchy.

- Create another custom role that is defined in the other branch. This new role requires the role assignment to be changed on the subscription also.

Limitations

There are limitations that exist when using custom roles on management groups.

- You can only define one management group in the assignable scopes of a new role. This limitation is in place to reduce the number of situations where role definitions and role assignments are disconnected. This situation happens when a subscription or management group with a role assignment moves to a different parent that doesn't have the role definition.
- Resource provider data plane actions can't be defined in management group custom roles. This restriction is in place as there's a latency issue with updating the data plane resource providers. This latency issue is being worked on and these actions will be disabled from the role definition to reduce any risks.
- Azure Resource Manager doesn't validate the management group's existence in the role definition's assignable scope. If there's a typo or an incorrect management group ID listed, the role definition is still created.

Moving management groups and subscriptions

To move a management group or subscription to be a child of another management group, three rules need to be evaluated as true.

If you're doing the move action, you need:

- Management group write and role assignment write permissions on the child subscription or management group.
 - Built-in role example: **Owner**
- Management group write access on the target parent management group.
 - Built-in role example: **Owner, Contributor, Management Group Contributor**
- Management group write access on the existing parent management group.
 - Built-in role example: **Owner, Contributor, Management Group Contributor**

Exception: If the target or the existing parent management group is the root management group, the permissions requirements don't apply. Since the root management group is the default landing spot for all new management groups and subscriptions, you don't need permissions on it to move an item.

If the **Owner** role on the subscription is inherited from the current management group, your move targets are limited. You can only move the subscription to another management group where you have the **Owner** role. You can't move it to a

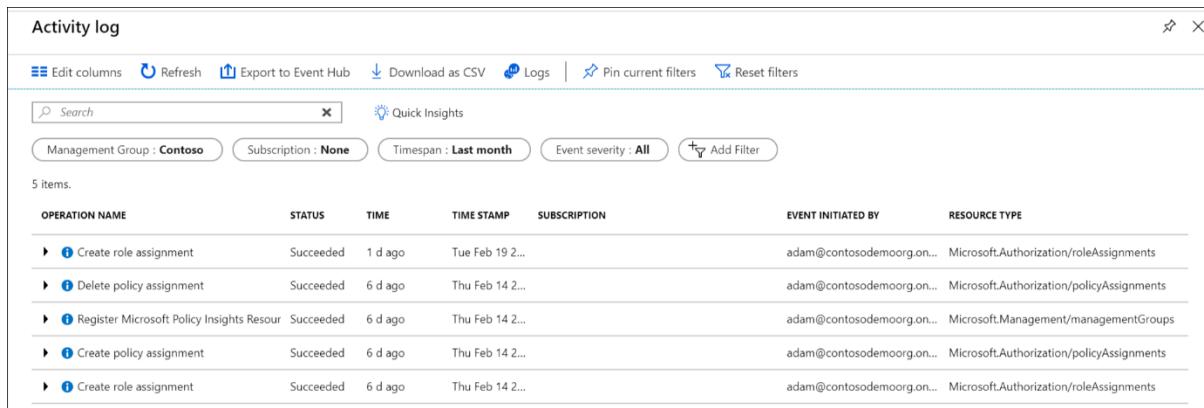
management group where you're a **Contributor** because you would lose ownership of the subscription. If you're directly assigned to the **Owner** role for the subscription (not inherited from the management group), you can move it to any management group where you're assigned the **Contributor** role.

Important

Azure Resource Manager caches management group hierarchy details for up to 30 minutes. As a result, moving a management group may not immediately be reflected in the Azure portal.

Audit management groups using activity logs

Management groups are supported within [Azure Activity log](#). You can search all events that happen to a management group in the same central location as other Azure resources. For example, you can see all role assignments or policy assignment changes made to a particular management group.



The screenshot shows the Azure Activity Log interface. At the top, there are various navigation and filter options: 'Edit columns', 'Refresh', 'Export to Event Hub', 'Download as CSV', 'Logs' (selected), 'Pin current filters', and 'Reset filters'. Below these are search and quick insights fields. The main area displays a table of events with the following columns: OPERATION NAME, STATUS, TIME, TIME STAMP, SUBSCRIPTION, EVENT INITIATED BY, and RESOURCE TYPE. The table shows five items related to management group operations:

OPERATION NAME	STATUS	TIME	TIME STAMP	SUBSCRIPTION	EVENT INITIATED BY	RESOURCE TYPE
>Create role assignment	Succeeded	1 d ago	Tue Feb 19 2...		adam@contosodemo.org.on...	Microsoft.Authorization/roleAssignments
Delete policy assignment	Succeeded	6 d ago	Thu Feb 14 2...		adam@contosodemo.org.on...	Microsoft.Authorization/policyAssignments
Register Microsoft Policy Insights Resour	Succeeded	6 d ago	Thu Feb 14 2...		adam@contosodemo.org.on...	Microsoft.Management/managementGroups
Create policy assignment	Succeeded	6 d ago	Thu Feb 14 2...		adam@contosodemo.org.on...	Microsoft.Authorization/policyAssignments
Create role assignment	Succeeded	6 d ago	Thu Feb 14 2...		adam@contosodemo.org.on...	Microsoft.Authorization/roleAssignments

When looking to query on management groups outside the Azure portal, the target scope for management groups looks like `"/providers/Microsoft.Management/managementGroups/{management-group-id}"`.

Manage your Azure subscriptions at scale with management groups

- Article
- 03/08/2023
- 12 contributors

Feedback

In this article

1. [Change the name of a management group](#)
2. [Delete a management group](#)
3. [View management groups](#)
4. [Moving management groups and subscriptions](#)

Show 5 more

If your organization has many 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. All subscriptions within a management group automatically inherit the conditions applied to the management group.

Management groups give you enterprise-grade management at a large scale no matter what type of subscriptions you might have. To learn more about management groups, see [Organize your resources with Azure management groups](#).

Note

This article provides steps about how to delete personal data from the device or service and can be used to support your obligations under the GDPR. For general information about GDPR, see the [GDPR section of the Microsoft Trust Center](#) and the [GDPR section of the Service Trust portal](#).

Important

Azure Resource Manager user tokens and management group cache lasts for 30 minutes before they are forced to refresh. After doing any action like moving a management group or subscription, it might take up to 30 minutes to show. To see the updates sooner you need to update your token by refreshing the browser, signing in and out, or requesting a new token.

Important

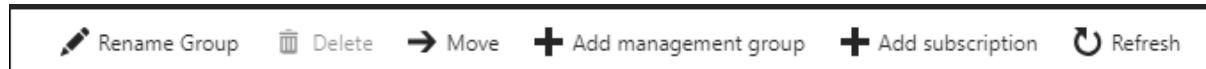
AzManagementGroup related Az PowerShell cmdlets mention that the **-GroupId** is alias of **-GroupName** parameter so we can use either of it to provide Management Group Id as a string value.

Change the name of a management group

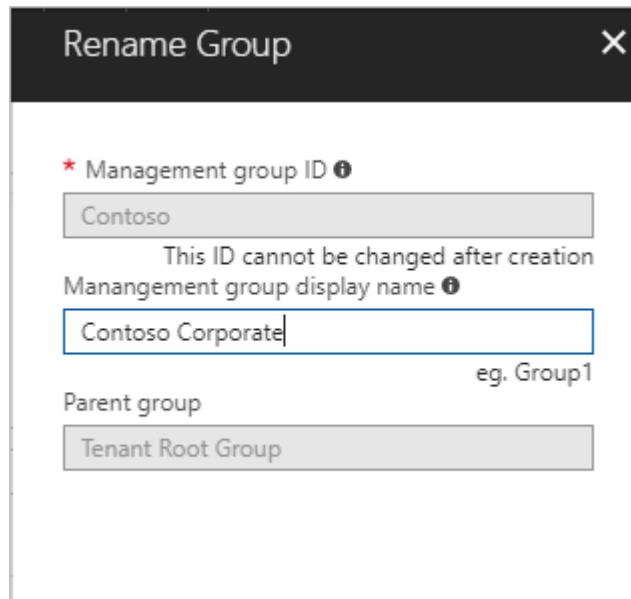
You can change the name of the management group by using the portal, PowerShell, or Azure CLI.

Change the name in the portal

1. Log into the [Azure portal](#).
2. Select **All services > Management groups**.
3. Select the management group you would like to rename.
4. Select **details**.
5. Select the **Rename group** option at the top of the page.



6. When the menu opens, enter the new name you would like to have displayed.



7. Select **Save**.

Change the name in PowerShell

To update the display name use **Update-AzManagementGroup**. For example, to change a management groups display name from "Contoso IT" to "Contoso Group", you run the following command:

Azure PowerShellCopy

Open Cloudshell

```
Update-AzManagementGroup -GroupId 'ContosoIt' -DisplayName 'Contoso Group'
```

Change the name in Azure CLI

For Azure CLI, use the update command.

Azure CLICopy

Open Cloudshell

```
az account management-group update --name 'Contoso' --display-name 'Contoso Group'
```

Delete a management group

To delete a management group, the following requirements must be met:

1. There are no child management groups or subscriptions under the management group. To move a subscription or management group to another management group, see [Moving management groups and subscriptions in the hierarchy](#).
2. You need write permissions on the management group ("Owner", "Contributor", or "Management Group Contributor"). To see what permissions you have, select the management group and then select **IAM**. To learn more on Azure roles, see [Azure role-based access control \(Azure RBAC\)](#).

Delete in the portal

1. Log into the [Azure portal](#).
2. Select **All services > Management groups**.
3. Select the management group you would like to delete.
4. Select **details**.
5. Select **Delete**

The screenshot shows the Azure portal's Management Groups blade. At the top, there are several buttons: 'Rename Group', 'Delete' (which is highlighted with a red box), 'Move', '+ Add management group', '+ Add subscription', and 'Refresh'. Below these buttons, there is a summary section with the following information:

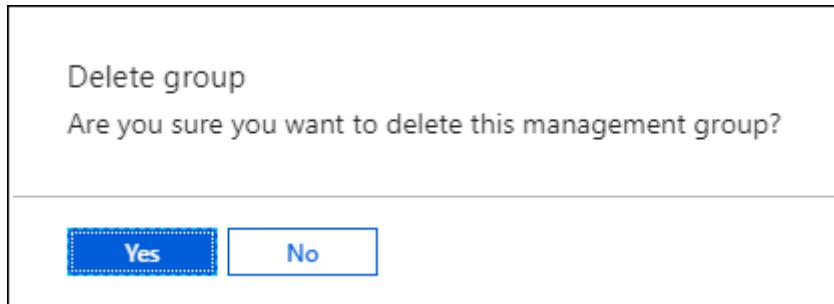
- Name: Newgroup1
- ID: Newgroup
- Access Level: Owner
- Parent management group: Contoso Redmond
- Child management groups: 0

Below this summary, there is a search bar labeled 'Search by name or ID'. Underneath the search bar, there is a table with two columns: 'NAME' and 'ID'. The table contains one row with the text 'No result'.

Tip

If the icon is disabled, hovering your mouse selector over the icon shows you the reason.

6. There's a window that opens confirming you want to delete the management group.



7. Select **Yes**.

Delete in PowerShell

Use the **Remove-AzManagementGroup** command within PowerShell to delete management groups.

Azure PowerShellCopy

Open Cloudshell

```
Remove-AzManagementGroup -GroupId 'Contoso'
```

Delete in Azure CLI

With Azure CLI, use the command az account management-group delete.

Azure CLICopy

Open Cloudshell

```
az account management-group delete --name 'Contoso'
```

View management groups

You can view any management group you have a direct or inherited Azure role on.

View in the portal

1. Log into the [Azure portal](#).
2. Select **All services > Management groups**.
3. The management group hierarchy page will load. This page is where you can explore all the management groups and subscriptions you have

access to. Selecting the group name takes you to a lower level in the hierarchy. The navigation works the same as a file explorer does.

4. To see the details of the management group, select the **(details)** link next to the title of the management group. If this link isn't available, you don't have permissions to view that management group.

The screenshot shows the Azure Management Groups blade. At the top, there's a search bar and a 'Management groups' section with a Microsoft logo. Below that is a navigation bar with 'Overview' (selected), 'Get Started', and 'Settings'. A help message says 'Click on a management group or subscription count to view and add a subscription'. There's also a search bar for 'Search by name or ID'. The main area displays 17 groups, with one expanded to show its children:

Management group	ID	Child Subscriptions	Total subscriptions
↳ [M] {ManagementGroup-GUID}	{ManagementGroup-GUID}	0	18
> [M] Contoso Marketing	ContosoMarketing	0	10
> [M] Contoso Storefront	ContosoStorefront	0	1
> [M] Contoso IT	ContosoIT	0	7

View in PowerShell

You use the `Get-AzManagementGroup` command to retrieve all groups.

See [Az.Resources](#) modules for the full list of management group GET PowerShell commands.

Azure PowerShellCopy

Open Cloudshell

[Get-AzManagementGroup](#)

For a single management group's information, use the `-GroupId` parameter

Azure PowerShellCopy

Open Cloudshell

[Get-AzManagementGroup -GroupId 'Contoso'](#)

To return a specific management group and all the levels of the hierarchy under it, use **-Expand** and **-Recurse** parameters.

Azure PowerShellCopy

Open Cloudshell

```
PS C:\> $response = Get-AzManagementGroup -GroupId TestGroupParent -Expand -Recurse
PS C:\> $response
```

```
Id          : /providers/Microsoft.Management/managementGroups/TestGroupParent
Type        : /providers/Microsoft.Management/managementGroups
Name        : TestGroupParent
TenantId    : 00000000-0000-0000-000000000000
DisplayName : TestGroupParent
UpdatedTime : 2/1/2018 11:15:46 AM
UpdatedBy   : 00000000-0000-0000-000000000000
ParentId    : /providers/Microsoft.Management/managementGroups/00000000-0000-0000-000000000000
ParentName   : 00000000-0000-0000-000000000000
ParentDisplayName : 00000000-0000-0000-000000000000
Children    : {TestGroup1DisplayName, TestGroup2DisplayName}
```

```
PS C:\> $response.Children[0]
```

```
Type      : /managementGroup
Id       : /providers/Microsoft.Management/managementGroups/TestGroup1
Name     : TestGroup1
DisplayName : TestGroup1DisplayName
Children  : {TestRecurseChild}
```

```
PS C:\> $response.Children[0].Children[0]
```

```
Type      : /managementGroup
Id       : /providers/Microsoft.Management/managementGroups/TestRecurseChild
Name     : TestRecurseChild
DisplayName : TestRecurseChild
Children  :
```

View in Azure CLI

You use the list command to retrieve all groups.

Azure CLICopy

Open Cloudshell

```
az account management-group list
```

For a single management group's information, use the show command

Azure CLICopy

Open Cloudshell

```
az account management-group show --name 'Contoso'
```

To return a specific management group and all the levels of the hierarchy under it, use **-Expand** and **-Recurse** parameters.

Azure CLICopy

Open Cloudshell

```
az account management-group show --name 'Contoso' -e -r
```

Moving management groups and subscriptions

One reason to create a management group is to bundle subscriptions together. Only management groups and subscriptions can be made children of another management group. A subscription that moves to a management group inherits all user access and policies from the parent management group

When moving a management group or subscription to be a child of another management group, three rules need to be evaluated as true.

If you're doing the move action, you need permission at each of the following layers:

- Child subscription / management group
 - Microsoft.management/managementgroups/write
 - Microsoft.management/managementgroups/subscriptions/write (only for Subscriptions)
 - Microsoft.Authorization/roleAssignments/write
 - Microsoft.Authorization/roleAssignments/delete
 - Microsoft.Management/register/action
- Target parent management group
 - Microsoft.management/managementgroups/write
- Current parent management group
 - Microsoft.management/managementgroups/write

Exception: If the target or the existing parent management group is the Root management group, the permissions requirements don't apply. Since the Root management group is the default landing spot for all new management groups and subscriptions, you don't need permissions on it to move an item.

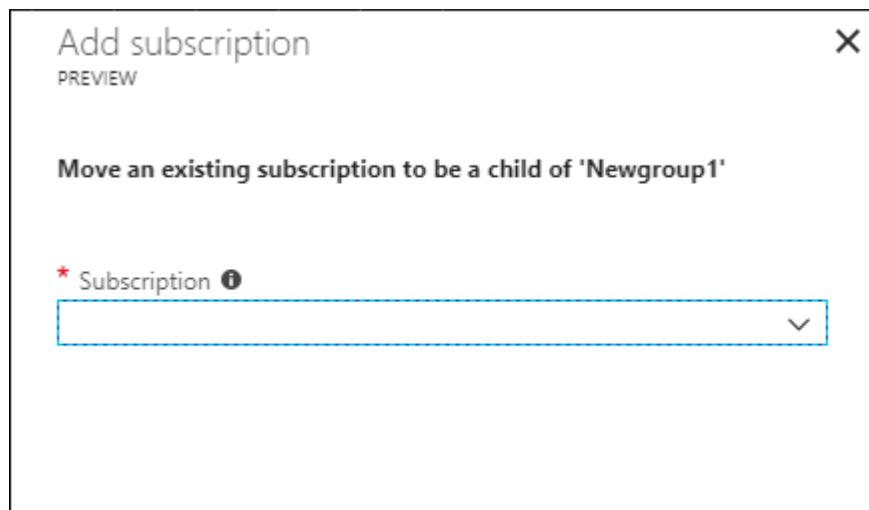
If the Owner role on the subscription is inherited from the current management group, your move targets are limited. You can only move the subscription to another management group where you have the Owner role. You can't move the subscription to a management group where you're only a contributor because you would lose ownership of the subscription. If you're directly assigned to the Owner role for the subscription, you can move it to any management group where you're a contributor.

To see what permissions you have in the Azure portal, select the management group and then select **IAM**. To learn more on Azure roles, see [Azure role-based access control \(Azure RBAC\)](#).

Move subscriptions

Add an existing Subscription to a management group in the portal

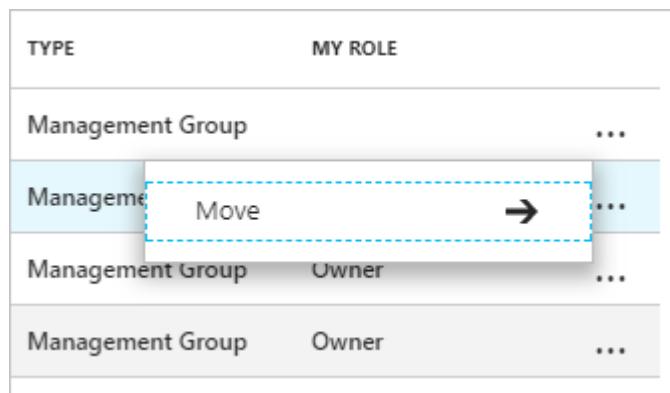
1. Log into the [Azure portal](#).
2. Select **All services > Management groups**.
3. Select the management group you're planning to be the parent.
4. At the top of the page, select **Add subscription**.
5. Select the subscription in the list with the correct ID.



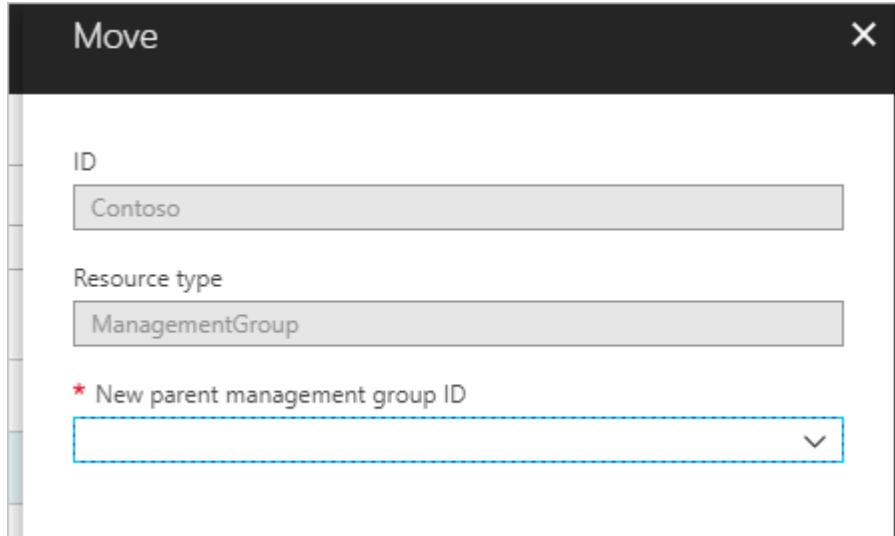
6. Select "Save".

Remove a subscription from a management group in the portal

1. Log into the [Azure portal](#).
2. Select **All services > Management groups**.
3. Select the management group you're planning that is the current parent.
4. Select the ellipsis at the end of the row for the subscription in the list you want to move.



5. Select **Move**.
6. On the menu that opens, select the **Parent management group**.



7. Select **Save**.

Move subscriptions in PowerShell

To move a subscription in PowerShell, you use the New-AzManagementGroupSubscription command.

Azure PowerShellCopy

Open Cloudshell

```
New-AzManagementGroupSubscription -GroupId 'Contoso' -SubscriptionId '12345678-1234-1234-1234-123456789012'
```

To remove the link between the subscription and the management group use the Remove-AzManagementGroupSubscription command.

Azure PowerShellCopy

Open Cloudshell

```
Remove-AzManagementGroupSubscription -GroupId 'Contoso' -SubscriptionId '12345678-1234-1234-1234-123456789012'
```

Move subscriptions in Azure CLI

To move a subscription in CLI, you use the add command.

Azure CLICopy

Open Cloudshell

```
az account management-group subscription add --name 'Contoso' --subscription '12345678-1234-1234-1234-123456789012'
```

To remove the subscription from the management group, use the subscription remove command.

Azure CLICopy

Open Cloudshell

```
az account management-group subscription remove --name 'Contoso' --subscription '12345678-1234-1234-1234-123456789012'
```

Move subscriptions in ARM template

To move a subscription in an Azure Resource Manager template (ARM template), use the following template and deploy it at [tenant level](#).

JSONCopy

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-08-01/managementGroupDeploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "targetMgId": {
            "type": "string",
            "metadata": {
                "description": "Provide the ID of the management group that you want to move the subscription to."
            }
        },
        "subscriptionId": {
            "type": "string",
            "metadata": {
                "description": "Provide the ID of the existing subscription to move."
            }
        }
    },
    "resources": [
        {
            "scope": "/",
            "type": "Microsoft.Management/managementGroups/subscriptions",
            "apiVersion": "2020-05-01",
            "name": "[concat(parameters('targetMgId'), '/', parameters('subscriptionId'))]",
            "properties": {}
        }
    ],
    "outputs": {}
}
```

Or, the following Bicep file.

BicepCopy

```
targetScope = 'managementGroup'

@description('Provide the ID of the management group that you want to move the
subscription to.')
param targetMgId string

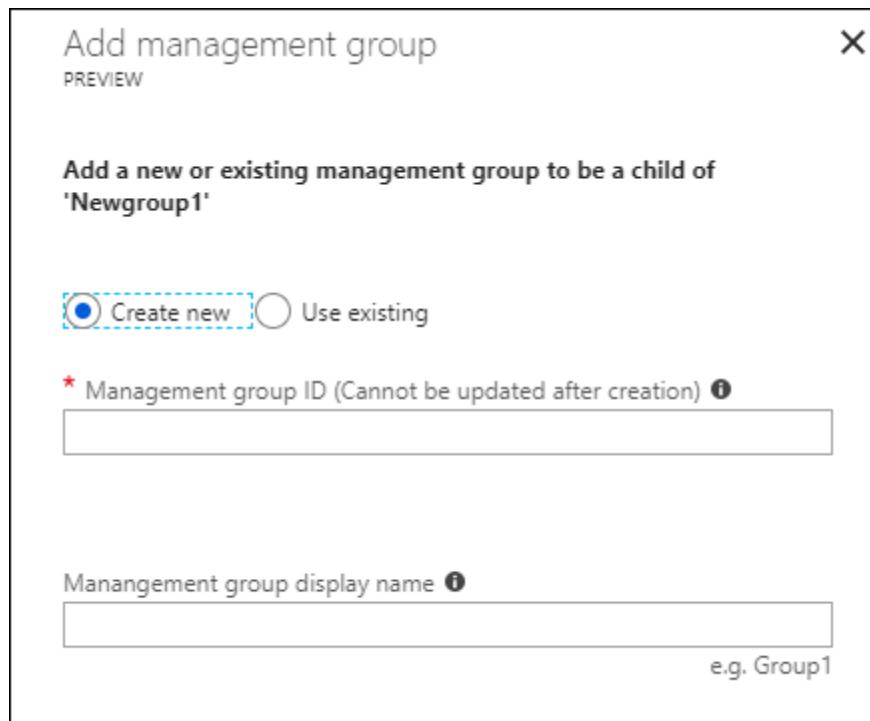
@description('Provide the ID of the existing subscription to move.')
param subscriptionId string

resource subToMG 'Microsoft.Management/managementGroups/subscriptions@2020-05-01'
= {
  scope: tenant()
  name: '${targetMgId}/${subscriptionId}'
}
```

Move management groups

Move management groups in the portal

1. Log into the [Azure portal](#).
2. Select **All services > Management groups**.
3. Select the management group you're planning to be the parent.
4. At the top of the page, select **Add management group**.
5. In the menu that opens, select if you want a new or use an existing management group.
 - Selecting new will create a new management group.
 - Selecting an existing will present you with a dropdown list of all the management groups you can move to this management group.



6. Select **Save**.

Move management groups in PowerShell

Use the `Update-AzManagementGroup` command in PowerShell to move a management group under a different group.

Azure PowerShellCopy

Open Cloudshell

```
$parentGroup = Get-AzManagementGroup -GroupId ContosoIT
Update-AzManagementGroup -GroupId 'Contoso' -ParentId $parentGroup.id
```

Move management groups in Azure CLI

Use the `update` command to move a management group with Azure CLI.

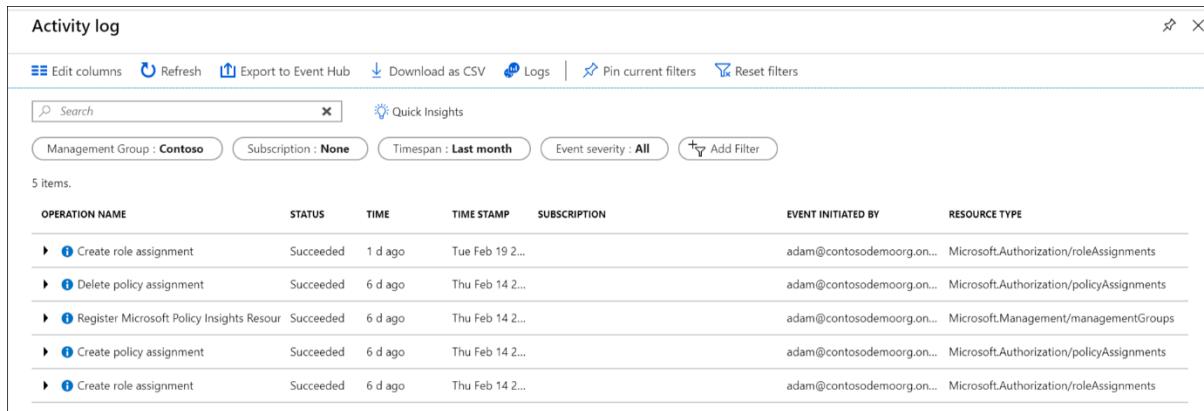
Azure CLICopy

Open Cloudshell

```
az account management-group update --name 'Contoso' --parent ContosoIT
```

Audit management groups using activity logs

Management groups are supported within [Azure Activity Log](#). You can query all events that happen to a management group in the same central location as other Azure resources. For example, you can see all Role Assignments or Policy Assignment changes made to a particular management group.



The screenshot shows the Azure Activity Log interface. At the top, there are buttons for 'Edit columns', 'Refresh', 'Export to Event Hub', 'Download as CSV', 'Logs', 'Pin current filters', and 'Reset filters'. Below these are search and filter fields: 'Search' (with a clear button), 'Quick Insights', 'Management Group : Contoso', 'Subscription : None', 'Timespan : Last month', 'Event severity : All', and a 'Add Filter' button. A message indicates '5 items.' Below this is a table with the following columns: OPERATION NAME, STATUS, TIME, TIME STAMP, SUBSCRIPTION, EVENT INITIATED BY, and RESOURCE TYPE. The table lists five events:

OPERATION NAME	STATUS	TIME	TIME STAMP	SUBSCRIPTION	EVENT INITIATED BY	RESOURCE TYPE
>Create role assignment	Succeeded	1 d ago	Tue Feb 19 2...	adam@contosodemo.org.on...	Microsoft.Authorization/roleAssignments	
Delete policy assignment	Succeeded	6 d ago	Thu Feb 14 2...	adam@contosodemo.org.on...	Microsoft.Authorization/policyAssignments	
Register Microsoft Policy Insights Resour	Succeeded	6 d ago	Thu Feb 14 2...	adam@contosodemo.org.on...	Microsoft.Management/managementGroups	
Create policy assignment	Succeeded	6 d ago	Thu Feb 14 2...	adam@contosodemo.org.on...	Microsoft.Authorization/policyAssignments	
Create role assignment	Succeeded	6 d ago	Thu Feb 14 2...	adam@contosodemo.org.on...	Microsoft.Authorization/roleAssignments	

When looking to query on Management Groups outside of the Azure portal, the target scope for management groups looks like **"/providers/Microsoft.Management/managementGroups/{yourMgID}"**.

Referencing management groups from other Resource Providers

When referencing management groups from other Resource Provider's actions, use the following path as the scope. This path is used when using PowerShell, Azure CLI, and REST APIs.

`/providers/Microsoft.Management/managementGroups/{yourMgID}`

An example of using this path is when assigning a new role assignment to a management group in PowerShell:

Azure PowerShellCopy

Open Cloudshell

```
New-AzRoleAssignment -Scope  
"/providers/Microsoft.Management/managementGroups/Contoso"
```

The same scope path is used when retrieving a policy definition at a management group.

HTTPCopy

GET

<https://management.azure.com/providers/Microsoft.Management/managementgroups/MyMan>

How to protect your resource hierarchy

- Article
- 08/18/2021
- 6 contributors

Feedback

In this article

1. [Azure RBAC permissions for hierarchy settings](#)
2. [Setting - Default management group](#)
3. [Setting - Require authorization](#)
4. [PowerShell sample](#)
5. [Next steps](#)

Your resources, resource groups, subscriptions, management groups, and tenant collectively make up your resource hierarchy. Settings at the root management group, such as Azure custom roles or Azure Policy policy assignments, can impact every resource in your resource hierarchy. It's important to protect the resource hierarchy from changes that could negatively impact all resources.

Management groups now have hierarchy settings that enable the tenant administrator to control these behaviors. This article covers each of the available hierarchy settings and how to set them.

Azure RBAC permissions for hierarchy settings

Configuring any of the hierarchy settings requires the following two resource provider operations on the root management group:

- Microsoft.Management/managementgroups/settings/write
- Microsoft.Management/managementgroups/settings/read

These operations only allow a user to read and update the hierarchy settings. The operations don't provide any other access to the management group hierarchy or resources in the hierarchy. Both of these operations are available in the Azure built-in role **Hierarchy Settings Administrator**.

Setting- Default management group

By default, a new subscription added within a tenant is added as a member of the root management group. If policy assignments, Azure role-based access control (Azure RBAC), and other governance constructs are assigned to the root management group, they immediately effect these new subscriptions. For this reason, many organizations don't apply these constructs at the root management group even though that is the desired place to assign

them. In other cases, a more restrictive set of controls is desired for new subscriptions, but shouldn't be assigned to all subscriptions. This setting supports both use cases.

By allowing the default management group for new subscriptions to be defined, organization-wide governance constructs can be applied at the root management group, and a separate management group with policy assignments or Azure role assignments more suited to a new subscription can be defined.

Set default management group in portal

To configure this setting in the Azure portal, follow these steps:

1. Use the search bar to search for and select 'Management groups'.
2. On the root management group, select **details** next to the name of the management group.
3. Under **Settings**, select **Hierarchy settings**.
4. Select the **Change default management group** button.

Note

If the **Change default management group** button is disabled, either the management group being viewed isn't the root management group or your security principal doesn't have the necessary permissions to alter the hierarchy settings.

5. Select a management group from your hierarchy and use the **Select** button.

Set default management group with REST API

To configure this setting with REST API, the [Hierarchy Settings](#) endpoint is called. To do so, use the following REST API URI and body format. Replace `{rootMgID}` with the ID of your root management group and `{defaultGroupID}` with the ID of the management group to become the default management group:

- REST API URI

HTTPCopy

PUT
`https://management.azure.com/providers/Microsoft.Management/managementGroups/{rootMgID}/settings/default?api-version=2020-05-01`

- Request Body

JSONCopy

```
{  
  "properties": {  
    "defaultManagementGroup":  
      "/providers/Microsoft.Management/managementGroups/{defaultGroupID}"  
  }  
}
```

```
    }  
}
```

To set the default management group back to the root management group, use the same endpoint and set **defaultManagementGroup** to a value of `/providers/Microsoft.Management/managementGroups/{rootMgID}`.

Setting- Require authorization

Any user, by default, can create new management groups within a tenant. Admins of a tenant may wish to only provide these permissions to specific users to maintain consistency and conformity in the management group hierarchy. If enabled, a user requires the `Microsoft.Management/managementGroups/write` operation on the root management group to create new child management groups.

Set require authorization in portal

To configure this setting in the Azure portal, follow these steps:

1. Use the search bar to search for and select 'Management groups'.
2. On the root management group, select **details** next to the name of the management group.
3. Under **Settings**, select **Hierarchy settings**.
4. Toggle the **Require permissions for creating new management groups**. option to on.

Note

If the **Require permissions for creating new management groups**. toggle is disabled, either the management group being viewed isn't the root management group or your security principal doesn't have the necessary permissions to alter the hierarchy settings.

Set require authorization with REST API

To configure this setting with REST API, the [Hierarchy Settings](#) endpoint is called. To do so, use the following REST API URI and body format. This value is a *boolean*, so provide either **true** or **false** for the value. A value of **true** enables this method of protecting your management group hierarchy:

- REST API URI

HTTPCopy

PUT

`https://management.azure.com/providers/Microsoft.Management/managementGroups/{rootMgID}/settings/default?api-version=2020-05-01`

- Request Body

JSONCopy

```
{  
    "properties": {  
        "requireAuthorizationForGroupCreation": true  
    }  
}
```

To turn the setting back off, use the same endpoint and set **requireAuthorizationForGroupCreation** to a value of **false**.

PowerShell sample

PowerShell doesn't have an 'Az' command to set the default management group or set require authorization, but as a workaround you can use the REST API with the PowerShell sample below:

PowerShellCopy

```
$root_management_group_id = "Enter the ID of root management group"  
$default_management_group_id = "Enter the ID of default management group (or use  
the same ID of the root management group)"  
  
$body = '{  
    "properties": {  
        "defaultManagementGroup":  
        "/providers/Microsoft.Management/managementGroups/" + $default_management_group_id  
        + '",  
        "requireAuthorizationForGroupCreation": true  
    }  
}'  
  
$token = (Get-AzAccessToken).Token  
$headers = @{"Authorization"= "Bearer $token"; "Content-Type"= "application/json"}  
$uri =  
"https://management.azure.com/providers/Microsoft.Management/managementGroups/$roo  
t_management_group_id/settings/default?api-version=2020-05-01"  
  
Invoke-RestMethod -Method PUT -Uri $uri -Headers $headers -Body $body
```

Next steps

To learn more about management groups, see:

- [Create management groups to organize Azure resources](#)
- [How to change, delete, or manage your management groups](#)

Azure Cosmos DB

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Hello and welcome to Azure Database Analytics and Compute Services.

In this module, you'll learn about several of the primary database services that are available on Azure.

You'll analyze some of the reasons why each of these database services might be the right choice for your needs.

In addition, you look at the big data and analysis services in Azure.

You'll also examine how to take advantage off several virtualization services in Azure Compute, which can help your applications scale out quickly on efficiently to meet increasing demands.

Let's take a look at our case study to see what is expected from you as the I T specialist of Tailwind traders.

Due to a growing number of acquisitions over the last decade, Tailwind traders uses a variety of database and analytics technologies.

As the company begins to migrate existing data workloads and deploy new data workloads to Azure, it needs to understand which Azure technology is the most appropriate for each workload.

The company's Chief Technology Officer has assigned you the task of researching the different database options that are available.

This will help tailwind traders choose the right options for each of their data scenarios.

In this lesson, you'll focus on several of the database services that are available on Microsoft Azure.

Such as Azure Cosmos DB, Azure SQL Database and Azure SQL Managed Instance.

Azure Database for MySQL and Azure Database for PostgreSQL.

In addition, you'll learn about several of the Big data and

Analysis services in Azure.

So let's dive right in and find out how you can help Tailwind traders.

Play video starting at :1:45 and follow transcript1:45

Over the years, Tailwind traders has acquired several smaller companies.

Each of these companies had teams of developers who use different database services on various APIs to work with their data.

The long term plan is to move all of the disparate data to a common database service.

For now though, you'd like to enable each of these teams to work with an environment where they can use their existing skills.

Fortunately for you, Azure Cosmos DB can help out.

Azure cosmos DB is a globally distributed multi model database service.

Using Azure Cosmos DB, you can elastically and independently scale throughput and storage across any number of Azure regions worldwide.

And take advantage of fast single digit millisecond data access by using any one of several popular APIs.

The Azure cosmos DB Service offers comprehensive service level agreements, which cover the guarantees for throughput, consistency, availability and lightensy.

It's important to note that servers are still running the code.

Play video starting at :2:55 and follow transcript2:55

Azure Cosmos DB also support schemaless data which lets you build highly responsive and always on applications to support constantly changing data.

You can use this feature to store data that's updated and maintained by users around the world.

Now let's take a look at an example of how Tailwind traders use Azure cosmos DB.

Tailwind Traders provides a public training portal that is used by customers across the globe to learn about the different tools that Tailwind traders creates.

Tailwind traders developers maintain and update the data.

This image shows a sample Azure Cosmos DB database that's used to store data for the Tailwind Traders Training Portal website.

At the lowest level,

Azure Cosmos DB stores Data in atom record sequence, ARS format.

The data is then abstracted and projected as an API, which you specify when you're creating your database.

Your choices include SQL, MongoDB, Cassandra, Tables and Gremlin.

Play video starting at :4:1 and follow transcript4:01

Now let's take a look at some of the key benefits you can provide to the CTO at Tailwind Traders.

Azure Cosmos DB is a flexible database that provides guaranteed single digit milliseconds response times and 99.999% availability backed by comprehensive SLAs.

Elastic and independent scale throughput and storage on demand, access to multiple data models and APIs for working with your data.

And the ability to globally distribute your data and build highly responsive applications.

Azure SQL Database

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

In this session, you will explore Azure SQL Database as a possible solution for Tailwind Traders.

So what is Azure SQL Database?

Azure SQL Database is a relational database based on the latest stable version of the Microsoft SQL Server database engine.

Azure SQL Database provides you with a high performance, reliable, fully managed and secure database.

You can use it to build data driven applications and websites in the programming language of your choice without needing to manage infrastructure.

Let's take a look at some of the features provided by Azure SQL Database.

Play video starting at ::38 and follow transcript0:38

Azure SQL Database is a platform as a service, PaaS database engine.

It handles most of the database management functions,

such as upgrading, patching, backups and monitoring without user involvement.

SQL database also provides 99.99 percent availability.

With Azure SQL Database, you can create a highly available and high performance data storage layer for the applications and solutions in Azure.

Microsoft handles all updates to the SQL on operating system code.

You don't have to manage the underlying infrastructure.

In a nutshell, Azure SQL Database is a fully managed service that has built in high availability, backups and other common maintenance operations.

The PaaS capabilities that are built into SQL Database enable you to focus on the domain specific database administration and optimization activities that are critical for your business.

Azure SQL Database can be the right choice for a variety of modern cloud applications because it enables you to process both relational data and non relational structures such as graphs, JSON, Spatial and XML.

You can also use advanced query processing features such as high performance in memory technologies and intelligent query processing.

In fact, the newest capabilities of SQL Server are released first to SQL Database and then to SQL Server itself.

You get the newest SQL Server capabilities with no overhead for updates or upgrades tested across millions of databases.

Tailwind Traders currently uses several on premises servers running SQL Server, which provide data storage for your public facing website, for example, customer data, order history and product catalogs.

In addition, your on premises servers running SQL Server also provide data storage for your internal only training portal website.

Tailwind Traders uses the website for new employees training materials, such as study materials, certification details and training transcripts.

This image illustrates the types of data that Tailwind Traders might store in the Azure SQL Database Training Portal website.

You can migrate your existing SQL Server databases with minimal downtime by using the Azure Database Migration Service.

After you assess and resolve any remediation required, you're ready to begin the migration process.

The Azure Database Migration Service performs all of the required steps, you just change the connection string in your apps.

Azure SQL Managed Instance

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Azure SQL Managed Instance is
a scalable cloud data service that provides
the broadest SQL Server
Database Engine compatibility with
all the benefits of
a fully managed platform as a service.

Depending on your scenario,
Azure SQL Managed Instance
might offer more options for your database needs.

Like Azure SQL Database,
Azure SQL Managed Instance is
a platform as a service Database Engine,
which means that your company
will be able to take advantage of
the best features of moving
your data to the cloud in a fully managed environment.

Let's take a look at an example of
the benefits for Tailwind Traders.

Using a fully managed environment,
Tailwind Traders will no longer need to
purchase and manage expensive hardware
and won't have to maintain
the additional overheads of
managing on-premises infrastructure.

On the other hand, Tailwind Traders will

benefit from the quick provisioning and services scaling features of Azure together with automated patching and version upgrades.

In addition, Tailwind Traders can rest assured that their data will always be there when they need it through built-in high availability features and a 99.99 percent uptime service level agreement.

They'll also be able to protect their data with automated backups and a configurable backup retention period.

Azure SQL Database and Azure SQL Managed Instance offer many of the same features.

However, Azure SQL Managed Instance provides several options that might not be available to Azure SQL Database.

In this example, Tailwind Traders currently uses several on-premises servers running SQL Server and they would like to migrate their existing databases to a SQL database running in the cloud.

However, several of their databases use Cyrillic characters for collation.

In this scenario, Tailwind Traders should migrate their databases to an Azure SQL Managed Instance.

One of the features of Azure SQL Managed Instance is that server-level collation could be specified when the instance is created.

On the other hand, Azure SQL Database only uses the default SQL_Latin1_General_CI_AS

Play video starting at :2:16 and follow transcript2:16 server collation.

It's important to note that although you can change server-level collation in SQL Managed Instance,

it cannot be changed once the instance has been created.

Azure SQL Managed Instance makes it easy to migrate your on-premises data on SQL Server to the client using the Azure Database Migration Service, DMS, or native backup and restore.

This image illustrates the migration process flow.

Let's go through this.

After you've discovered all of the features that your company uses, you need to assess which on-premises SQL Server Instances you can migrate to Azure SQL Managed Instance to see if you have any blocking issues. Once you have resolved any issues, you can migrate your data, then cut over from your on-premises SQL Server to your Azure SQL Managed Instance by changing the connection string in your applications.

Azure Database for MySQL

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

As part of your planning for your migration strategy, the different teams at Tailwind Traders have been researching the available service offerings that Azure provides.

You've been tasked with investigating whether the database requirements for

the web development team will continue to be met after the migration to Azure.

Tailwind Traders currently manages several websites on-premises that use the lamp stack. Linux, Apache, MySQL, PHP.

You'll discover that the web apps feature of the Azure App Service provides built-in functionality to create web applications that use PHP on a Linux server running Apache.

Now, let's explore Azure Database for MySQL.

Azure Database for MySQL is a relational database service in the cloud, and it's based on the MySQL community edition database engine.

With it, you have a 99.99 percent availability service level agreement from Azure, powered by a global network of Microsoft managed data centers.

This helps keep your app running 24/7.

With every Azure Database for MySQL server, you take advantage of built-in security, fault tolerance, and data protection that you would otherwise have to buy or design, build, and manage.

With Azure Database for MySQL, you can use point-in-time restore to recover a server to an earlier state as far back as 35 days.

Azure Database for MySQL delivers built-in high availability with no additional cost, predictable performance, and inclusive pay-as-you-go pricing, scale is needed within seconds, ability to protect sensitive data at rest and in motion, automatic backups,

and enterprise-grade security, and compliance.

These capabilities require almost no administration, and all are provided at no additional cost.

They allow you to focus on rapid app development and accelerating your time-to-market, rather than having to manage

Virtual Machines and Infrastructure.

In addition, you can migrate your existing MySQL databases with minimal downtime by using the Azure Database Migration Service.

After you've completed your migration, you can continue to develop your application with the open source tools and platform of your choice.

You don't have to learn new skills.

Azure database for MySQL offers several service tiers, and each tier provides different performance and capabilities to support lightweight to heavyweight database workloads.

You can build your first app on a small database for a few dollars a month, and then adjust the scale to meet the needs of your solution.

Dynamic scalability enables your database to transparently respond to rapidly changing resource requirements.

You only pay for the resources you need, and only when you need them.

Now that you've explored the Azure database for MySQL, let's go through some of the key findings in your investigation.

By using Azure Database MySQL, Tailwind Traders will be able to focus on rapid app development,

and accelerating time-to-market, rather than having to manage Virtual Machines and Infrastructure. Develop applications with the open source tools using a platform of their choice. Deliver with speed and efficiency without having to learn new skills. Use built-in features such as automated patching, high availability, automated backups, elastic scaling, enterprise-grade security, compliance and governance, monitoring and alerting, and pay only for what is used, with options to scale up or scale [inaudible] for greater control with no interruption.

Azure Database for PostgreSQL

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

As part of its overall data strategy, Tailwind Traders has been using PostgreSQL for several years.

You and your team probably already know the benefits of PostgreSQL. Part of your migration is to use Azure database for PostgreSQL, and you want to make sure that you'll have access to the same benefits as

your on-premises server before moving to the cloud.

Let's take a closer look at

Azure database for PostgreSQL.

Azure database for PostgreSQL

is a relational database service in the cloud.

The server's software is based on

the community version of

the open-source PostgreSQL database engine.

Your familiarity with the tools and expertise with

PostgreSQL is applicable when

you're using Azure database for PostgreSQL.

Let's discuss the key benefits

of Azure database for PostgreSQL.

Azure database for PostgreSQL

delivers the following benefits.

Built-in high availability

compared to on-premises resources.

There's no additional configuration, replication,

or cost required to make sure

your applications are always available.

Simple and flexible pricing,

you have predictable performance based on

a selected pricing tier choice

that includes software patching,

automatic backups, monitoring, and security.

Scale up or down as needed,

within seconds, you can scale, compute,

or storage independently as needed to make

sure you adapt your service to much usage.

Adjustable, automatic backups and

point-in-time restore for up to 35 days,

and enterprise-grade security and compliance

to protect sensitive data at rest and in motion.

This security covers Data Encryption on disk,

and SSL encryption between

client and server communication.

Azure database for PostgreSQL

is available in two deployment options,

Single Server and Hyperscale Citus.

First, let's take a look at

the Single Server option capabilities.

The single server option delivers

built-in high availability with

no additional cost and a

99.99% uptime service level agreement.

Predictable performance and

inclusive pay-as-you-go pricing.

Vertical scale is needed within seconds.

Enterprise-grade security and compliance,

monitoring and alerting to assess your server,

ability to protect sensitive data at rest and in motion,

and automatic backups and

point-in-time restore for up to 35 days.

These capabilities require almost no administration

and all are provided at no additional cost.

They allow you to focus on

rapid up development and

accelerating your time-to-market,

rather than having to manage

Virtual Machines and infrastructure.

You can continue to develop your application with

the open-source tools and platform of

your choice without having to learn new skills.

The single server option offers three pricing tiers,

basic, general-purpose, and memory-optimized.

Each tier offers different resource capabilities

to support your database workloads.

You can build your first app on a small database for

a few dollars a month and then adjust

the scale to meet the needs of your solution.

Dynamic scalability enables your database to transparently respond to rapidly changing resource requirements.

You only pay for the resources you need and only when you need them.

Now, let's explore the second PostgreSQL deployment option, Hyperscale Citus.

The Hyperscale Citus option horizontally scales queries across multiple machines by using sharding.

Its query engine parallelizes incoming sequel queries across these servers for faster responses on large data sets.

It serves applications that require greater scale and performance.

Generally, workloads that are approaching or already exceed 100 gigabytes of data.

The Hyperscale Citus deployment option supports multitenant applications, real-time operational analytics, and high-throughput transactional workloads.

Applications built for PostgreSQL can run distributed queries on Hyperscale Citus with standard connection libraries and minimal changes.

Thinking on Tailwind Traders' overall migration strategy, you can quickly and easily develop applications using Azure Database for PostgreSQL.

Tailwind Traders will also be able to use native PostgreSQL tools, drivers, and libraries, without worrying about having to manage and administrate the instances themselves.

Big data and analytics

[Save note](#)
[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Several years ago, Tailwind Traders rolled out
a new GPS tracking system
for all of its delivery vehicles.

The new system provides
real-time tracking data to your primary data center.

Your CTO wants your team to look at
several years of tracking data
in order to determine trends.

For example, an important trend might
be a spike in deliveries around the holidays,
that would require hiring additional staff.

Through an in-depth analysis
of the tracking data that you've recorded,
your CTO seeks to predict when changes are necessary and
proactively take the steps that are
necessary to manage spikes appropriately.

In this Tailwind Traders scenario,
data is collected from the GPS sensors,
which includes location information,
data from weather systems,
and many other sources that
generate large amounts of data.

This amount of data becomes increasingly
hard to make sense of and to base decisions on.
The volumes are so large that

traditional forms of processing and
analysis are no longer appropriate.

Data comes in all forms and formats.

When we talk about big data,

we're referring to large volumes of data.

Open-source cluster technologies have been developed over

time to try to deal with these large datasets.

In this session, we will

explore Microsoft Azure's broad range of

technologies and services that

provide big data and analytic solutions,

including Azure Synapse Analytics,

Azure HDInsight, Azure Databricks,

and Azure Data Lake Analytics.

Let's start exploring these solutions.

Azure Synapse Analytics,

formerly Azure SQL Data Warehouse,

is a limitless analytics service that brings together

enterprise data warehousing and big data analytics.

You can query data on your terms by using

either serverless or provisioned resources at scale.

With Azure Synapse Analytics,

you have a unified experience to ingest,

prepare, manage, and serve data

for immediate via machine learning needs.

Azure HDInsight is

a fully managed open-source

analytics service for enterprises.

It's a Cloud service that makes it easier, faster,

and more cost-effective to

process massive amounts of data.

You can run popular open-source frameworks and

create cluster types such as Apache Spark,

Apache Hadoop, Apache Kafka,

Apache HBase, Apache Storm,

and Machine Learning Services.

HDInsight also supports a broad range

of scenarios such as extraction,
transformation, and loading, ETL,
data warehousing, machine learning and IoT.

Azure Databricks helps you unlock insights from
all your data and build
artificial intelligence solutions.

You can set up your Apache Spark environment in minutes,
and then autoscale and collaborate on
shared projects in an interactive workspace.

Azure Databricks supports Python,
Scala, R, Java, and SQL,
as well as data science frameworks and libraries
including TensorFlow, PyTorch, and scikit-learn.

Azure Data Lake Analytics is
an on-demand analytics job service
that simplifies big data.

Instead of deploying, configuring, and tuning hardware,
you write queries to transform
your data and extract valuable insights.

The analytics service can handle jobs of any scale
instantly by setting the dial
for how much power you need.

With Azure Data Lake Analytics,
you only pay for your job when it's running,
making it more cost-effective.

Congratulations.

You have completed this lesson
on Azure Database fundamentals.

In this lesson, you explored several of
the database services that are
available on Microsoft Azure.

In addition, you learned how you can use big data and
analysis services like Azure Synapse Analytics,
Azure HDInsight, Azure Databricks,
and Azure Data Lake Analytics

to analyze large volumes of data.

By now you should be able to describe the benefits and usage of Azure Cosmos DB, Azure SQL Database, and SQL Managed Instance, Azure Database for MySQL and PostgreSQL, and Azure Synapse Analytics, HDInsight, and Azure Databricks.

What is Azure SQL?

- Article
- 04/25/2023
- 17 contributors

Feedback

In this article

1. [Overview](#)
2. [Service comparison](#)
3. [Cost](#)
4. [Administration](#)

Show 4 more

Applies to: Azure SQL Database Azure SQL Managed Instance SQL Server on Azure VM

Azure SQL is a family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud.

- **Azure SQL Database:** Support modern cloud applications on an intelligent, managed database service that includes serverless compute.
- **Azure SQL Managed Instance:** Modernize your existing SQL Server applications at scale with an intelligent fully managed instance as a service, with almost 100% feature parity with the SQL Server database engine. Best for most migrations to the cloud.
- **SQL Server on Azure VMs:** Lift-and-shift your SQL Server workloads with ease and maintain 100% SQL Server compatibility and operating system-level access.

Azure SQL is built upon the familiar SQL Server engine, so you can migrate applications with ease and continue to use the tools, languages, and resources you're familiar with. Your skills and experience transfer to the cloud, so you can do even more with what you already have.

Learn how each product fits into Microsoft's Azure SQL data platform to match the right option for your business requirements. Whether you prioritize cost savings or minimal administration, this article can help you decide which approach delivers against the business requirements you care about most.

If you're new to Azure SQL, check out the *What is Azure SQL* video from our in-depth [Azure SQL video series](#):

Overview

In today's data-driven world, driving digital transformation increasingly depends on our ability to manage massive amounts of data and harness its potential. But today's data estates are increasingly complex, with data hosted on-premises, in the cloud, or at the edge of the network. Developers who are building intelligent and immersive applications can find themselves constrained by limitations that can ultimately impact their experience. Limitations arising from incompatible platforms, inadequate data security, insufficient resources and price-performance barriers create complexity that can inhibit app modernization and development.

One of the first things to understand in any discussion of Azure versus on-premises SQL Server databases is that you can use it all. Microsoft's data platform leverages SQL Server technology and makes it available across physical on-premises machines, private cloud environments, third-party hosted private cloud environments, and the public cloud.

Fully managed and always up to date

Spend more time innovating and less time patching, updating, and backing up your databases. Azure is the only cloud with evergreen SQL that automatically applies the latest updates and patches so that your databases are always up to date—eliminating end-of-support hassle. Even complex tasks like performance tuning, high availability, disaster recovery, and backups are automated, freeing you to focus on applications.

Protect your data with built-in intelligent security

Azure constantly monitors your data for threats. With Azure SQL, you can:

- Remediate potential threats in real time with intelligent [advanced threat detection](#) and proactive vulnerability assessment alerts.

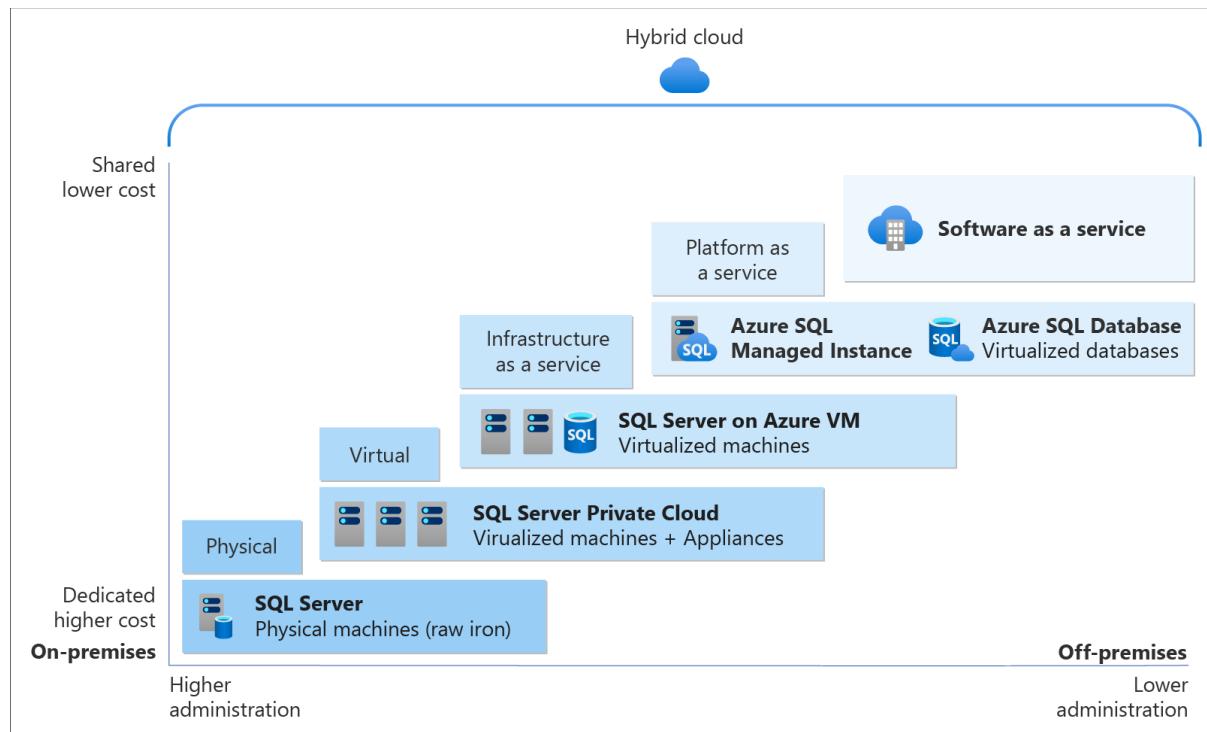
- Get industry-leading, multi-layered protection with [built-in security controls](#) including T-SQL, authentication, networking, and key management.
- Take advantage of the most comprehensive [compliance](#) coverage of any cloud database service.

Business motivations

There are several factors that can influence your decision to choose between the different data offerings:

- [Cost](#): Both platform as a service (PaaS) and infrastructure as a service (IaaS) options include base price that covers underlying infrastructure and licensing. However, with the IaaS option you need to invest additional time and resources to manage your database, while in PaaS you get administration features included in the price. IaaS enables you to shut down resources while you aren't using them to decrease the cost, while PaaS is always running unless you drop and re-create your resources when they're needed.
- [Administration](#): PaaS options reduce the amount of time that you need to invest to administer the database. However, it also limits the range of custom administration tasks and scripts that you can perform or run. For example, the CLR isn't supported with SQL Database, but is supported for an instance of SQL Managed Instance. Also, no deployment options in PaaS support the use of trace flags.
- [Service-level agreement](#): Both IaaS and PaaS provide high, industry standard SLA. PaaS option guarantees 99.99% SLA, while IaaS guarantees 99.95% SLA for infrastructure, meaning that you need to implement additional mechanisms to ensure availability of your databases. You can attain 99.99% SLA by creating an additional SQL virtual machine, and implementing the SQL Server Always On availability group high availability solution.
- [Time to move to Azure](#): SQL Server on Azure VM is the exact match of your environment, so migration from on-premises to the Azure VM is no different than moving the databases from one on-premises server to another. SQL Managed Instance also enables easy migration; however, there might be some changes that you need to apply before your migration.

Service comparison



As seen in the diagram, each service offering can be characterized by the level of administration you have over the infrastructure, and by the degree of cost efficiency.

In Azure, you can have your SQL Server workloads running as a hosted service ([PaaS](#)), or a hosted infrastructure ([IaaS](#)) supporting the software layer, such as Software-as-a-Service (SaaS) or an application. Within PaaS, you have multiple product options, and service tiers within each option. The key question that you need to ask when deciding between PaaS or IaaS is - do you want to manage your database, apply patches, and take backups - or do you want to delegate these operations to Azure?

Azure SQL Database

[Azure SQL Database](#) is a relational database-as-a-service (DBaaS) hosted in Azure that falls into the industry category of *Platform-as-a-Service (PaaS)*.

- Best for modern cloud applications that want to use the latest stable SQL Server features and have time constraints in development and marketing.
- A fully managed SQL Server database engine, based on the latest stable Enterprise Edition of SQL Server. SQL Database has two deployment options built on standardized hardware and software that is owned, hosted, and maintained by Microsoft.

With SQL Server, you can use built-in features and functionality that requires extensive configuration (either on-premises or in an Azure virtual machine). When

using SQL Database, you pay-as-you-go with options to scale up or out for greater power with no interruption. SQL Database has some additional features that aren't available in SQL Server, such as built-in high availability, intelligence, and management.

Azure SQL Database offers the following deployment options:

- As a *single database* with its own set of resources managed via a [logical SQL server](#). A single database is similar to a [contained database](#) in SQL Server. This option is optimized for modern application development of new cloud-born applications. [Hyperscale](#) and [serverless](#) options are available.
- An *elastic pool*, which is a collection of databases with a shared set of resources managed via a [logical server](#). Single databases can be moved into and out of an elastic pool. This option is optimized for modern application development of new cloud-born applications using the multi-tenant SaaS application pattern. Elastic pools provide a cost-effective solution for managing the performance of multiple databases that have variable usage patterns.

Note

[Elastic pools for Hyperscale](#) are currently in preview.

Azure SQL Managed Instance

[Azure SQL Managed Instance](#) falls into the industry category of *Platform-as-a-Service* (*PaaS*), and is best for most migrations to the cloud. SQL Managed Instance is a collection of system and user databases with a shared set of resources that is lift-and-shift ready.

- Best for new applications or existing on-premises applications that want to use the latest stable SQL Server features and that are migrated to the cloud with minimal changes. An instance of SQL Managed Instance is similar to an instance of the [Microsoft SQL Server database engine](#) offering shared resources for databases and additional instance-scoped features.
- SQL Managed Instance supports database migration from on-premises with minimal to no database change. This option provides all of the PaaS benefits of Azure SQL Database but adds capabilities that were previously only available in SQL Server VMs. This includes a native virtual network and near 100% compatibility with on-premises SQL Server. Instances of SQL Managed Instance provide full SQL Server access and feature compatibility for migrating SQL Servers to Azure.

SQL Server on Azure VM

[SQL Server on Azure VM](#) falls into the industry category *Infrastructure-as-a-Service (IaaS)* and allows you to run SQL Server inside a fully managed virtual machine (VM) in Azure.

- SQL Server installed and hosted in the cloud runs on Windows Server or Linux virtual machines running on Azure, also known as an infrastructure as a service (IaaS). SQL virtual machines are a good option for migrating on-premises SQL Server databases and applications without any database change. All recent versions and editions of SQL Server are available for installation in an IaaS virtual machine.
- Best for migrations and applications requiring OS-level access. SQL virtual machines in Azure are lift-and-shift ready for existing applications that require fast migration to the cloud with minimal changes or no changes. SQL virtual machines offer full administrative control over the SQL Server instance and underlying OS for migration to Azure.
- The most significant difference from SQL Database and SQL Managed Instance is that SQL Server on Azure Virtual Machines allows full control over the database engine. You can choose when to start maintenance activities including system updates, change the recovery model to simple or bulk-logged, pause or start the service when needed, and you can fully customize the SQL Server database engine. With this additional control comes the added responsibility to manage the virtual machine.
- Rapid development and test scenarios when you don't want to buy on-premises hardware for SQL Server. SQL virtual machines also run on standardized hardware that is owned, hosted, and maintained by Microsoft. When using SQL virtual machines, you can either pay-as-you-go for a SQL Server license already included in a SQL Server image or easily use an existing license. You can also stop or resume the VM as needed.
- Optimized for migrating existing applications to Azure or extending existing on-premises applications to the cloud in hybrid deployments. In addition, you can use SQL Server in a virtual machine to develop and test traditional SQL Server applications. With SQL virtual machines, you have the full administrative rights over a dedicated SQL Server instance and a cloud-based VM. It is a perfect choice when an organization already has IT resources available to maintain the virtual machines. These capabilities allow you to build a highly customized system to address your application's specific performance and availability requirements.

Comparison table

Additional differences are listed in the following table, but *both SQL Database and SQL Managed Instance are optimized to reduce overall management costs to a minimum for provisioning and managing many databases*. Ongoing administration costs are reduced since you don't have to manage any virtual machines, operating system, or database software. You don't have to manage upgrades, high availability, or [backups](#).

In general, SQL Database and SQL Managed Instance can dramatically increase the number of databases managed by a single IT or development resource. [Elastic pools](#) also support SaaS multi-tenant application architectures with features including tenant isolation and the ability to scale to reduce costs by sharing resources across databases. [SQL Managed Instance](#) provides support for instance-scoped features enabling easy migration of existing applications, as well as sharing resources among databases. Whereas [SQL Server on Azure VMs](#) provide DBAs with an experience most similar to the on-premises environment they're familiar with.

Azure SQL Database	Azure SQL Managed Instance	SQL Server on Azure VM
Supports most on-premises database-level capabilities. The most commonly used SQL database-level capabilities. Server features are available. 99.995% availability guaranteed. Built-in backups, patching, recovery. Latest stable Database Engine version. Ability to assign necessary resources (CPU/storage) to individual databases. Built-in advanced intelligence and security. Online change of resources (CPU/storage).	Supports almost all on-premises instance-level and High compatibility with SQL Server. 99.99% availability guaranteed. Built-in backups, patching, recovery. Latest stable Database Engine version. Easy migration from SQL Server.	You have full control over the SQL Server engine. Supports all on-premises capabilities. Up to 99.99% availability. Full parity with the matching version of on-premises SQL Server. Fixed, well-known Database Engine version. Easy migration from SQL Server. Private IP address within Azure Virtual Network.
Migration from SQL Server might be challenging. Some SQL Server features aren't available. Configurable maintenance windows . Compatibility with the SQL Server version can be achieved only using database compatibility levels. Private IP address support with Azure Private Link .	There's still some minimal number of SQL Server features that aren't available. Configurable maintenance windows . Compatibility with the SQL Server version can be achieved only using database compatibility levels.	You may use manual or automated backups . You need to implement your own High-Availability solution. There's a downtime while changing the resources(CPU/storage)
Databases of up to 100 TB.	Up to 16 TB.	SQL Server instances with up to 256 TB of storage. The instance can

Azure SQL Database	Azure SQL Managed Instance	SQL Server on Azure VM
On-premises application can access data in Azure SQL Database.	Native virtual network implementation and connectivity to your on-premises environment using Azure Express Route or VPN Gateway.	support as many databases as needed.

Cost

Whether you're a startup that is strapped for cash, or a team in an established company that operates under tight budget constraints, limited funding is often the primary driver when deciding how to host your databases. In this section, you learn about the billing and licensing basics in Azure associated with the Azure SQL family of services. You also learn about calculating the total application cost.

Billing and licensing basics

Currently, both **SQL Database** and **SQL Managed Instance** are sold as a service and are available with several options and in several service tiers with different prices for resources, all of which are billed hourly at a fixed rate based on the service tier and compute size you choose. For the latest information on the current supported service tiers, compute sizes, and storage amounts, see [DTU-based purchasing model for SQL Database](#) and [vCore-based purchasing model for both SQL Database and SQL Managed Instance](#).

- With SQL Database, you can choose a service tier that fits your needs from a wide range of prices starting from 5\$/month for basic tier and you can create [elastic pools](#) to share resources among databases to reduce costs and accommodate usage spikes.
- With SQL Managed Instance, you can also bring your own license. For more information on bring-your-own licensing, see [License Mobility through Software Assurance on Azure](#) or use the [Azure Hybrid Benefit calculator](#) to see how to **save up to 40%**.

In addition, you're billed for outgoing Internet traffic at regular [data transfer rates](#). You can dynamically adjust service tiers and compute sizes to match your application's varied throughput needs.

With **SQL Database** and **SQL Managed Instance**, the database software is automatically configured, patched, and upgraded by Azure, which reduces your administration costs. In addition, its [built-in backup](#) capabilities help you achieve significant cost savings, especially when you have a large number of databases.

With **SQL on Azure VMs**, you can use any of the platform-provided SQL Server images (which includes a license) or bring your SQL Server license. All the supported SQL Server versions (2008R2, 2012, 2014, 2016, 2017, 2019) and editions (Developer, Express, Web, Standard, Enterprise) are available. In addition, Bring-Your-Own-License versions (BYOL) of the images are available. When using the Azure provided images, the operational cost depends on the VM size and the edition of SQL Server you choose. Regardless of VM size or SQL Server edition, you pay per-minute licensing cost of SQL Server and the Windows or Linux Server, along with the Azure Storage cost for the VM disks. The per-minute billing option allows you to use SQL Server for as long as you need without buying additional SQL Server licenses. If you bring your own SQL Server license to Azure, you are charged for server and storage costs only. For more information on bring-your-own licensing, see [License Mobility through Software Assurance on Azure](#). In addition, you are billed for outgoing Internet traffic at regular [data transfer rates](#).

Calculating the total application cost

When you start using a cloud platform, the cost of running your application includes the cost for new development and ongoing administration costs, plus the public cloud platform service costs.

For more information on pricing, see the following resources:

- [SQL Database & SQL Managed Instance pricing](#)
- [Virtual machine pricing for SQL and for Windows](#)
- [Azure Pricing Calculator](#)

Administration

For many businesses, the decision to transition to a cloud service is as much about offloading complexity of administration as it's cost. With IaaS and PaaS, Azure administers the underlying infrastructure and automatically replicates all data to provide disaster recovery, configures and upgrades the database software, manages

load balancing, and does transparent failover if there's a server failure within a data center.

- With **SQL Database** and **SQL Managed Instance**, you can continue to administer your database, but you no longer need to manage the database engine, the operating system, or the hardware. Examples of items you can continue to administer include databases and logins, index and query tuning, and auditing and security. Additionally, configuring high availability to another data center requires minimal configuration and administration.
- With **SQL on Azure VM**, you have full control over the operating system and SQL Server instance configuration. With a VM, it's up to you to decide when to update/upgrade the operating system and database software and when to install any additional software such as anti-virus. Some automated features are provided to dramatically simplify patching, backup, and high availability. In addition, you can control the size of the VM, the number of disks, and their storage configurations. Azure allows you to change the size of a VM as needed. For information, see [Virtual Machine and Cloud Service Sizes for Azure](#).

Service-level agreement (SLA)

For many IT departments, meeting up-time obligations of a service-level agreement (SLA) is a top priority. In this section, we look at what SLA applies to each database hosting option.

For both **Azure SQL Database** and **Azure SQL Managed Instance**, Microsoft provides an availability SLA of 99.99%. For the latest information, see [Service-level agreement](#).

For **SQL on Azure VM**, Microsoft provides an availability SLA of 99.95% for two virtual machines in an availability set, or 99.99% for two virtual machines in different availability zones. This means that at least one of the two virtual machines will be available for the given SLA, but it does not cover the processes (such as SQL Server) running on the VM. For the latest information, see the [VM SLA](#). For database high availability (HA) within VMs, you should configure one of the supported high availability options in SQL Server, such as [Always On availability groups](#). Using a supported high availability option doesn't provide an additional SLA, but allows you to achieve >99.99% database availability.

Time to move to Azure

Azure SQL Database is the right solution for cloud-designed applications when developer productivity and fast time-to-market for new solutions are critical. With programmatic DBA-like functionality, it's perfect for cloud architects and developers as it lowers the need for managing the underlying operating system and database.

Azure SQL Managed Instance greatly simplifies the migration of existing applications to Azure, enabling you to bring migrated database applications to market in Azure quickly.

SQL on Azure VM is perfect if your existing or new applications require large databases or access to all features in SQL Server or Windows/Linux, and you want to avoid the time and expense of acquiring new on-premises hardware. It's also a good fit when you want to migrate existing on-premises applications and databases to Azure as-is - in cases where SQL Database or SQL Managed Instance isn't a good fit. Since you don't need to change the presentation, application, and data layers, you save time and budget on rearchitecting your existing solution. Instead, you can focus on migrating all your solutions to Azure and in doing some performance optimizations that may be required by the Azure platform. For more information, see [Performance Best Practices for SQL Server on Azure Virtual Machines](#).

Create and manage Azure SQL resources with the Azure portal

The Azure portal provides a single page where you can manage [all of your Azure SQL resources](#) including your SQL Server on Azure virtual machines (VMs).

To access the **Azure SQL** page, from the Azure portal menu, select **Azure SQL** or search for and select **Azure SQL** in any page.

Note

Azure SQL provides a quick and easy way to access all of your SQL resources in the Azure portal, including single and pooled databases in Azure SQL Database as well as the logical server hosting them, Azure SQL Managed Instances, and SQL Server on Azure VMs. [Azure SQL](#) is not a service or resource, but rather a family of SQL-related services.

To manage existing resources, select the desired item in the list. To create new Azure SQL resources, select + **Create**.

Home >

Azure SQL ...

Contoso, Ltd. (contoso.onmicrosoft.com)

+ Create Reservations Manage view Refresh Export to CSV Open query | Assign tags Delete | Feedback

Filter for any field... Subscription == Contoso Team Resource group == chrisqpublic-resources Location == all Add filter

Showing 1 to 8 of 8 records.

Name	Resource Type	Service tier	Resource group	Location	Subscription
AdventureWorksLT (chrisqpublictest/AdventureWorksLT)	SQL database	General Purpose: Ge...	chrisqpublic-resources	East US	Contoso Team
chrisqpublic-elasticpool (chrisqpublictest/chrisqpublic-el...)	SQL elastic ...	General Purpose: Ge...	chrisqpublic-resources	East US	Contoso Team
chrisqpublic-mi	SQL manag...	General Purpose: Ge...	chrisqpublic-resources	East US	Contoso Team
chrisqpublictest	SQL server	--	chrisqpublic-resources	East US	Contoso Team
chrisqpublictestwest3	SQL server	--	chrisqpublic-resources	West US 3	Contoso Team
mySampleDatabase (chrisqpublictest/mySampleDatabase)	SQL database	Hyperscale: Gen5, 2 ...	chrisqpublic-resources	East US	Contoso Team
ContosoDatabase (contososerver/ContosoDatabase)	SQL database	General Purpose: Ge...	chrisqpublic-resources	East US	Contoso Team
ContosoHyperscale (chrisqpublictest/ContosoHyperscale)	SQL database	Hyperscale: Gen4, 2 ...	chrisqpublic-resources	East US	Contoso Team

After selecting **+ Create**, view additional information about the different options by selecting **Show details** on any tile.

Home > Azure SQL >

Select SQL deployment option ...

Contoso, Ltd.

Feedback

How do you plan to use the service?

SQL databases
Best for modern cloud applications. Hyperscale and serverless options are available.
Resource type: Single database
Create Show details

SQL managed instances
Best for most migrations to the cloud. Lift-and-shift ready.
Resource type: Single instance
Create Show details

SQL virtual machines
Best for migrations and applications requiring OS-level access. Lift-and-shift ready.
Image
Create Show details

For details, see:

- [Create a single database](#)
- [Create an elastic pool](#)
- [Create a managed instance](#)
- [Create a SQL virtual machine](#)

Next steps

- See [Your first Azure SQL Database](#) to get started with SQL Database.
- See [Your first Azure SQL Managed Instance](#) to get started with SQL Managed Instance.

- See [SQL Database pricing](#).
- See [Azure SQL Managed Instance pricing](#).
- See [Provision a SQL Server virtual machine in Azure](#) to get started with SQL Server on Azure VMs.
- [Identify the right SQL Database or SQL Managed Instance SKU for your on-premises database](#).

What is Azure Database for MySQL?

- Article
- 03/28/2023
- 4 contributors

Feedback

In this article

1. [Deployment models](#)
2. [Contacts](#)
3. [Next steps](#)

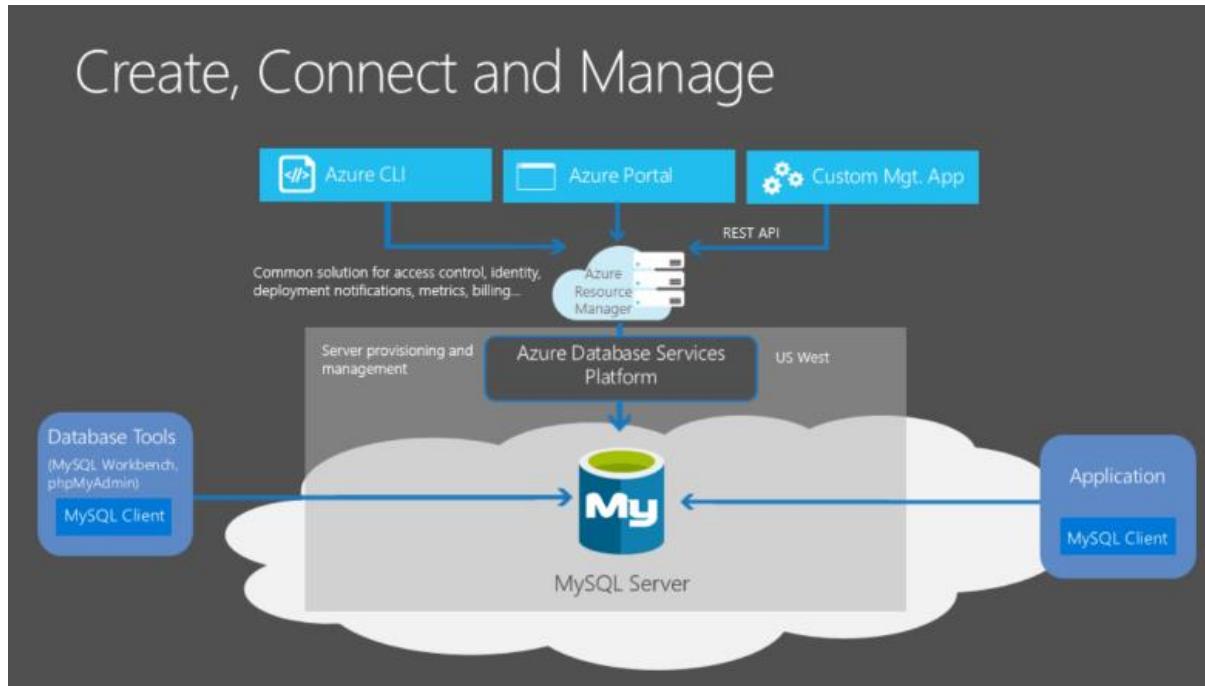
APPLIES TO:  Azure Database for MySQL - Single Server  Azure Database for MySQL - Flexible Server

Azure Database for MySQL is a relational database service in the Microsoft cloud based on the [MySQL Community Edition](#) (available under the GPLv2 license) database engine, versions 5.6 (retired), 5.7, and 8.0. Azure Database for MySQL delivers:

- Zone redundant and same zone high availability.
- Maximum control with ability to select your scheduled maintenance window.
- Data protection using automatic backups and point-in-time-restore for up to 35 days.
- Automated patching and maintenance for underlying hardware, operating system and database engine to keep the service secure and up to date.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Elastic scaling within seconds.
- Cost optimization controls with low cost burstable SKU and ability to stop/start server.
- Enterprise grade security, industry-leading compliance, and privacy to protect sensitive data at-rest and in-motion.
- Monitoring and automation to simplify management and monitoring for large-scale deployments.
- Industry-leading support experience.

These capabilities require almost no administration and all are provided at no extra cost. They allow you to focus on rapid app development and accelerating your time

to market rather than allocating precious time and resources to managing virtual machines and infrastructure. In addition, you can continue to develop your application with the open-source tools and platform of your choice to deliver with the speed and efficiency your business demands, all without having to learn new skills.



Deployment models

Azure Database for MySQL powered by the MySQL community edition is available in two deployment modes:

- Flexible Server
- Single Server

Azure Database for MySQL - Flexible Server

Azure Database for MySQL Flexible Server is a fully managed production-ready database service designed for more granular control and flexibility over database management functions and configuration settings. The flexible server architecture allows users to opt for high availability within single availability zone and across multiple availability zones. Flexible servers provides better cost optimization controls with the ability to stop/start server and burstable compute tier, ideal for workloads that don't need full compute capacity continuously. Flexible Server also supports reserved instances allowing you to save up to 63% cost, ideal for production workloads with predictable compute capacity requirements. The service supports

community version of MySQL 5.7 and 8.0. The service is generally available today in wide variety of [Azure regions](#).

The Flexible Server deployment option offers three compute tiers: Burstable, General Purpose, and Memory Optimized. Each tier offers different compute and memory capacity to support your database workloads. You can build your first app on a burstable tier for a few dollars a month, and then adjust the scale to meet the needs of your solution. Dynamic scalability enables your database to transparently respond to rapidly changing resource requirements. You only pay for the resources you need, and only when you need them. See [Compute and Storage](#) for details.

Flexible servers are best suited for

- Ease of deployments, simplified scaling and low database management overhead for functions like backups, high availability, security and monitoring
- Application developments requiring community version of MySQL with better control and customizations
- Production workloads with same-zone, zone redundant high availability and managed maintenance windows
- Simplified development experience
- Enterprise grade security

For detailed overview of flexible server deployment mode, refer [flexible server overview](#). For latest updates on Flexible Server, refer to [What's new in Azure Database for MySQL - Flexible Server](#).

Azure Database for MySQL - Single Server

Important

Azure Database for MySQL - Single Server is on the retirement path. We strongly recommend for you to upgrade to Azure Database for MySQL - Flexible Server. For more information about migrating to Azure Database for MySQL - Flexible Server, see [What's happening to Azure Database for MySQL Single Server?](#)

Azure Database for MySQL single server is a fully managed database service designed for minimal customization. The single server platform is designed to handle most of the database management functions such as patching, backups, high availability, security with minimal user configuration and control. The architecture is optimized for built-in high availability with 99.99% availability on single availability zone. It supports community version of MySQL 5.6 (retired), 5.7 and 8.0. The service is generally available today in wide variety of [Azure regions](#).

Single servers are best suited **only for existing applications already leveraging single server**. For all new developments or migrations, Flexible Server would be the recommended deployment option. To learn about the differences between Flexible Server and Single Server deployment options, refer to [select the right deployment option for your documentation](#).

For detailed overview of single server deployment mode, refer [single server overview](#).
For latest updates on Flexible Server, refer to [What's new in Azure Database for MySQL - Single Server](#).

What is Azure Database for PostgreSQL?

- Article
- 03/29/2023
- 7 contributors

Feedback

In this article

1. [Deployment models](#)
2. [Next steps](#)

APPLIES TO:  Azure Database for PostgreSQL - Single Server

Important

Azure Database for PostgreSQL - Single Server is on the retirement path. We strongly recommend for you to upgrade to Azure Database for PostgreSQL - Flexible Server. For more information about migrating to Azure Database for PostgreSQL - Flexible Server, see [What's happening to Azure Database for PostgreSQL Single Server?](#)

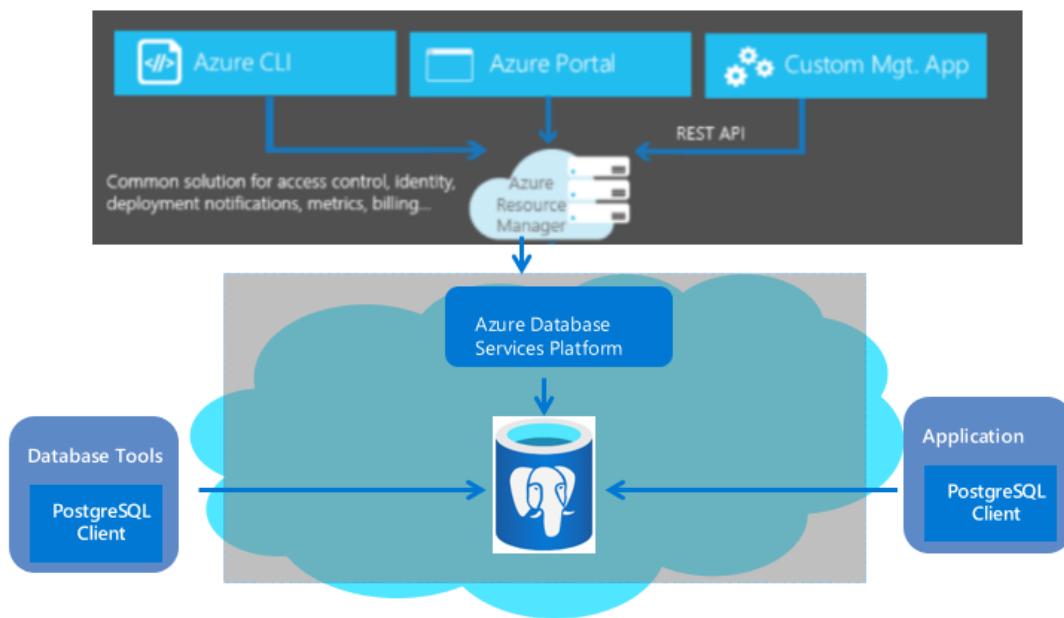
Important

Azure Database for PostgreSQL - Hyperscale (Citus) is now [Azure Cosmos DB for PostgreSQL](#). To learn more about this change, see [Where is Hyperscale \(Citus\)?](#)

Azure Database for PostgreSQL is a relational database service in the Microsoft cloud based on the [PostgreSQL open source relational database](#). Azure Database for PostgreSQL delivers:

- Built-in high availability.

- Data protection using automatic backups and point-in-time-restore for up to 35 days.
- Automated maintenance for underlying hardware, operating system and database engine to keep the service secure and up to date.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Elastic scaling within seconds.
- Enterprise grade security and industry-leading compliance to protect sensitive data at-rest and in-motion.
- Monitoring and automation to simplify management and monitoring for large-scale deployments.
- Industry-leading support experience.



These capabilities require almost no administration, and all are provided at no additional cost. They allow you to focus on rapid application development and accelerating your time to market rather than allocating precious time and resources to managing virtual machines and infrastructure. In addition, you can continue to develop your application with the open-source tools and platform of your choice to deliver with the speed and efficiency your business demands, all without having to learn new skills.

Deployment models

Azure Database for PostgreSQL powered by the PostgreSQL community edition is available in two deployment modes:

- Single Server
- Flexible Server

Azure Database for PostgreSQL - Single Server

Azure Database for PostgreSQL Single Server is a fully managed database service with minimal requirements for customizations of database. The single server platform is designed to handle most of the database management functions such as patching, backups, high availability, security with minimal user configuration and control. The architecture is optimized for built-in high availability with 99.99% availability on single availability zone. It supports community version of PostgreSQL 9.5, 9.6, 10, and 11. The service is generally available today in wide variety of [Azure regions](#).

The Single Server deployment option offers three pricing tiers: Basic, General Purpose, and Memory Optimized. Each tier offers different resource capabilities to support your database workloads. You can build your first app on a small database for a few dollars a month, and then adjust the scale to meet the needs of your solution. Dynamic scalability enables your database to transparently respond to rapidly changing resource requirements. You only pay for the resources you need, and only when you need them. See [Pricing tiers](#) for details.

Single servers are best suited for cloud native applications designed to handle automated patching without the need for granular control on the patching schedule and custom PostgreSQL configuration settings.

For detailed overview of single server deployment mode, refer [single server overview](#).

Azure Database for PostgreSQL - Flexible Server

Azure Database for PostgreSQL Flexible Server is a fully managed database service designed to provide more granular control and flexibility over database management functions and configuration settings. In general, the service provides more flexibility and customizations based on the user requirements. The flexible server architecture allows users to opt for high availability within single availability zone and across multiple availability zones. Flexible Server provides better cost optimization controls with the ability to stop/start server and burstable compute tier, ideal for workloads that don't need full-compute capacity continuously. The service currently supports community version of PostgreSQL 11, 12, 13 and 14, with plans to add newer versions soon. The service is generally available today in wide variety of Azure regions.

Flexible servers are best suited for

- Application developments requiring better control and customizations
- Cost optimization controls with ability to stop/start server
- Zone redundant high availability
- Managed maintenance windows

Choose the right PostgreSQL server option in Azure

- Article
- 03/29/2023
- 3 contributors

Feedback

In this article

1. [Total cost of ownership \(TCO\)](#)
2. [Billing](#)
3. [Administration](#)
4. [Time to move to Azure PostgreSQL Service \(PaaS\)](#)
5. [Next steps](#)

APPLIES TO:  Azure Database for PostgreSQL - Flexible Server

Important

Azure Database for PostgreSQL - Single Server is on the retirement path. We strongly recommend for you to upgrade to Azure Database for PostgreSQL - Flexible Server. For more information about migrating to Azure Database for PostgreSQL - Flexible Server, see [What's happening to Azure Database for PostgreSQL Single Server?](#)

With Azure, your PostgreSQL Server workloads can run in a hosted virtual machine infrastructure as a service (IaaS) or as a hosted platform as a service (PaaS). PaaS has multiple deployment options, each with multiple service tiers. When you choose between IaaS and PaaS, you must decide if you want to manage your database, apply patches, and make backups, or if you want to delegate these operations to Azure.

When making your decision, consider the following option in PaaS or alternatively running on Azure VMs (IaaS)

- [Azure Database for PostgreSQL Flexible Server](#)

PostgreSQL on Azure VMs option falls into the industry category of IaaS. With this service, you can run PostgreSQL Server inside a fully managed virtual machine on the Azure cloud platform. All recent versions and editions of PostgreSQL can be installed on an IaaS virtual machine. In the most significant difference from Azure Database for PostgreSQL, PostgreSQL on Azure VMs offers control over the database engine. However, this control comes at the cost of responsibility to manage the VMs and many database administration (DBA) tasks. These tasks include maintaining and patching database servers, database recovery, and high-availability design.

The main differences between these options are listed in the following table:

Attribute	Postgres on Azure VMs	PostgreSQL as PaaS
Availability SLA	- Virtual Machine SLA	- Flexible Server
OS and PostgreSQL patching	- Customer managed	- Flexible Server – Automatic with optional customer managed window
High availability	- Customers architect, implement, test, and maintain high availability. Capabilities might include clustering, replication etc.	- Flexible Server: built-in
Zone Redundancy	- Azure VMs can be set up to run in different availability zones. For an on-premises solution, customers must create, manage, and maintain their own secondary data center.	- Flexible Server: Yes
Hybrid Scenario	- Customer managed	- Flexible Server: supported
Backup and Restore	- Customer Managed	- Flexible Server: built-in with user configuration on zone-redundant storage
Monitoring Database Operations	- Customer Managed	- Flexible Server: All offer customers the ability to set alerts on the database operation and act upon reaching thresholds
Advanced Threat Protection	- Customers must build this protection for themselves.	- Flexible Server: Not available during Preview
Disaster Recovery	- Customer Managed	- Flexible Server: supported
Intelligent Performance	- Customer Managed	- Flexible Server: supported

Total cost of ownership (TCO)

TCO is often the primary consideration that determines the best solution for hosting your databases. This is true whether you're a startup with little cash or a team in an established company that operates under tight budget constraints. This section describes billing and licensing basics in Azure as they apply to Azure Database for PostgreSQL and PostgreSQL on Azure VMs.

Billing

Azure Database for PostgreSQL is currently available as a service in several tiers with different prices for resources. All resources are billed hourly at a fixed rate. For the latest information on the currently supported service tiers, compute sizes, and storage amounts, see [pricing page](#). You can dynamically adjust service tiers and compute sizes to match your application's varied throughput needs. You're billed for outgoing Internet traffic at regular [data transfer rates](#).

With Azure Database for PostgreSQL, Microsoft automatically configures, patches, and upgrades the database software. These automated actions reduce your administration costs. Also, Azure Database for PostgreSQL has [automated backup-link](#) capabilities. These capabilities help you achieve significant cost savings, especially when you have a large number of databases. In contrast, with PostgreSQL on Azure VMs you can choose and run any PostgreSQL version. However, you need to pay for the provisioned VM, storage cost associated with the data, backup, monitoring data and log storage and the costs for the specific PostgreSQL license type used (if any).

Azure Database for PostgreSQL Flexible Server provides built-in high availability at the zonal-level (within an AZ) for any kind of node-level interruption while still maintaining the [SLA guarantee](#) for the service. Flexible Server provides [uptime SLAs](#) with and without zone-redundant configuration. However, for database high availability within VMs, you use the high availability options like [Streaming Replication](#) that are available on a PostgreSQL database. Using a supported high availability option doesn't provide an additional SLA. But it does let you achieve greater than 99.99% database availability at additional cost and administrative overhead.

For more information on pricing, see the following articles:

- [Azure Database for PostgreSQL pricing](#)
- [Virtual machine pricing](#)
- [Azure pricing calculator](#)

Administration

For many businesses, the decision to transition to a cloud service is as much about offloading complexity of administration as it is about cost.

With IaaS, Microsoft:

- Administers the underlying infrastructure.
- Provides automated patching for underlying hardware and OS

With PaaS, Microsoft:

- Administers the underlying infrastructure.
- Provides automated patching for underlying hardware, OS and database engine.
- Manages high availability of the database.
- Automatically performs backups and replicates all data to provide disaster recovery.
- Encrypts the data at rest and in motion by default.
- Monitors your server and provides features for query performance insights and performance recommendations.

With Azure Database for PostgreSQL, you can continue to administer your database. But you no longer need to manage the database engine, the operating system, or the hardware. Examples of items you can continue to administer include:

- Databases
- Sign-in
- Index tuning
- Query tuning
- Auditing
- Security

Additionally, configuring high availability to another data center requires minimal to no configuration or administration.

- With PostgreSQL on Azure VMs, you have full control over the operating system and the PostgreSQL server instance configuration. With a VM, you decide when to update or upgrade the operating system and database software and what patches to apply. You also decide when to install any additional software such as an antivirus application. Some automated features are provided to greatly simplify patching, backup, and high availability. You can control the size of the VM, the number of disks, and their storage configurations. For more information, see [Virtual machine and cloud service sizes for Azure](#).

Time to move to Azure PostgreSQL Service (PaaS)

- Azure Database for PostgreSQL is the right solution for cloud-designed applications when developer productivity and fast time to market for new solutions are critical. With programmatic functionality that is like DBA, the service is suitable for cloud architects and developers because it lowers the need for managing the underlying operating system and database.

- When you want to avoid the time and expense of acquiring new on-premises hardware, PostgreSQL on Azure VMs is the right solution for applications that require a granular control and customization of PostgreSQL engine not supported by the service or requiring access of the underlying OS.

Overview of Azure compute services

Save note

TranscriptNotesDownloadsDiscuss

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Let's start this session by

looking at an overview of Azure compute.

Azure compute is an on-demand computing service
for running Cloud-based applications.

It provides computing resources such as disks,
processors, memory, networking, and operating systems.

The resources are available on demand and can
typically be made available in minutes or even seconds.

You pay only for the resources you use,
and only for as long as you're using them.

Azure supports a wide range of
computing solutions for development and testing,
running applications and extending your data center.

The service supports Linux, Windows Server,
SQL Server, Oracle, IBM and SAP.

Azure also has many services
that can run Virtual Machines.

Each service provides different options
depending on your requirements.

In this lesson, you will examine some of the main services including Azure Virtual Machines, including Virtual Machines Scale Set, Azure Container instances including Azure Kubernetes, Azure App Service, and Azure functions, or serverless computing.

What are Virtual Machines?

Virtual Machines are software emulations of physical computers.

They include a virtual processor, memory, storage, and networking resources.

Virtual Machines host an operating system and you can install and run software just like a physical computer.

When using a remote desktop client, you can use and control the Virtual Machine as if you were sitting in front of it.

With Azure Virtual Machines, you can create and use Virtual Machines in the Cloud.

Virtual Machines, also known as VMs, provide infrastructure as a service, IaaS, and can be used in different ways.

When you need total control over an operating system and environment,

Virtual Machines are an ideal choice.

Just like a physical computer, you can customize all the software running on the Virtual Machine.

This ability is helpful when you're running custom software or custom hosting configurations.

For example, if Tailwind Traders wants to provision Linux and Windows Virtual Machines with the configurations of their choice, they could do so in seconds

using Azure Virtual Machine services.

You know that virtual machines are software emulations of physical computers, but what are Virtual Machine Scale Sets?

Virtual Machine Scale Sets are an Azure compute resource that you can use to deploy and manage a set of identical Virtual Machines, Azure Virtual Machine Scale Sets lets you create and manage a group of identical load balanced Virtual Machines.

Scale Sets allow you to centrally manage, configure and update a large number of Virtual Machines in minutes to provide highly available applications.

The number of Virtual Machines instances can automatically increase or decrease in response to demand or defined schedule.

For this reason, it's easier to build large-scale services targeting Big Compute, Big Data, and containerized workloads.

As demand goes up, more Virtual Machine instances can be added.

As demand goes down, Virtual Machine instances can be removed.

The process can be manual, automated, or a combination of both.

For example, if Tailwind Traders want to achieve high availability by auto-scaling to create thousands of Virtual Machines, they could do so in minutes using Virtual Machine Scale Sets.

Let's take a look at some more Azure compute resources. Container instances and Azure Kubernetes Service or AKS, are Azure compute resources that you can

use to deploy and manage containers.

Containers are lightweight
virtualized application environments.

They're designed to be quickly created,
scaled out, and stopped dynamically.

You can run multiple instances of
a containerized application on a single host machine.

For example, if Tailwind Traders wants to
containerize apps and easily
run containers with a single command,
they would use Container Instances.

Azure Kubernetes Service is also
ideal to simplify the deployment,
management and operations of Kubernetes.

You've learned that Azure Virtual Machines provide
infrastructure as a service or IaaS.

Azure App Service, on the other hand,
is a platform as a service or PaaS offering.

With Azure App Service,
you can quickly build, deploy,
and scale enterprise grade web,
mobile and API apps running on any platform.

You can meet rigorous performance, scalability,
security and compliance requirements while using
a fully managed platform
to perform infrastructure maintenance.

For example, if Tailwind Traders wants to quickly create
Cloud apps for web and
mobile with fully managed platform,
they can use Azure App Service.

Let's suppose you're not concerned about
the underlying platform or infrastructure,
but only about the code running your service.

Functions are ideal for this scenario.

They're commonly used when you need to

perform work in response to an event,
often via a rest request,
timer or message from another Azure service and
when that work can be completed
quickly within seconds or less.

For example, if Tailwind Traders wants to accelerate
app development using
an event-driven serverless architecture,
they can use Azure Functions.

In this session, you have started
gathering information that will
help you to resolve Tailwind Traders challenges.

You looked at an overview of
Azure Virtual Machines, Azure App Service,
Azure Container Instances, Azure Kubernetes,
Azure Functions and Virtual Machine Scale Set.

When to use Azure Virtual Machines

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

In this session, you continue to explore the possible Azure solutions that can help you scale out your applications.

One possible solution to Tailwind Traders lack of physical service is using virtual machines.

Let's dive right in and explore virtual machines as a solution.

With Azure Virtual Machines, you can create and use virtual machines in the cloud.

Virtual machines provide infrastructure as a service in

the form of a virtualized server, and could be used in many ways.

Just like a physical computer,

you can customize all of the software running on the virtual machines.

Virtual machines are an ideal choice when you need total control over the operating system, the ability to run custom software, and to use custom hosting configurations.

An Azure Virtual Machine gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs the virtual machine.

You still need to configure, update, and maintain the software that runs on the virtual machine.

You can create a provision of virtual machine in minutes when you select a pre-configured virtual machine image.

Selecting an image is one of the most important decisions you'll make when you create a virtual machine.

An image is a template used to create a virtual machine.

These templates already include an operating system and often other software like development tools or web hosting environments.

Here are some examples of when to use virtual machines.

During testing and development.

Virtual machines provide a quick and easy way to create different operating system and application configurations.

Test and development personnel could then easily delete the virtual machines when they no longer need them.

When running applications in the cloud.

The ability to run certain applications in the public cloud, as opposed to creating a traditional infrastructure to run them, can provide substantial economic benefits.

For example, an application might need to handle fluctuations in demand.

Shutting down VMS when you don't need them or quickly starting them up to meet a sudden increase in demand means you pay only for the resources you use.

When extending your datacenter to the cloud.

An organization can extend the capabilities of its own on-premises network by creating a virtual network in Azure and adding VMS to that virtual network.

Applications like SharePoint can then run on an Azure VM instead of running locally.

This arrangement makes it easier or

less expensive to deploy than in an on- premises environment.

During disaster recovery.

As with running certain types of applications in the cloud and extending an on- premises network to the cloud.

You can get significant cost savings by using an IAS based approach to disaster recovery.

If a primary datacenter fails,

you can create VMS running on Azure to run your critical applications and then shut them down when the primary data center becomes operational again.

Virtual machines are also an excellent choice when you move from a physical server to the cloud.

This is also known as lift and shift.

You can create an image of the physical server and host it within a virtual machine with little or no changes.

Just like a physical on-premises server, you must maintain the virtual machine.

You update the installed operating system on the software it runs.

You can run single virtual machines for testing, development, or minor tasks.

Or you can group virtual machines together to provide high availability, scalability, and redundancy.

No matter what you're up time requirements are,

Azure has several features that can meet them.

These features include virtual machine scale sets, Azure Batch.

Let's start off by looking at virtual machine scale sets.

Virtual machine scale sets lets you create and manage a group of identical load balanced virtual machines.

Imagine you're running a website that enables scientists to upload astronomy images that need to be processed.

If you duplicated the virtual machine, you'd normally need to configure an additional service to write requests between multiple instances of the website.

Virtual machine scale sets could do that work for you.

Scale sets allow you to centrally manage, configure, and update a large number of virtual machines in minutes to provide highly available applications.

The number of virtual machine instances can automatically increase or decrease in response to demand or define schedule.

With virtual machine scale sets, you can build large scale services for

areas such as compute, big data, and other container workloads.

Now that you have explored virtual scale sets, let's take a look at Azure Batch.

Azure Batch enables large scale parallel and high performance computing or HPC, batch jobs with the ability to scale to tens, hundreds or thousands of virtual machines.

When you're ready to run a job, Batch does the following.

Starts a pool of compute virtual machines for you.

Installs applications and staging data,
runs jobs with as many tasks as you have, identifies failures,
requests work, scales down the pool as work completes.

There might be situations in which you need more computing power or supercomputer level compute power.

Azure provides these capabilities.

In this session, we learned that Tailwind Traders frontend web servers are operating near capacity during peak periods of the day.

As a solution to match customer demand, scale sets can automatically increase the number of virtual machine instances as application demand increases.

Then reduce the number of VM instances as demand decreases.

This ability helps reduce costs and
efficiently create Azure resources as required.

Azure Container services

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

In the previous session,
you identified that Virtual Machines are
an ideal choice when you need
total control over an operating system and environment.

While Virtual Machines are an excellent way to reduce costs versus the investments that are necessary for physical hardware, they're still limited to a single operating system per virtual machine.

If you want to run multiple instances of an application on a single host machine, Containers are an excellent choice.

Containers are a virtualization environment, much like running multiple virtual machines on a single physical host,

you can run multiple containers on a single physical or virtual host.

Unlike Virtual Machines, you don't manage the operating system for a Container.

Virtual Machines appear to be an instance of an operating system that you can connect to and manage but

Containers are lightweight and designed to be created, scaled out and stopped dynamically.

While it's possible to create and deploy virtual machines as application demand increases,

Containers are designed to allow you to respond to changes on-demand.

With containers, you can quickly restart in case of a crash or hardware interruption.

One of the most popular Container Engines is Docker, which is supported by Azure.

Azure Kubernetes Services

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

As you investigate the possible Azure solutions

that are available to meet your needs,

it's important to compare

the different features and benefits.

If you were to select Container Instances as an option,

how would you manage this?

Containers are managed through a container orchestrator,

which can start, stop,

and scale-out application instances as needed.

There are two ways to manage

both Docker and Microsoft-based containers in Azure.

Azure Container Instances or

ACI and Azure Kubernetes Service or AKS.

Azure Container Instances offers

the fastest and simplest way to run a container in Azure,

without having to manage

any Virtual Machines or adopt any additional services.

It's a platform as

a service offering that

allows you to upload your containers,

which it runs for you.

Azure Kubernetes Service is

a complete orchestration service for

containers with distributed architectures

and large volumes of containers.

Orchestration is the task of automating and

managing a large number of

containers and how they interact.

Microservice architecture

[Save note](#)
[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Containers are often used to create solutions by using a micro service architectures.

A microservice architecture consists of a collection of small autonomous services.

Each service is self contained and should implement a single business capability.

This image illustrates the microservices architecture,

this architecture is where you break solutions into smaller independent pieces.

For example, you might split a website into a container hosting your front end, another hosting your back end, and a third for storage.

This split allows you to separate portions of your app into logical sections that could be maintained, scaled or updated independently.

Imagine your website back end has reached capacity, but the front end and storage aren't being stressed.

You could scale the back end separately to improve performance, decide to use a different storage service, and replace the storage container without affecting the rest of the application.

When to use Azure App Service

[Save note](#)
[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

In your research for Tailwind Traders,

you've looked at two different ways
that you can virtualize your application.
Another alternative is to deploy
your application's front end websites
to Azure App Service,
which makes it easy to respond to application demand.
App Service enables you to build and
host web apps, background jobs,
mobile backends, and RESTful APIs in
the programming language of
your choice, without managing infrastructure.
It offers automatic scaling and high availability.
App Service supports Windows and Linux,
and enables automated deployments from GitHub,
Azure DevOps, or any git
repo to support a continuous deployment model.
This platform as a service environment allows you
to focus on the website and API logic,
while Azure handles the infrastructure to
run and scale your web applications.
You pay for the Azure compute
resources your app uses while it
processes requests based on
the App Service plan you choose.
The App Service plan determines
how much hardware is devoted to your host.
For example, the plan
determines whether it's dedicated or
shared hardware and how much memory is reserved for it.
There's even a free tier you can use to
host small, low traffic sites.
With App Service, you could host
most common App Service styles like web apps,
API apps, WebJobs, and mobile apps.
App Service handles most of

the infrastructure decisions you deal with
in hosting web accessible apps.

Deployment and management are
integrated into the platform.

Endpoints can be secured.

Sites can be scaled quickly to handle high traffic loads.

The built-in load balancing and
traffic manager provide high availability.

All of these app styles are hosted in
the same Infrastructure and share these benefits.

This flexibility makes App Service the
ideal choice to host web-oriented applications.

App Service includes full support for
hosting web apps by using ASP.NET,
ASP.NET Core, Java, Ruby,
Node.js, PHP, or Python.

You can choose either Windows or
Linux as the host operating system.

Much like hosting a website,
you can build REST-based web APIs
by using your choice of language and framework.

You get full swagger support and the ability to
package and publish your API in Azure Marketplace.

The produced apps can be consumed from
any HTTP or HTTPS based client.

You can use the WebJobs feature to
run a program,.EXE, Java,
PHP, Python or Node.js, or script.cmd.

PowerShell or Bash in the same context as a web app,
API app, or a mobile app.

They can be scheduled or run by a trigger.

WebJobs are often used to run
background tasks as part of your application logic.

You can also use the mobile apps feature of App Service
to quickly build a backend for iOS and Android apps.

With just a few clicks in the Azure portal,
you can store mobile app data
in a cloud-based SQL database.
Authenticate customers against common social providers,
such as MSA, Google, Twitter, and Facebook.
Send push notifications, execute
custom backend logic in C# or Node.js.
On the mobile app side,
this SDK support for native iOS and Android,
Xamarin, and React native apps.
To summarize, Azure App Service is
a fully managed web hosting
service for building web apps,
mobile backends, and RESTful APIs.
Azure App Service also provides pricing and
performance options that cater for every need,
from small websites to globally scaled web applications.

When to use Azure Functions

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

After consulting with several of
your fellow developers at Tailwind Traders,
you've determined that some of
your application logic is event-driven.
In other words, for a large amount of time,
your application is waiting for
a particular input before it performs any processing.

To reduce your costs,
you want to avoid having to pay for
the time that your application is waiting for input.
With that in mind, you've decided to
investigate Azure functions to see if it can help.
Azure functions is the serverless computing service
hosted on the Microsoft Azure public cloud.
Azure functions and serverless computing in general is
designed to accelerate and
simplify application development.
Before launching straight in,
let's first look at serverless computing.
Serverless computing is the abstraction of servers,
infrastructure, and operating systems.
With serverless computing, Azure takes care of managing
the server infrastructure and the allocation and
deallocation of resources based on demand.
Infrastructure isn't your responsibility.
Scaling and performance are handled automatically.
You build only for the exact resources you use.
There's no need to even reserve capacity.
Serverless computing includes the abstraction of
servers and event-driven scale and micro-billing.
Let's explore each of these concepts.
Serverless computing abstracts the servers you run on.
You never explicitly reserved server instances.
The platform manages that for you.
Each function execution can
run on a different Compute Instance.
This execution context is transparent to the code.
With serverless architecture, you deploy
your code which then runs with high availability.
Serverless computing is an excellent fit for
workloads that respond to incoming events.
Events include triggers by timers,

for example, if a function needs
to run every day at 10:00 AM UTC,
HTTP for example, API and web hooks scenarios,
queues, for example, with order processing and much more.
Instead of writing an entire application,
the developer offers a function,
which contains both code and
metadata about its triggers and bindings.
The platform automatically schedules
the function to run and
scales the number of
compute instances based on the rate of incoming events.
Triggers define how a function is invoked.
Bindings provide a declarative way
to connect to services from within the code.
Traditional computing bills for a block of time,
like paying a monthly or annual rate for website hosting.
This method of billing is convenient,
but isn't always cost-effective.
Even if a customer's website gets only one hit today,
they still pay for a full day's worth of availability.
With serverless computing, they pay
only for the time their code runs.
If no active function executions
occur, they're not charged.
For example, if the code runs once a day for two minutes,
they're charged for one execution
and two minutes of computing time.

Azure Functions and Logic Apps

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your

selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Azure has two implementations of serverless computer.

Functions can execute code in almost any modern language.

Logic Apps are designed in

a web-based designer and can execute

logic triggered by Azure services

without writing any code.

Let's first take a look at Azure Functions.

When you're concerned only about

the code running your service and

not the underlying platform or

infrastructure using Azure Functions is ideal.

Functions are commonly used when you need to

perform work in response to an event,

often via a REST request,

timer or message from another Azure service,

and when that work can be completed

quickly within seconds or less.

Using a virtual machine-based approach you'd

incur costs even when the virtual machine is idle.

With Functions,

Azure runs your code when it's triggered and

automatically deallocates resources when

the function is finished.

In this model, you're only charged for

the CPU time used while your function runs.

Functions can be either stateless or stateful.

When they're stateless, the default,

they behave as if they're

restarted every time they respond to an event.

When they're stateful called Durable Functions

a context is passed through

the function to track prior activity.

Functions are a key component of serverless computing.

They're also a general compute platform
for running any type of code.

If the needs of the developer's app change you can
deploy the project in
an environment that isn't serverless.

This flexibility allows you to manage scaling,
run on virtual networks,
and even completely isolate the Functions.

In our case study, Tailwind Traders,
you determined that some of
your application logic is event-driven.

To reduce your costs,
you want to avoid having to pay for
the time that your application is waiting for input.

Azure Functions is ideal in this scenario.

The serverless app runs
only when it's triggered by an event.

The provider charges only for
compute time used by that execution
rather than a flat monthly fee
for maintaining a physical or virtual server.

Now, you'll explore Azure Logic Apps.

Logic Apps are similar to Functions.

Both enable you to trigger logic based on an event.

Where Functions execute code,
Logic Apps execute workflows that are designed to
automate business scenarios and are
built from predefined logic blocks.

Every Azure Logic Apps workflow
starts with a trigger which fires when
a specific event happens or when
newly available data meets specific criteria.

Many triggers include basic scheduling capabilities.

Developers can specify how

regularly their workloads will run.

Each time the trigger fires,
the Logic Apps engine creates
a Logic App instance that
runs the actions in the workflow.

These actions can also include data conversions and
flow controls such as conditional statements,
switch statements, loops, and branching.

You create Logic App workflows by using
a visual Designer on
the Azure portal or in Visual Studio.

The workflows are persisted as
a JSON file with a known workflow schema.

Azure provides more than 200 different connectors and
processing blocks to interact with different services.

These resources include the most popular enterprise apps.
You can also build custom connectors and workflow steps,
if the service you need to interact with isn't covered.

You then use the visual Designer
to link connectors and blocks together.

You pass data through
the workflow to do custom processing,
often all without writing any code.

As an example, let's say a ticket arrives in Zendesk.

You could detect the intent
of the message with cognitive services,
create an item in SharePoint to track the issue,
add the customer to your Dynamics 365 CRM system,
if they aren't already in your database,
send a follow-up e-mail to acknowledge their request.

All of those actions could be
designed in a visual Designer,
which makes it easy to see the logic flow.
For this reason, it's ideal for a business analyst role.

Functions versus Logic Apps

Functions versus Logic Apps

Functions and Logic Apps can both create complex orchestrations. An orchestration is a collection of functions or steps that are executed to accomplish a complex task.

- With Functions, you write code to complete each step.
- With Logic Apps, you use a GUI to define the actions and how they relate to one another.

You can mix and match services when you build an orchestration, calling functions from logic apps and calling logic apps from functions.

Here are some common differences between the two.

	Functions	Logic Apps
State	Normally stateless, but Durable Functions provide state.	Stateful.
Development	Code-first (imperative).	Designer-first (declarative).
Connectivity	About a dozen built-in binding types. Write code for custom buildings.	Large collection of connectors. Enterprise Integration Pack for B2B scenarios. Build custom connectors.
Actions	Each activity is an Azure function. Write code for activity functions.	Large collection of ready-made actions.
Monitoring	Azure Application insights.	Azure portal, Log Analytics.
Management	REST API, Visual Studio.	Azure portal, REST API, PowerShell, Visual Studio.
Execution Context	Can run locally or in the cloud.	Runs only in the cloud.

When to use Windows Virtual Desktop

Save note

TranscriptNotesDownloadsDiscuss

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

In addition to the challenges that

Tailwind Traders has been facing with application scale, your manager has asked you to put together a new development team of remote workers.

This task would normally require setting up several new computers, with all of the requisite development tools for your new team.

Then you would need to ship them to the respective developers across the country.

The time to procure setup and ship each of these computers would be costly.

Also, all of your new developers have their own computing devices, that are running a mixture of Windows, Android, and MacOS Operating Systems.

You want to find a way to expedite the deployment process, for your remote workers.

You also want to keep your management costs to a minimum.

With that in mind, you want to see how Windows Virtual Desktop can help your organization.

Windows Virtual Desktop on Azure, is a desktop and application virtualization service, that runs on the Cloud.

It enables your users to use a client hosted version of Windows from any location.

Windows Virtual Desktop works across devices like Windows, Mac, iOS, Android, and Linux.

It works with apps that you can use to access remote desktops and apps.

You can also use most modern browsers to access Windows Virtual Desktop hosted experiences.

Why should you use Windows Virtual Desktop?

Windows Virtual Desktop provides the best user experience.

Users have the freedom to connect to Windows Virtual Desktop with

any device over the Internet.

They use a Windows Virtual Desktop client,
to connect to their published
Windows desktops and applications.

This kind could either be
a native application on the device,
or the Windows Virtual Desktop HTML5 web client.

You can make sure
your session host virtual machines
run near apps and services,
that connect to your data center or the Cloud.

This way, your users stay productive
and don't encounter long load times.

Users sign in to Windows.

Virtual Desktop is fast,
because user profiles are containerized by using FSLogix.

At signing the user profile container is
dynamically attached to the computing environment.

The user profile is immediately available and
appears in the system exactly like a native user profile.

You can provide individual ownership
through personal persistent desktops.

For example, you might want to provide
personal remote desktops for
members of an engineering team,
then they can add or remove programs
without impacting other users on that remote desktop.

Windows Virtual Desktop also provides enhanced security.

Windows Virtual Desktop provides
centralized security management,
for users desktops with Azure Active Directory, Azure AD.

You can enable multi-factor authentication
to secure user sign-ins.
You can also secure access to data by assigning
granular role-based access controls or a box to users.

With Windows Virtual Desktop,
the data and apps are separated from the local hardware,
Windows Virtual Desktop runs
them instead on a remote server,
the risk of confidential data being
left on a personal device is reduced.

User sessions are isolated in
both single and multi-session environments.

Windows Virtual Desktop also improves
security by using reverse connect technology.

This connection type is more secure,
than the Remote Desktop Protocol,
we don't open in-band ports to
the session host virtual machines.

Some of the key features of
Windows Virtual Desktop include simplified management,
performance management,
and multi-session Windows 10 Deployment.

Let's dive right in to find that more.

Let's start off with simplified management.

Windows Virtual Desktop is an Azure service,
so it will be familiar to Azure administrators.

You use Azure AD and
our backs to manage access to resources.

With Azure, you also get tools
to automate VM deployments,
manage VM updates, and provide disaster recovery.

As with other Azure services,
Windows Virtual Desktop uses
Azure Monitor for monitoring and alerts.

This standardization, Let's Admins
identify issues through a single interface.

Now, let's take a look at
the performance management feature
of Windows Virtual Desktop.

Windows Virtual Desktop gives you options to load balance users on your virtual machine host pools. Host pools are collections of Virtual Machines, with the same configuration assigned to multiple users. For the best performance, you can configure low balancing to occur as user sign-in,. Breath mode. With breath mode, users are sequentially allocated across the host pool for your workload. To save costs, you can configure your virtual machines for depth mode, load balancing for users are fully allocated, on one virtual machine before moving on to the next. Windows Virtual Desktop provides tools to automatically provision additional virtual machines, when incoming demand exceeds a specified threshold. Another key feature, is the ability to set up a multi-session Windows 10 deployment, that delivers a full Windows 10 with scalability. Windows Virtual Desktop lets you use Windows 10 Enterprise multi-session, the only Windows client based operating system, that enables multiple concurrent users on a single virtual machine. Windows Virtual Desktop also provides a more consistent experience, with broader application supports compared to Windows Server based operating systems. You have investigated the features and benefits that are available using Windows Virtual Desktop. What about costs? Costs are always something you should consider before you make a decision on using any services. How can you reduce costs with Windows Virtual Desktop?

One way to reduce costs is that you can
use your existing Microsoft licenses.

Windows Virtual Desktop is
available to you at no additional cost.

If you have an eligible Microsoft 365 license,
you only pay for
the Azure resources used by Windows Virtual Desktop.

Further examples of cost savings
include, Windows 10 Enterprise,
and Windows 7 Enterprise desktops
and apps are available at
no additional cost when you present
an eligible Windows or Microsoft 365 license.

Windows Server, Remote Desktop Services desktops
and apps are also available at no additional cost,
if you are Microsoft Remote Services Client
Access License customer.

Another way to reduce costs
associated with Windows Virtual Desktop,
is to save on Compute costs.

If you buy a one-year or three-year
Azure reserved Virtual Machine instances,
you can save up to 72 percent,
versus paying as you go pricing.

You can pay for reservation upfront or monthly.

Reservations provide a billing discount and
don't affect the runtime state of your resources.

In our case study,
you saw that Windows Virtual Desktop was a great way to
expedite the deployment process for your remote workers,
while simultaneously managing the associated costs.

Let's review some of the key benefits
Windows Virtual Desktop can bring to your business.
Windows Virtual Desktop works
across devices like Windows,

Mac, iOS, Android, and Linux.

Provides virtualization on any personal device, from any Internet-connected location.

Gives you access at no additional cost when you use eligible Windows or Microsoft 365 licenses, pay only for what you use.

Keeps your Virtual Desktop secure by leveraging reverse connections on security solutions.

Virtual machines in Azure

- Article
- 03/01/2023
- 4 contributors

Feedback

In this article

1. [What do I need to think about before creating a virtual machine?](#)
2. [Availability](#)
3. [Sizes and pricing](#)
4. [Virtual machine total core limits](#)

Show 8 more

Applies to: ✓ Linux VMs ✓ Windows VMs ✓ Flexible scale sets

Azure virtual machines are one of several types of [on-demand, scalable computing resources](#) that Azure offers. Typically, you choose a virtual machine when you need more control over the computing environment than the other choices offer. This article gives you information about what you should consider before you create a virtual machine, how you create it, and how you manage it.

An Azure virtual machine gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it. However, you still need to maintain the virtual machine by performing tasks, such as configuring, patching, and installing the software that runs on it.

Azure virtual machines can be used in various ways. Some examples are:

- **Development and test** – Azure virtual machines offer a quick and easy way to create a computer with specific configurations required to code and test an application.
- **Applications in the cloud** – Because demand for your application can fluctuate, it might make economic sense to run it on a virtual machine in Azure.

- You pay for extra virtual machines when you need them and shut them down when you don't.
- **Extended datacenter** – virtual machines in an Azure virtual network can easily be connected to your organization's network.

The number of virtual machines that your application uses can scale up and out to whatever is required to meet your needs.

What do I need to think about before creating a virtual machine?

There's always a multitude of [design considerations](#) when you build out an application infrastructure in Azure. These aspects of a virtual machine are important to think about before you start:

- The names of your application resources
- The location where the resources are stored
- The size of the virtual machine
- The maximum number of virtual machines that can be created
- The operating system that the virtual machine runs
- The configuration of the virtual machine after it starts
- The related resources that the virtual machine needs

Locations

There are multiple [geographical regions](#) around the world where you can create Azure resources. Usually, the region is called **location** when you create a virtual machine. For a virtual machine, the location specifies where the virtual hard disks will be stored.

This table shows some of the ways you can get a list of available locations.

Method	Description
Azure portal	Select a location from the list when you create a virtual machine.
Azure PowerShell	Use the Get-AzLocation command.
REST API	Use the List locations operation.
Azure CLI	Use the az account list-locations operation.

Availability

There are multiple options to manage the availability of your virtual machines in Azure.

- **[Availability Zones](#)** are physically separated zones within an Azure region. Availability zones guarantee virtual machine connectivity to at least one instance at least 99.99% of the time when you've two or more instances deployed across two or more Availability Zones in the same Azure region.
- **[Virtual Machine Scale Sets](#)** let you create and manage a group of load balanced virtual machines. The number of virtual machine instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update many virtual machines. Virtual machines in a scale set can also be deployed into multiple availability zones, a single availability zone, or regionally.

For more information see [Availability options for Azure virtual machines](#) and [SLA for Azure virtual machines](#).

Sizes and pricing

The [size](#) of the virtual machine that you use is determined by the workload that you want to run. The size that you choose then determines factors such as processing power, memory, storage capacity, and network bandwidth. Azure offers a wide variety of sizes to support many types of uses.

Azure charges an [hourly price](#) based on the virtual machine's size and operating system. For partial hours, Azure charges only for the minutes used. Storage is priced and charged separately.

Virtual machine total core limits

Your subscription has default [quota limits](#) in place that could impact the deployment of many virtual machines for your project. The current limit on a per subscription basis is 20 virtual machine total cores per region. Limits can be raised by [filing a support ticket requesting an increase](#)

Managed Disks

Managed Disks handles Azure Storage account creation and management in the background for you, and ensures that you don't have to worry about the scalability limits of the storage account. You specify the disk size and the performance tier (Standard or Premium), and Azure creates and manages the disk. As you add disks or scale the virtual machine up and down, you don't have to worry about the storage being used. If you're creating new virtual machines, [use the Azure CLI](#) or the Azure portal to create virtual machines with Managed OS and data disks. If you have virtual

machines with unmanaged disks, you can [convert your virtual machines to be backed with Managed Disks](#).

You can also manage your custom images in one storage account per Azure region, and use them to create hundreds of virtual machines in the same subscription. For more information about Managed Disks, see the [Managed Disks Overview](#).

Distributions

Microsoft Azure supports a variety of Linux and Windows distributions. You can find available distributions in the [marketplace](#), Azure portal or by querying results using CLI, PowerShell and REST APIs.

This table shows some ways that you can find the information for an image.

Method	Description
Azure portal	The values are automatically specified for you when you select an image to use.
Azure PowerShell	Get-AzVMImagePublisher -Location <i>location</i> Get-AzVMImageOffer -Location <i>location</i> -Publisher <i>publisherName</i> Get-AzVMImageSku -Location <i>location</i> -Publisher <i>publisherName</i> - Offer <i>offerName</i>
REST APIs	List image publishers List image offers List image skus
Azure CLI	az vm image list-publishers --location <i>location</i> az vm image list-offers --location <i>location</i> --publisher <i>publisherName</i> az vm image list-skus --location <i>location</i> --publisher <i>publisherName</i> -- offer <i>offerName</i>

Microsoft works closely with partners to ensure the images available are updated and optimized for an Azure runtime. For more information on Azure partner offers, see the [Azure Marketplace](#)

Cloud-init

Azure supports for [cloud-init](#) across most Linux distributions that support it. we're actively working with our Linux partners in order to have cloud-init enabled images available in the Azure Marketplace. These images will make your cloud-init deployments and configurations work seamlessly with virtual machines and virtual machine scale sets.

For more information, see [Using cloud-init on Azure Linux virtual machines](#).

Storage

- [Introduction to Microsoft Azure Storage](#)
- [Add a disk to a Linux virtual machine using the azure-cli](#)
- [How to attach a data disk to a Linux virtual machine in the Azure portal](#)

Networking

- [Virtual Network Overview](#)
- [IP addresses in Azure](#)
- [Opening ports to a Linux virtual machine in Azure](#)
- [Create a Fully Qualified Domain Name in the Azure portal](#)

Service disruptions

At Microsoft, we work hard to make sure that our services are always available to you when you need them. Forces beyond our control sometimes impact us in ways that cause unplanned service disruptions.

Microsoft provides a Service Level Agreement (SLA) for its services as a commitment for uptime and connectivity. The SLA for individual Azure services can be found at [Azure Service Level Agreements](#).

Azure already has many built-in platform features that support highly available applications. For more about these services, read [Disaster recovery and high availability for Azure applications](#).

This article covers a true disaster recovery scenario, when a whole region experiences an outage due to major natural disaster or widespread service interruption. These are rare occurrences, but you must prepare for the possibility that there is an outage of an entire region. If an entire region experiences a service disruption, the locally redundant copies of your data would temporarily be unavailable. If you have enabled geo-replication, three additional copies of your Azure Storage blobs and tables are stored in a different region. In the event of a complete regional outage or a disaster in which the primary region isn't recoverable, Azure remaps all of the DNS entries to the geo-replicated region.

To help you handle these rare occurrences, we provide the following guidance for Azure virtual machines in the case of a service disruption of the entire region where your Azure virtual machine application is deployed.

Option 1: Initiate a failover by using Azure Site Recovery

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 Azure region of your choice and not restricted to paired regions. You can get started by [replicating your virtual machines](#). You can [create a recovery plan](#) so that you can automate the entire failover process for your application. You can [test your failovers](#) beforehand without impacting production application or the ongoing replication. In the event of a primary region disruption, you just [initiate a failover](#) and bring your application in target region.

Option 2: Wait for recovery

In this case, no action on your part is required. Know that we're working diligently to restore service availability. You can see the current service status on our [Azure Service Health Dashboard](#).

This is the best option if you have not set up Azure Site Recovery, read-access geo-redundant storage, or geo-redundant storage prior to the disruption. If you have set up geo-redundant storage or read-access geo-redundant storage for the storage account where your VM virtual hard drives (VHDs) are stored, you can look to recover the base image VHD and try to provision a new VM from it. This isn't a preferred option because there are no guarantees of synchronization of data. Consequently, this option isn't guaranteed to work.

Note

Be aware that you don't have any control over this process, and it will only occur for region-wide service disruptions. Because of this, you must also rely on other application-specific backup strategies to achieve the highest level of availability. For more information, see the section on [Data strategies for disaster recovery](#).

Resources for service disruptions

- Start [protecting your applications running on Azure virtual machines](#) using Azure Site Recovery
- To learn more about how to implement a disaster recovery and high availability strategy, see [Disaster recovery and high availability for Azure applications](#).
- To develop a detailed technical understanding of a cloud platform's capabilities, see [Azure resiliency technical guidance](#).
- If the instructions aren't clear, or if you would like Microsoft to do the operations on your behalf, contact [Customer Support](#).

Data residency

In Azure, the feature to enable storing customer data in a single region is currently only available in the Southeast Asia Region (Singapore) of the Asia Pacific Geo and Brazil South (Sao Paulo State) Region of Brazil Geo. For all other regions, customer data is stored in Geo. For more information, see [Trust Center](#).

What is reliability in Virtual Machines?

- Article
- 03/14/2023
- 4 contributors

Feedback

In this article

1. [Availability zone support](#)
2. [Disaster recovery: cross-region failover](#)
3. [Additional guidance](#)
4. [Next steps](#)

This article describes reliability support in Virtual Machines (VM), and covers both regional resiliency with availability zones and cross-region resiliency with disaster recovery. For a more detailed overview of reliability in Azure, see [Azure reliability](#).

Availability zone support

Azure availability zones are at least three physically separate groups of datacenters within each Azure region. Datacenters within each zone are equipped with independent power, cooling, and networking infrastructure. In the case of a local zone failure, availability zones are designed so that if the one zone is affected, regional services, capacity, and high availability are supported by the remaining two zones.

Failures can range from software and hardware failures to events such as earthquakes, floods, and fires. Tolerance to failures is achieved with redundancy and logical isolation of Azure services. For more detailed information on availability zones in Azure, see [Regions and availability zones](#).

Azure availability zones-enabled services are designed to provide the right level of reliability and flexibility. They can be configured in two ways. They can be either zone redundant, with automatic replication across zones, or zonal, with instances pinned

to a specific zone. You can also combine these approaches. For more information on zonal vs. zone-redundant architecture, see [Build solutions with availability zones](#).

Virtual machines support availability zones with three availability zones per supported Azure region and are also zone-redundant and zonal. For more information, see [availability zones support](#). The customer will be responsible for configuring and migrating their virtual machines for availability. Refer to the following readiness options below for availability zone enablement:

- See [availability options for VMs](#)
- Review [availability zone service and region support](#)
- [Migrate existing VMs](#) to availability zones

Prerequisites

Your virtual machine SKUs must be available across the zones in for your region. To review which regions support availability zones, see the [list of supported regions](#). Check for VM SKU availability by using PowerShell, the Azure CLI, or review list of foundational services. For more information, see [reliability prerequisites](#).

SLA improvements

Because availability zones are physically separate and provide distinct power source, network, and cooling, SLAs (Service-level agreements) increase. For more information, see the [SLA for Virtual Machines](#).

Create a resource with availability zone enabled

Get started by creating a virtual machine (VM) with availability zone enabled from the following deployment options below:

- [Azure CLI](#)
- [PowerShell](#)
- [Azure portal](#)

Zonal failover support

Customers can set up virtual machines to failover to another zone using the Site Recovery service. For more information, see [Site Recovery](#).

Fault tolerance

Virtual machines can failover to another server in a cluster, with the VM's operating system restarting on the new server. Customers should refer to the failover process for disaster recovery, gathering virtual machines in recovery planning, and running disaster recovery drills to ensure their fault tolerance solution is successful.

For more information, see the [site recovery processes](#).

Zone down experience

During a zone-wide outage, the customer should expect brief degradation of performance, until the virtual machine service self-healing re-balances underlying capacity to adjust to healthy zones. This isn't dependent on zone restoration; it is expected that the Microsoft-managed service self-healing state will compensate for a lost zone, leveraging capacity from other zones.

Customers should also prepare for the possibility that there's an outage of an entire region. If there's a service disruption for an entire region, the locally redundant copies of your data would temporarily be unavailable. If geo-replication is enabled, three additional copies of your Azure Storage blobs and tables are stored in a different region. In the event of a complete regional outage or a disaster in which the primary region isn't recoverable, Azure remaps all of the DNS entries to the geo-replicated region.

Zone outage preparation and recovery

The following guidance is provided for Azure virtual machines in the case of a service disruption of the entire region where your Azure virtual machine application is deployed:

- Configure [Azure Site Recovery](#) for your VMs
- Check the [Azure Service Health Dashboard](#) status if Azure Site Recovery hasn't been configured
- Review how the [Azure Backup service](#) works for VMs
 - See the [support matrix](#) for Azure VM backups
- Determine which [VM restore option and scenario](#) will work best for your environment

Low-latency design

Cross Region (secondary region), Cross Subscription (preview), and Cross Zonal (preview) are available options to consider when designing a low-latency virtual machine solution. For more information on these options, see the [supported restore methods](#).

Important

By opting out of zone-aware deployment, you forego protection from isolation of underlying faults. Use of SKUs that don't support availability zones or opting out from availability zone configuration forces reliance on resources that don't obey zone placement and separation (including underlying dependencies of these resources). These resources shouldn't be expected to survive zone-down scenarios. Solutions that leverage such resources should define a disaster recovery strategy and configure a recovery of the solution in another region.

Safe deployment techniques

When you opt for availability zones isolation, you should utilize safe deployment techniques for application code, as well as application upgrades. In addition to configuring Azure Site Recovery, below are recommended safe deployment techniques for VMs:

- [Virtual Machine Scale Sets](#)
- [Availability Sets](#)
- [Azure Load Balancer](#)
- [Azure Storage Redundancy](#)

As Microsoft periodically performs planned maintenance updates, there may be rare instances when these updates require a reboot of your virtual machine to apply the required updates to the underlying infrastructure. To learn more, see [availability considerations](#) during scheduled maintenance.

Follow the health signals below for monitoring before upgrading your next set of nodes in another zone:

- Check the [Azure Service Health Dashboard](#) for the virtual machines service status for your expected regions
- Ensure that [replication](#) is enabled on your VMs

Availability zone redeployment and migration

For migrating existing virtual machine resources to a zone redundant configuration, refer to the below resources:

- Move a VM to another subscription or resource group
 - [CLI](#)
 - [PowerShell](#)
- [Azure Resource Mover](#)
- [Move Azure VMs to availability zones](#)

- [Move region maintenance configuration resources](#)

Disaster recovery: cross-region failover

In the case of a region-wide disaster, Azure can provide protection from regional or large geography disasters with disaster recovery by making use of another region. For more information on Azure disaster recovery architecture, see [Azure to Azure disaster recovery architecture](#).

Customers can use Cross Region to restore Azure VMs via paired regions. You can restore all the Azure VMs for the selected recovery point if the backup is done in the secondary region. For more details on Cross Region restore, refer to the Cross Region table row entry in our [restore options](#).

Cross-region disaster recovery in multi-region geography

While Microsoft is working diligently to restore the virtual machine service for region-wide service disruptions, customers will have to rely on other application-specific backup strategies to achieve the highest level of availability. For more information, see the section on [Data strategies for disaster recovery](#).

Outage detection, notification, and management

When the hardware or the physical infrastructure for the virtual machine fails unexpectedly. This can include local network failures, local disk failures, or other rack level failures. When detected, the Azure platform automatically migrates (heals) your virtual machine to a healthy physical machine in the same data center. During the healing procedure, virtual machines experience downtime (reboot) and in some cases loss of the temporary drive. The attached OS and data disks are always preserved.

For more detailed information on virtual machine service disruptions, see [disaster recovery guidance](#).

Set up disaster recovery and outage detection

When setting up disaster recovery for virtual machines, understand what [Azure Site Recovery provides](#). Enable disaster recovery for virtual machines with the below methods:

- Set up disaster recovery to a [secondary Azure region for an Azure VM](#)
- Create a Recovery Services vault
 - [Bicep](#)

- [ARM template](#)
- Enable disaster recovery for [Linux virtual machines](#)
- Enable disaster recovery for [Windows virtual machines](#)
- Failover virtual machines to [another region](#)
- Failover virtual machines to the [primary region](#)

Single-region geography disaster recovery

With disaster recovery set up, Azure VMs will continuously replicate to a different target region. If an outage occurs, you can fail over VMs to the secondary region, and access them from there.

For more information, see [Azure VMs architectural components](#) and [region pairing](#).

Capacity and proactive disaster recovery resiliency

Microsoft and its customers operate under the Shared responsibility model. This means that for customer-enabled DR (customer-responsible services), the customer must address DR for any service they deploy and control. To ensure that recovery is proactive, customers should always pre-deploy secondaries because there's no guarantee of capacity at time of impact for those who haven't pre-allocated.

For deploying virtual machines, customers can use [flexible orchestration](#) mode on Virtual Machine Scale Sets. All VM sizes can be used with flexible orchestration mode. Flexible orchestration mode also offers high availability guarantees (up to 1000 VMs) by spreading VMs across fault domains in a region or within an Availability Zone.

Additional guidance

- [Well-Architected Framework for virtual machines](#)
- [Azure to Azure disaster recovery architecture](#)
- [Accelerated networking with Azure VM disaster recovery](#)
- [Express Route with Azure VM disaster recovery](#)
- [Virtual Machine Scale Sets](#)

Service disruptions

At Microsoft, we work hard to make sure that our services are always available to you when you need them. Forces beyond our control sometimes impact us in ways that cause unplanned service disruptions.

Microsoft provides a Service Level Agreement (SLA) for its services as a commitment for uptime and connectivity. The SLA for individual Azure services can be found at [Azure Service Level Agreements](#).

Azure already has many built-in platform features that support highly available applications. For more about these services, read [Disaster recovery and high availability for Azure applications](#).

This article covers a true disaster recovery scenario, when a whole region experiences an outage due to major natural disaster or widespread service interruption. These are rare occurrences, but you must prepare for the possibility that there is an outage of an entire region. If an entire region experiences a service disruption, the locally redundant copies of your data would temporarily be unavailable. If you have enabled geo-replication, three additional copies of your Azure Storage blobs and tables are stored in a different region. In the event of a complete regional outage or a disaster in which the primary region isn't recoverable, Azure remaps all of the DNS entries to the geo-replicated region.

To help you handle these rare occurrences, we provide the following guidance for Azure virtual machines in the case of a service disruption of the entire region where your Azure virtual machine application is deployed.

Option 1: Initiate a failover by using Azure Site Recovery

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 Azure region of your choice and not restricted to paired regions. You can get started by [replicating your virtual machines](#). You can [create a recovery plan](#) so that you can automate the entire failover process for your application. You can [test your failovers](#) beforehand without impacting production application or the ongoing replication. In the event of a primary region disruption, you just [initiate a failover](#) and bring your application in target region.

Option 2: Wait for recovery

In this case, no action on your part is required. Know that we're working diligently to restore service availability. You can see the current service status on our [Azure Service Health Dashboard](#).

This is the best option if you have not set up Azure Site Recovery, read-access geo-redundant storage, or geo-redundant storage prior to the disruption. If you have set up geo-redundant storage or read-access geo-redundant storage for the storage account where your VM virtual hard drives (VHDs) are stored, you can look to recover

the base image VHD and try to provision a new VM from it. This isn't a preferred option because there are no guarantees of synchronization of data. Consequently, this option isn't guaranteed to work.

Note

Be aware that you don't have any control over this process, and it will only occur for region-wide service disruptions. Because of this, you must also rely on other application-specific backup strategies to achieve the highest level of availability. For more information, see the section on [Data strategies for disaster recovery](#).

Resources for service disruptions

- Start [protecting your applications running on Azure virtual machines](#) using Azure Site Recovery
- To learn more about how to implement a disaster recovery and high availability strategy, see [Disaster recovery and high availability for Azure applications](#).
- To develop a detailed technical understanding of a cloud platform's capabilities, see [Azure resiliency technical guidance](#).
- If the instructions aren't clear, or if you would like Microsoft to do the operations on your behalf, contact [Customer Support](#).

Data residency

In Azure, the feature to enable storing customer data in a single region is currently only available in the Southeast Asia Region (Singapore) of the Asia Pacific Geo and Brazil South (Sao Paulo State) Region of Brazil Geo. For all other regions, customer data is stored in Geo. For more information, see [Trust Center](#).

An overview of Azure VM backup

- Article
- 03/24/2023
- 19 contributors

Feedback

In this article

1. [Backup process](#)
2. [Encryption of Azure VM backups](#)
3. [Snapshot creation](#)

4. [Snapshot consistency](#)

Show 4 more

This article describes how the [Azure Backup service](#) backs up Azure virtual machines (VMs).

Azure Backup provides independent and isolated backups to guard against unintended destruction of the data on your VMs. Backups are stored in a Recovery Services vault with built-in management of recovery points. Configuration and scaling are simple, backups are optimized, and you can easily restore as needed.

As part of the backup process, a [snapshot is taken](#), and the data is transferred to the Recovery Services vault with no impact on production workloads. The snapshot provides different levels of consistency, as described [here](#).

Azure Backup also has specialized offerings for database workloads like [SQL Server](#) and [SAP HANA](#) that are workload-aware, offer 15 minute RPO (recovery point objective), and allow backup and restore of individual databases.

Backup process

Here's how Azure Backup completes a backup for Azure VMs:

1. For Azure VMs that are selected for backup, Azure Backup starts a backup job according to the backup schedule you specify.
2. During the first backup, a backup extension is installed on the VM if the VM is running.
 - For Windows VMs, the [VMSnapshot extension](#) is installed.
 - For Linux VMs, the [VMSnapshotLinux extension](#) is installed.
3. For Windows VMs that are running, Backup coordinates with Windows Volume Shadow Copy Service (VSS) to take an app-consistent snapshot of the VM.
 - By default, Backup takes full VSS backups.
 - If Backup can't take an app-consistent snapshot, then it takes a file-consistent snapshot of the underlying storage (because no application writes occur while the VM is stopped).
4. For Linux VMs, Backup takes a file-consistent backup. For app-consistent snapshots, you need to manually customize pre/post scripts.
5. After Backup takes the snapshot, it transfers the data to the vault.
 - The backup is optimized by backing up each VM disk in parallel.
 - For each disk that's being backed up, Azure Backup reads the blocks on the disk and identifies and transfers only the data blocks that changed (the delta) since the previous backup.
 - Snapshot data might not be immediately copied to the vault. It might take some hours at peak times. Total backup time for a VM will be less than 24 hours for daily backup policies.

- Changes made to a Windows VM after Azure Backup is enabled on it are:
 - Microsoft Visual C++ 2013 Redistributable(x64) - 12.0.40660 is installed in the VM
 - Startup type of Volume Shadow Copy service (VSS) changed to automatic from manual
 - IaaSVmProvider Windows service is added

Encryption of Azure VM backups

When you back up Azure VMs with Azure Backup, VMs are encrypted at rest with Storage Service Encryption (SSE). Azure Backup can also back up Azure VMs that are encrypted by using Azure Disk Encryption.

Encryption Details	Support
SSE With SSE, Azure Storage provides encryption at rest by automatically encrypting data before storing it. Azure Storage also decrypts data before retrieving it. Azure Backup supports backups of VMs with two types of Storage Service Encryption: <ul style="list-style-type: none"> SSE with platform-managed keys: This encryption is by default for all disks in your VMs. See more here. SSE with customer-managed keys. With CMK, you manage the keys used to encrypt the disks. See more here. 	Azure Backup uses SSE for at-rest encryption of Azure VMs.
Azure Disk Encryption Azure Disk Encryption encrypts both OS and data disks for Azure VMs. Azure Disk Encryption integrates with BitLocker encryption keys (BEKs), which are safeguarded in a key vault as secrets. Azure Disk Encryption also integrates with Azure Key Vault key encryption keys (KEKs).	Azure Backup supports backup of managed and unmanaged Azure VMs encrypted with BEKs only, or with BEKs together with KEKs. Both BEKs and KEKs are backed up and encrypted. Because KEKs and BEKs are backed up, users with the necessary permissions can restore keys and secrets back to the key vault if needed. These users can also recover the encrypted VM.

Encryption Details	Support
	read by unauthorized users or by Azure.

For managed and unmanaged Azure VMs, Backup supports both VMs encrypted with BEKs only or VMs encrypted with BEKs together with KEKs.

The backed-up BEKs (secrets) and KEKs (keys) are encrypted. They can be read and used only when they're restored back to the key vault by authorized users. Neither unauthorized users, or Azure, can read or use backed-up keys or secrets.

BEKs are also backed up. So, if the BEKs are lost, authorized users can restore the BEKs to the key vault and recover the encrypted VMs. Only users with the necessary level of permissions can back up and restore encrypted VMs or keys and secrets.

Snapshot creation

Azure Backup takes snapshots according to the backup schedule.

- **Windows VMs:** For Windows VMs, the Backup service coordinates with VSS to take an app-consistent snapshot of the VM disks. By default, Azure Backup takes a full VSS backup (it truncates the logs of application such as SQL Server at the time of backup to get application level consistent backup). If you're using a SQL Server database on Azure VM backup, then you can modify the setting to take a VSS Copy backup (to preserve logs). For more information, see [this article](#).
- **Linux VMs:** To take app-consistent snapshots of Linux VMs, use the Linux pre-script and post-script framework to write your own custom scripts to ensure consistency.
 - Azure Backup invokes only the pre/post scripts written by you.
 - If the pre-scripts and post-scripts execute successfully, Azure Backup marks the recovery point as application-consistent. However, when you're using custom scripts, you're ultimately responsible for the application consistency.
 - [Learn more](#) about how to configure scripts.

Snapshot consistency

The following table explains the different types of snapshot consistency:

Snapshot	Details	Recovery	Consideration
Application-consistent	App-consistent backups capture memory content and pending I/O operations. App-consistent snapshots use a VSS writer (or pre/post scripts for Linux) to ensure the consistency of the app data before a backup occurs.	When you're recovering a VM with an app-consistent snapshot, the VM boots up. There's no data corruption or loss. The apps start in a consistent state.	Windows: All VSS writers succeeded Linux: Pre/post scripts are configured and succeeded
File-system consistent	File-system consistent backups provide consistency by taking a snapshot of all files at the same time.	When you're recovering a VM with a file-system consistent snapshot, the VM boots up. There's no data corruption or loss. Apps need to implement their own "fix-up" mechanism to make sure that restored data is consistent.	Windows: Some VSS writers failed Linux: Default (if pre/post scripts aren't configured or failed)
Crash-consistent	Crash-consistent snapshots typically occur if an Azure VM shuts down at the time of backup. Only the data that already exists on the disk at the time of backup is captured and backed up.	Starts with the VM boot process followed by a disk check to fix corruption errors. Any in-memory data or write operations that weren't transferred to disk before the crash are lost. Apps implement their own data verification. For example, a database app can use its transaction log for verification. If the transaction log has entries that aren't in the database, the database software rolls transactions back until the data is consistent.	VM is in shutdown (stopped/deallocated)

Note

If the provisioning state is **succeeded**, Azure Backup takes file-system consistent backups. If the provisioning state is **unavailable** or **failed**, crash-consistent backups are taken. If the provisioning state is **creating** or **deleting**, that means Azure Backup is retrying the operations.

Backup and restore considerations

Consideration Details

Disk	Backup of VM disks is parallel. For example, if a VM has four disks, the Backup service attempts to back up all four disks in parallel. Backup is incremental (only changed data).
-------------	--

Consideration Details

Scheduling To reduce backup traffic, back up different VMs at different times of the day and make sure the times don't overlap. Backing up VMs at the same time causes traffic jams.

Preparing backups Keep in mind the time needed to prepare the backup. The preparation time includes installing or updating the backup extension and triggering a snapshot according to the backup schedule.

Data transfer Consider the time needed for Azure Backup to identify the incremental changes from the previous backup.

In an incremental backup, Azure Backup determines the changes by calculating the checksum of the block. If a block is changed, it's marked for transfer to the vault. The service analyzes the identified blocks to attempt to further minimize the amount of data to transfer. After evaluating all the changed blocks, Azure Backup transfers the changes to the vault.

There might be a lag between taking the snapshot and copying it to vault. At peak times, it can take up to eight hours for the snapshots to be transferred to the vault. The backup time for a VM will be less than 24 hours for the daily backup.

Initial backup Although the total backup time for incremental backups is less than 24 hours, that might not be the case for the first backup. The time needed for the initial backup will depend on the size of the data and when the backup is processed.

Restore queue Azure Backup processes restore jobs from multiple storage accounts at the same time, and it puts restore requests in a queue.

Restore copy During the restore process, data is copied from the vault to the storage account.

The total restore time depends on the I/O operations per second (IOPS) and the throughput of the storage account.

To reduce the copy time, select a storage account that isn't loaded with other application writes and reads.

Note

Azure Backup now enables you to back up your Azure VMs multiple times a day using the Enhanced policy. With this capability, you can also define the duration in which your backup jobs would trigger and align your backup schedule with the working hours when there are frequent updates to Azure Virtual Machines. [Learn more.](#)

Backup performance

These common scenarios can affect the total backup time:

- **Adding a new disk to a protected Azure VM:** If a VM is undergoing incremental backup and a new disk is added, the backup time will increase. The total backup time might last more than 24 hours because of initial replication of the new disk, along with delta replication of existing disks.
- **Fragmented disks:** Backup operations are faster when disk changes are contiguous. If changes are spread out and fragmented across a disk, backup will be slower.
- **Disk churn:** If protected disks that are undergoing incremental backup have a daily churn of more than 200 GB, backup can take a long time (more than eight hours) to complete.
- **Backup versions:** The latest version of Backup (known as the Instant Restore version) uses a more optimized process than checksum comparison for identifying changes. But if you're using Instant Restore and have deleted a backup snapshot, the backup switches to checksum comparison. In this case, the backup operation will exceed 24 hours (or fail).

Restore performance

These common scenarios can affect the total restore time:

- The total restore time depends on the Input/output operations per second (IOPS) and the throughput of the storage account.
- The total restore time can be affected if the target storage account is loaded with other application read and write operations. To improve restore operation, select a storage account that isn't loaded with other application data.

Best practices

When you're configuring VM backups, we suggest following these practices:

- Modify the default schedule times that are set in a policy. For example, if the default time in the policy is 12:00 AM, increment the timing by several minutes so that resources are optimally used.
- If you're restoring VMs from a single vault, we highly recommend that you use different [general-purpose v2 storage accounts](#) to ensure that the target storage account doesn't get throttled. For example, each VM must have a different storage account. For example, if 10 VMs are restored, use 10 different storage accounts.
- For backup of VMs that are using premium storage with Instant Restore, we recommend allocating 50% free space of the total allocated storage space, which is required **only** for the first backup. The 50% free space isn't a requirement for backups after the first backup is complete
- The limit on the number of disks per storage account is relative to how heavily the disks are being accessed by applications that are running on an infrastructure as a service (IaaS) VM. As a general practice, if 5 to 10 disks or

- more are present on a single storage account, balance the load by moving some disks to separate storage accounts.
- To restore VMs with managed disks using PowerShell, provide the additional parameter ***TargetResourceGroupName*** to specify the resource group to which managed disks will be restored, [Learn more here](#).

Backup costs

Azure VMs backed up with Azure Backup are subject to [Azure Backup pricing](#).

Billing doesn't start until the first successful backup finishes. At this point, the billing for both storage and protected VMs begins. Billing continues as long as any backup data for the VM is stored in a vault. If you stop protection for a VM, but backup data for the VM exists in a vault, billing continues.

Billing for a specified VM stops only if the protection is stopped and all backup data is deleted. When protection stops and there are no active backup jobs, the size of the last successful VM backup becomes the protected instance size used for the monthly bill.

The protected-instance size calculation is based on the *actual* size of the VM. The VM's size is the sum of all the data in the VM, excluding the temporary storage. Pricing is based on the actual data that's stored on the data disks, not on the maximum supported size for each data disk that's attached to the VM.

Similarly, the backup storage bill is based on the amount of data that's stored in Azure Backup, which is the sum of the actual data in each recovery point.

For example, take an A2-Standard-sized VM that has two additional data disks with a maximum size of 32 TB each. The following table shows the actual data stored on each of these disks:

Disk	Max size	Actual data present
OS disk	32 TB	17 GB
Local/temporary disk	135 GB	5 GB (not included for backup)
Data disk 1	32 TB	30 GB
Data disk 2	32 TB	0 GB

The actual size of the VM in this case is $17\text{ GB} + 30\text{ GB} + 0\text{ GB} = 47\text{ GB}$. This protected-instance size (47 GB) becomes the basis for the monthly bill. As the

amount of data in the VM grows, the protected-instance size used for billing changes to match.

Use Azure portal to back up multiple virtual machines

- Article
- 02/27/2023
- 9 contributors

Feedback

In this article

1. [Sign in to the Azure portal](#)
2. [Create a Recovery Services vault](#)
3. [Set backup policy to protect VMs](#)
4. [Initial backup](#)

Show 2 more

When you back up data in Azure, you store that data in an Azure resource called a Recovery Services vault. The Recovery Services vault resource is available from the Settings menu of most Azure services. The benefit of having the Recovery Services vault integrated into the Settings menu of most Azure services is the ease of backing up data. However, working individually with each database or virtual machine in your business is tedious. What if you want to back up the data for all virtual machines in one department, or in one location? It's easy to back up multiple virtual machines by creating a backup policy and applying that policy to the desired virtual machines.

This tutorial explains how to:

- Create a Recovery Services vault
- Define a backup policy
- Apply the backup policy to protect multiple virtual machines
- Trigger an on-demand backup job for the protected virtual machines

Sign in to the Azure portal

Sign in to the [Azure portal](#).

Note

The functionality described in the following sections can also be accessed via [Backup center](#). Backup center is a single unified management experience in Azure. It enables

enterprises to govern, monitor, operate, and analyze backups at scale. With this solution, you can perform most of the key backup management operations without being limited to the scope of an individual vault.

Create a Recovery Services vault

A Recovery Services vault is a management entity that stores recovery points that are created over time, and it provides an interface to perform backup-related operations. These operations include taking on-demand backups, performing restores, and creating backup policies.

To create a Recovery Services vault:

1. Sign in to the [Azure portal](#).
2. Search for **Backup center**, and then go to the **Backup center** dashboard.
3. On the **Overview** pane, select **Vault**.
4. Select **Recovery Services vault** > **Continue**.
5. On the **Recovery Services vault** pane, enter the following values:
 - **Subscription**: Select the subscription to use. If you're a member of only one subscription, you'll see that name. If you're not sure which subscription to use, use the default subscription. There are multiple choices only if your work or school account is associated with more than one Azure subscription.
 - **Resource group**: Use an existing resource group or create a new one. To view a list of available resource groups in your subscription, select **Use existing**, and then select a resource in the dropdown list. To create a new resource group, select **Create new**, and then enter the name. For more information about resource groups, see [Azure Resource Manager overview](#).

- **Vault name:** Enter a friendly name to identify the vault. The name must be unique to the Azure subscription. Specify a name that has at least 2 but not more than 50 characters. The name must start with a letter and consist only of letters, numbers, and hyphens.
- **Region:** Select the geographic region for the vault. For you to create a vault to help protect any data source, the vault *must* be in the same region as the data source.

Important

If you're not sure of the location of your data source, close the window. Go to the list of your resources in the portal. If you have data sources in multiple regions, create a Recovery Services vault for each region. Create the vault in the first location before you create a vault in another location. There's no need to specify storage accounts to store the backup data. The Recovery Services vault and Azure Backup handle that automatically.

- 6.
7. After providing the values, select **Review + create**.
8. To finish creating the Recovery Services vault, select **Create**.

It can take a while to create the Recovery Services vault. Monitor the status notifications in the **Notifications** area at the upper right. After the vault is created, it appears in the list of Recovery Services vaults. If the vault doesn't appear, select **Refresh**.

Note

Azure Backup now supports immutable vaults that help you ensure that recovery points once created can't be deleted before their expiry as per the backup policy. You can make the immutability irreversible for maximum protection to your backup data from various threats, including ransomware attacks and malicious actors. [Learn more](#).

When you create a Recovery Services vault, by default the vault has geo-redundant storage. To provide data resiliency, geo-redundant storage replicates the data multiple times across two Azure regions.

Set backup policy to protect VMs

After creating the Recovery Services vault, the next step is to configure the vault for the type of data, and to set the backup policy. Backup policy is the schedule for how often and when recovery points are taken. Policy also includes the retention range for the recovery points. For this tutorial, let's assume your business is a sports complex with a hotel, stadium, and restaurants and concessions, and you're protecting the data on the virtual machines. The following steps create a backup policy for the financial data.

To set a backup policy to your Azure VMs, follow these steps:

1. Go to **Backup center** and click **+Backup** from the **Overview** tab.
2. Select **Azure Virtual machines** as the **Datasource type** and select the vault you have created. Then click **Continue**.
3. Assign a Backup policy.
 - The default policy backs up the VM once a day. The daily backups are retained for 30 days. Instant recovery snapshots are retained for two days.
 - If you don't want to use the default policy, select **Create New**, and create a custom policy as described in the next procedure.
4. Under **Virtual Machines**, select **Add**.
5. The **Select virtual machines** pane will open. Select the VMs you want to back up using the policy. Then select **OK**.
 - The selected VMs are validated.
 - You can only select VMs in the same region as the vault.
 - VMs can only be backed up in a single vault.

6. Note

7. All the VMs in the same region and subscription as that of the vault are available to configure backup. When configuring backup, you can browse to the virtual machine name and its resource group, even though you don't have the required permission on those VMs. If your VM is in soft deleted state, then it won't be visible in this list. If you need to re-protect the VM, then you need to wait for the soft delete period to expire or undelete the VM from the soft deleted list. For more information, see [the soft delete for VMs article](#).
8. In **Backup**, select **Enable backup**. This deploys the policy to the vault and to the VMs, and installs the backup extension on the VM agent running on the Azure VM.

After enabling backup:

- The Backup service installs the backup extension whether or not the VM is running.
- An initial backup will run in accordance with your backup schedule.
- When backups run, note that:
 - A VM that's running has the greatest chance for capturing an application-consistent recovery point.
 - However, even if the VM is turned off, it's backed up. Such a VM is known as an offline VM. In this case, the recovery point will be crash-consistent.
- Explicit outbound connectivity isn't required to allow backup of Azure VMs.

Note

You can also set Enhanced policy to back up Azure VMs multiple times a day. Learn about [Enhanced policy](#).

Initial backup

You've enabled backup for the Recovery Services vaults, but an initial backup hasn't been created. It's a disaster recovery best practice to trigger the first backup, so that your data is protected.

The initial backup will run in accordance with the schedule, but you can run it immediately as follows:

1. Go to **Backup center** and select the **Backup Instances** menu item.
2. Select **Azure Virtual machines** as the **Datasource type**. Then search for the VM that you have configured for backup.
3. Right-click the relevant row or select the more icon (...), and then click **Backup Now**.

4. In **Backup Now**, use the calendar control to select the last day that the recovery point should be retained. Then select **OK**.
5. Monitor the portal notifications. To monitor the job progress, go to **Backup center** > **Backup Jobs** and filter the list for **In progress** jobs. Depending on the size of your VM, creating the initial backup may take a while.

Clean up resources

If you plan to continue on to work with subsequent tutorials, don't clean up the resources created in this tutorial. If you don't plan to continue, use the following steps to delete all resources created by this tutorial in the Azure portal.

1. On the **myRecoveryServicesVault** dashboard, select **3** under **Backup Items** to open the Backup Items menu.
2. On the **Backup Items** menu, select **Azure Virtual Machine** to open the list of virtual machines associated with the vault.

The **Backup Items** list opens.

3. In the **Backup Items** menu, select the ellipsis to open the Context menu.
4. On the context menu, select **Stop backup** to open Stop Backup menu.
5. In the **Stop Backup** menu, select the upper drop-down menu and choose **Delete Backup Data**.
6. In the **Type the name of the Backup item** dialog, type *myVM*.
7. Once the backup item is verified (a check mark appears), **Stop backup** button is enabled. Select **Stop Backup** to stop the policy and delete the restore points.

Note

Deleted items are retained in the soft delete state for 14 days. Only after that period can the vault be deleted. For more information, see [Delete an Azure Backup Recovery Services vault](#).

8. When there are no more items in the vault, select **Delete**.

Once the vault is deleted, you'll return to the list of Recovery Services vaults.

Next steps

In this tutorial, you used the Azure portal to:

- Create a Recovery Services vault
- Set the vault to protect virtual machines
- Create a custom backup and retention policy
- Assign the policy to protect multiple virtual machines
- Trigger an on-demand back up for virtual machines

How to restore Azure VM data in Azure portal

- Article
- 06/13/2023
- 39 contributors

Feedback

In this article

1. [Restore options](#)
2. [Storage accounts](#)
3. [Before you start](#)
4. [Select a restore point](#)

Show 13 more

This article describes how to restore Azure VM data from the recovery points stored in [Azure Backup](#) Recovery Services vaults.

Restore options

Azure Backup provides several ways to restore a VM.

Restore option Details

Create a new VM Quickly creates and gets a basic VM up and running from a restore point.

You can specify a name for the VM and select the resource group and virtual network (VNet) in which it will be placed. The new VM must be created in the same region as the source VM.

If a VM restore fails because an Azure VM SKU wasn't available in the specified region of Azure, or because of any other issues, Azure Backup still restores the disks in the specified resource group.

Restore disk Restores a VM disk, which can then be used to create a new VM.

Azure Backup provides a template to help you customize and create a VM.

The restore job generates a template that you can download and use to specify custom VM settings, and create a VM.

The disks are copied to the Resource Group you specify.

Alternatively, you can attach the disk to an existing VM, or create a new VM using PowerShell.

This option is useful if you want to customize the VM, add configuration settings that weren't there at the time of backup, or add settings that must be configured using the template or PowerShell.

Replace existing You can restore a disk, and use it to replace a disk on the existing VM.

The current VM must exist. If it's been deleted, this option can't be used.

Azure Backup takes a snapshot of the existing VM before replacing the disk. The snapshot is copied to the vault and retained in accordance with the retention policy.

When choosing a Vault-Standard recovery point, a VHD file with the content of the chosen recovery point is also created in the staging location you specify. Existing disks connected to the VM are replaced with the selected restore point.

After the replace disk operation, the original disk is retained in the resource group. You can choose to manually delete the original disks if they aren't needed.

Replace existing is supported for unencrypted managed VMs, including

Restore option	Details
	<p>VMs created using custom images. It's unsupported for classic VMs, unmanaged VMs, and generalized VMs.</p> <p>If the restore point has more or less disks than the current VM, then the number of disks in the restore point will only reflect the VM configuration.</p> <p>Replace existing is also supported for VMs with linked resources, like user-assigned managed-identity or Key Vault.</p>
Cross Region (secondary region)	<p>Cross Region restore can be used to restore Azure VMs in the secondary region, which is an Azure paired region.</p> <p>You can restore all the Azure VMs for the selected recovery point if the backup is done in the secondary region.</p> <p>During the backup, snapshots aren't replicated to the secondary region. Only the data stored in the vault is replicated. So secondary region restores are only vault tier restores. The restore time for the secondary region will be almost the same as the vault tier restore time for the primary region.</p> <p>This feature is available for the options below:</p> <ul style="list-style-type: none"> - Create a VM - Restore Disks <p>We don't currently support the Replace existing disks option.</p> <p>Permissions The restore operation on secondary region can be performed by Backup Admins and App admins.</p>
Cross Subscription Restore (preview)	<p>Allows you to restore Azure Virtual Machines or disks to any subscription (as per the Azure RBAC capabilities) from restore points.</p> <p>You can trigger Cross Subscription Restore for managed virtual machines only.</p> <p>Cross Subscription Restore is supported for Restore with Managed System Identities (MSI).</p> <p>It's unsupported for snapshots and secondary region restores.</p> <p>It's unsupported for unmanaged VMs, Encrypted Azure VMs and Trusted Launch VMs.</p>
Cross Zonal Restore	Allows you to restore Azure Virtual Machines or disks pinned to any zone to different available zones (as per the Azure RBAC capabilities) from restore points.

Restore option Details

You can trigger Cross Zonal Restore for managed virtual machines only.

Cross Zonal Restore is supported for [Restore with Managed System Identities \(MSI\)](#).

Cross Zonal Restore supports restore of an Azure zone pinned/non-zone pinned VM from a vault with Zonal-redundant storage (ZRS) enabled. Learn [how to set Storage Redundancy](#).

It's supported to restore an Azure zone pinned VM only from a [vault with Cross Region Restore \(CRR\)](#) (if the secondary region supports zones) or Zone Redundant Storage (ZRS) enabled.

Cross Zonal Restore is supported from [secondary regions](#).

It's unsupported from [snapshots](#) restore point.

It's unsupported for [Encrypted Azure VMs](#) and [Trusted Launch VMs](#).

Tip

To receive alerts/notifications when a restore operation fails, use [Azure Monitor alerts for Azure Backup](#). This helps you to monitor such failures and take necessary actions to remediate the issues.

Note

You can also recover specific files and folders on an Azure VM. [Learn more](#).

Storage accounts

Some details about storage accounts:

- **Create VM:** When creating a new VM with managed disks, nothing is placed in the storage account you specify. If using unmanaged disks, the VHD files for the VM's disks will be placed in the storage account you specify.
- **Restore disk:** The restore job generates a template that you can download and use to specify custom VM settings. This template is placed in the specified storage account. VHD files are also copied to the storage account when you restore managed disks from a Vault-Standard recovery point if the disk size is less than 4 TB, or when you restore unmanaged disks.
- **Replace disk:** When you replace a managed disk from a Vault-Standard recovery point and the disk size is less than 4 TB, a VHD file with the data from the chosen recovery point is created in the specified storage account. After the replace disk operation, the disks of the source Azure VM are left in the specified

Resource group for your operation and the VHDs are stored in the specified storage account. You can choose to delete or retain these VHDs and disks.

- **Storage account location:** The storage account must be in the same region as the vault. Only these accounts are displayed. If there are no storage accounts in the location, you need to create one.
- **Storage type:** Blob storage isn't supported.
- **Storage redundancy:** Zone redundant storage (ZRS) isn't supported. The replication and redundancy information for the account is shown in parentheses after the account name.
- **Premium storage:**
 - When you restore non-premium VMs, premium storage accounts aren't supported.
 - When you restore managed VMs, premium storage accounts configured with network rules aren't supported.

Before you start

To restore a VM (create a new VM), make sure you have the correct Azure role-based access control (Azure RBAC) [permissions](#) for the Restore VM operation.

If you don't have permissions, you can [restore a disk](#), and then after the disk is restored, you can [use the template](#) that was generated as part of the restore operation to create a new VM.

Note

The functionality described in the following sections can also be accessed via [Backup center](#). Backup center is a single unified management experience in Azure. It enables enterprises to govern, monitor, operate, and analyze backups at scale. With this solution, you can perform most of the key backup management operations without being limited to the scope of an individual vault.

Select a restore point

1. Navigate to **Backup center** in the Azure portal and click **Restore** from the **Overview** tab.
2. Select **Azure Virtual machines** as the **Datasource type**, and then select a Backup instance.

3. Select a VM and click **Continue**.
4. In the next screen that appears, select a restore point to use for the recovery.

Choose a VM restore configuration

1. In **Restore Virtual Machine**, select a restore option:
 - **Create new**: Use this option if you want to create a new VM. You can create a VM with simple settings, or restore a disk and create a customized VM.
 - **Replace existing**: Use this option if you want to replace disks on an existing VM.
2. Specify settings for your selected restore option.

Create a VM

As one of the [restore options](#), you can create a VM quickly with basic settings from a restore point.

1. In **Restore Virtual Machine > Create new > Restore Type**, select **Create new virtual machine**.
2. In **Virtual machine name**, specify a VM that doesn't exist in the subscription.
3. In **Resource group**, select an existing resource group for the new VM, or create a new one with a globally unique name. If you assign a name that already exists, Azure assigns the group the same name as the VM.
4. In **Virtual network**, select the VNet in which the VM will be placed. All VNets associated with the subscription in the same location as the vault, which is active and not attached with any affinity group, are displayed. Select the subnet.

The first subnet is selected by default.

5. In **Staging Location**, specify the storage account for the VM. [Learn more](#).

6. Choose the required subscription from the **Subscription** drop-down list to restore an Azure VM to a different subscription.

Azure Backup now supports Cross Subscription Restore (CSR), you can now restore an Azure VM using a recovery point from default subscription to another. Default subscription is the subscription where recovery point is available.

The following screenshot lists all subscriptions under the tenant where you've permissions, which enable you to restore the Azure VM to another subscription.

7. Choose the required zone from the **Availability Zone** drop-down list to restore an Azure VM pinned to any zone to a different zone.

Azure Backup now supports Cross Zonal Restore (CZR), you can now restore an Azure VM from the default zone to any available zones. Default zone is the zone in which Azure VM is running.

The following screenshot lists all zones that enable you to restore Azure VM to another zone.

Note

Azure Backup supports CZR only for vaults with ZRS or CRR redundancy.

8. Select **Restore** to trigger the restore operation.

Note

Before you modify any NSG settings, ensure the VM restore operation is complete. Learn about [tracking the restore operation](#).

Restore disks

As one of the [restore options](#), you can create a disk from a restore point. Then with the disk, you can do one of the following actions:

- Use the template that's generated during the restore operation to customize settings, and trigger VM deployment. You edit the default template settings, and submit the template for VM deployment.
 - [Attach restored disks](#) to an existing VM.
 - [Create a new VM](#) from the restored disks using PowerShell.
1. In **Restore configuration > Create new > Restore Type**, select **Restore disks**.
 2. In **Resource group**, select an existing resource group for the restored disks, or create a new one with a globally unique name.
 3. In **Staging location**, specify the storage account. The template file is stored here, and VHD files are also created in some scenarios. [Learn more](#).

4. Choose the required subscription from the **Subscription** drop-down list to restore the VM disks to a different subscription.

Azure Backup now supports Cross Subscription Restore (CSR). Like Azure VM, you can now restore Azure VM disks using a recovery point from default subscription to another. Default subscription is the subscription where recovery point is available.

5. Choose the required zone from the **Availability Zone** drop-down list to restore the VM disks to a different zone.

Azure Backup now supports Cross Zonal Restore (CZR). Like Azure VM, you can now restore Azure VM disks from the default zone to any available zones. Default zone is the zone in which the VM disks reside.

Note

Azure Backup supports CZR only for vaults with ZRS or CRR redundancy.

6. Select **Restore** to trigger the restore operation.

When your virtual machine uses managed disks and you select the **Create virtual machine** option, Azure Backup doesn't use the specified storage account. In the case of **Restore disks** and **Instant Restore**, the storage account is used only for storing the template. Managed disks are created in the specified resource group. When your virtual machine uses unmanaged disks, they're restored as blobs to the storage account.

While you restore disks for a Managed VM from a Vault-Standard recovery point, it restores the Managed disk and Azure Resource Manager (ARM) templates, along with the VHD files of the disks in staging location. If you restore disks from an Instant recovery point, it restores the Managed disks and ARM templates only.

Note

- For restoring disk from a Vault-Standard recovery point that is/was greater than 4 TB, Azure Backup doesn't restore the VHD files.
- For information on managed/premium disk performance after restored via Azure Backup, see the [Latency](#) section.

Use templates to customize a restored VM

After the disk is restored, use the template that was generated as part of the restore operation to customize and create a new VM:

1. In **Backup Jobs**, select the relevant restore job.
2. In **Restore**, select **Deploy Template** to initiate template deployment.

Note

For a shared access signature (SAS) that has **Allow storage account key access** set to disabled, the template won't deploy when you select **Deploy Template**.

3. To customize the VM setting provided in the template, select **Edit template**. If you want to add more customizations, select **Edit parameters**.
 - [Learn more](#) about deploying resources from a custom template.
 - [Learn more](#) about authoring templates.
4. Enter the custom values for the VM, accept the **Terms and Conditions** and select **Purchase**.

Replace existing disks

As one of the [restore options](#), you can replace an existing VM disk with the selected restore point. [Review](#) all restore options.

1. In **Restore configuration**, select **Replace existing**.
2. In **Restore Type**, select **Replace disk/s**. This is the restore point that will be used replace existing VM disks.
3. In **Staging Location**, specify a storage account. VHD files are created here in some scenarios. [Learn more](#).

Cross Region Restore

As one of the [restore options](#), Cross Region Restore (CRR) allows you to restore Azure VMs in a secondary region, which is an Azure paired region.

To begin using the feature, read the [Before You Begin section](#).

To see if CRR is enabled, follow the instructions in [Configure Cross Region Restore](#).

Note

Cross-region restore is currently not supported for machines running on Ultra disks. [Learn more about Ultra disk backup supportability](#).

View backup items in secondary region

If CRR is enabled, you can view the backup items in the secondary region.

1. From the portal, go to **Recovery Services vault > Backup items**.
2. Select **Secondary Region** to view the items in the secondary region.

Note

Only Backup Management Types supporting the CRR feature will be shown in the list. Currently, only support for restoring secondary region data to a secondary region is allowed.

CRR for Azure VMs is supported for Azure Managed VMs (including encrypted Azure VMs). See the [management types that support Cross Region Restore](#).

Restore in secondary region

The secondary region restore user experience will be similar to the primary region restore user experience. When configuring details in the Restore Configuration pane to configure your restore, you'll be prompted to provide only secondary region parameters.

Currently, secondary region [RPO](#) is *36 hours*. This is because the RPO in the primary region is *24 hours* and can take up to *12 hours* to replicate the backup data from the primary to the secondary region.

- To restore and create a VM, refer to [Create a VM](#).
- To restore as a disk, refer to [Restore disks](#).

Note

- The Cross Region Restore feature restores CMK (customer-managed keys) enabled Azure VMs, which aren't backed-up in a CMK enabled Recovery Services vault, as non-CMK enabled VMs in the secondary region.
- The Azure roles needed to restore in the secondary region are the same as those in the primary region.
- While restoring an Azure VM, Azure Backup configures the virtual network settings in the secondary region automatically. If you are [restoring disks](#) while deploying the template, ensure to provide the virtual network settings, corresponding to the secondary region.
- If VNet/Subnet is not available in the primary region or is not configured in the secondary region, Azure portal doesn't auto-populate any default values during restore operation.
- For Cross Region Restores, the **Staging Location** (that is the storage account location) must be in the region that the Recovery Services vault treats as the *secondary* region. For example, a Recovery Services vault is located in East US 2 region (with Geo-Redundancy and Cross Region Restore enabled). This means that the *secondary* region would be *Central US*. Therefore, you need to create a storage account in *Central US* to perform a Cross Region Restore of the VM.

Learn more about [Azure cross-region replication pairings for all geographies](#).

[Azure zone pinned VMs](#) can be restored in any [availability zones](#) of the same region.

In the restore process, you'll see the option **Availability Zone**. You'll see your default zone first. To choose a different zone, choose the number of the zone of your choice. If the pinned zone is unavailable, you won't be able to restore the data to another zone because the backed-up data isn't zonally replicated. The restore in availability zones is possible from recovery points in vault tier only.

In summary, the **Availability Zone** will only appear when

- The source VM is zone pinned and is NOT encrypted
- The recovery point is present in vault tier only (Snapshots only or snapshot and vault tier are not supported)
- The recovery option is to either create a new VM or to restore disks (replace disks option replaces source data and hence the availability zone option is not applicable)
- Creating VM/disks in the same region when vault's storage redundancy is ZRS (Doesn't work when vault's storage redundancy is GRS even though the source VM is zone pinned)
- Creating VM/disks in the paired region when vault's storage redundancy is enabled for Cross-Region-Restore AND if the paired region supports zones

Note

Cross region restore jobs once triggered, can't be canceled.

Monitoring secondary region restore jobs

1. From the portal, go to **Recovery Services vault > Backup Jobs**
2. Select **Secondary Region** to view the items in the secondary region.

Cross Subscription Restore (preview)

Azure Backup now allows you to perform Cross Subscription Restore (CSR), which helps you to restore Azure VMs in a subscription that is different from the default one. Default subscription contains the recovery points.

This feature is enabled for Recovery Services vault by default. However, there may be instances when you may need to block Cross Subscription Restore based on your

cloud infrastructure. So, you can enable, disable, or permanently disable Cross Subscription Restore for the existing vaults by going to *Vault > Properties > Cross Subscription Restore*.

Note

- CSR once permanently disabled on a vault can't be re-enabled because it's an irreversible operation.
- If CSR is disabled but not permanently disabled, then you can reverse the operation by selecting *Vault > Properties > Cross Subscription Restore > Enable*.
- If a Recovery Services vault is moved to a different subscription when CSR is disabled or permanently disabled, restore to the original subscription fails.

Restoring unmanaged VMs and disks as managed

You're provided with an option to restore [unmanaged disks](#) as [managed disks](#) during restore. By default, the unmanaged VMs / disks are restored as unmanaged VMs / disks. However, if you choose to restore as managed VMs / disks, it's now possible to do so. These restore operations aren't triggered from the snapshot phase but only from the vault phase. This feature isn't available for unmanaged encrypted VMs.

Restore VMs with special configurations

There are many common scenarios in which you might need to restore VMs.

Scenario	Guidance
Restore VMs using Hybrid Use Benefit	If a Windows VM uses Hybrid Use Benefit (HUB) licensing , restore the disks, and create a new VM using the provided template (with License Type set to Windows_Server), or PowerShell. This setting can also be applied after creating the VM.
Restore VMs during a datacenter disaster	If the vault uses GRS and the primary datacenter for the VM goes down, Azure Backup supports restoring backed-up VMs to the paired datacenter. You select a storage account in the paired datacenter, and restore as normal. Azure Backup uses the compute service in the paired region to create the restored VM. Learn more about datacenter resiliency.

If the vault uses GRS, you can choose the new feature, [Cross Region Restore](#).

Scenario	Guidance
	This lets you restore to a second region in either full or partial outage scenarios, or even if there's no outage at all.
Bare-metal restore	The major difference between Azure VMs and on-premises hypervisors is that there's no VM console available in Azure. A console is required for certain scenarios, such as recovering by using a bare-metal recovery (BMR)-type backup. However, VM restore from the vault is a full replacement for BMR.
Restore VMs with special network configurations	Special network configurations include VMs using internal or external load balancing, using multiple NICs, or multiple reserved IP addresses. You restore these VMs by using the restore disk option . This option makes a copy of the VHDs into the specified storage account, and you can then create a VM with an internal or external load balancer, multiple NICs , or multiple reserved IP addresses , in accordance with your configuration.
Network Security Group (NSG) on NIC/Subnet	Azure VM backup supports Backup and Restore of NSG information at vnet, subnet, and NIC level.
Zone Pinned VMs	If you back up an Azure VM that's pinned to a zone (with Azure Backup), then you can restore it in the same zone where it was pinned. Learn more
Restore VM in any availability set	When you restore a VM from the portal, there's no option to choose an availability set. A restored VM doesn't have an availability set. If you use the restore disk option, then you can specify an availability set when you create a VM from the disk using the provided template or PowerShell.
Restore special VMs such as SQL VMs	If you're backing up a SQL VM using Azure VM backup and then use the restore VM option or create a VM after restoring disks, then the newly created VM must be registered with the SQL provider as mentioned here . This will convert the restored VM into a SQL VM.

Restore domain controller VMs

Scenario	Guidance
Restore a single domain controller VM in a single domain	Restore the VM like any other VM. Note that: From an Active Directory perspective, the Azure VM is like any other VM. Directory Services Restore Mode (DSRM) is also available, so all Active Directory recovery scenarios are viable. Learn more about backup and restore considerations for virtualized domain controllers.

Scenario	Guidance
Restore multiple domain controller VMs in a single domain	If other domain controllers in the same domain can be reached over the network, the domain controller can be restored like any VM. If it's the last remaining domain controller in the domain, or a recovery in an isolated network is performed, use a forest recovery .
Restore a single domain controller VM in a multiple domain configuration	Restore the disks and create a VM by using PowerShell
Restore multiple domains in one forest	We recommend a forest recovery .

For more information, see [Back up and restore Active Directory domain controllers](#).

Restore VMs with managed identities

Managed identities eliminate the need for the user to maintain the credentials. Managed identities provide an identity for applications to use when connecting to resources that support Azure Active Directory (Azure AD) authentication.

Azure Backup offers the flexibility to restore the managed Azure VM with [managed identities](#). You can choose to select [system-managed identities](#) or user-managed identities as shown in the figure below. This is introduced as one of the input parameters in the [Restore configuration blade](#) of Azure VM. Managed identities used as one of the input parameters is only used for accessing the storage accounts, which are used as staging location during restore and not for any other Azure resource controlling. These managed identities have to be associated to the vault.

If you choose to select system-assigned or user-assigned managed identities, check for the below actions for managed identity on the target staging Storage Account.

JSONCopy

```
"permissions": [
  {
    "actions": [
      "Microsoft.Authorization/*/read",
      "Microsoft.Storage/storageAccounts/blobServices/containers/delete",
      "Microsoft.Storage/storageAccounts/blobServices/containers/read",
      "Microsoft.Storage/storageAccounts/blobServices/containers/write"
    ],
  }
]
```

```
        "notActions": [],
        "dataActions": [
            "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/delete",
            "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read",
            "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/write",
            "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/add/action"
        ],
        "notDataActions": []
    }
```

Or, add the role assignment on the staging location (Storage Account) to have [Storage account Backup Contributor](#) and [Storage Blob data Contributor](#) for the successful restore operation.

You can also select the [user-managed identity](#) by providing the input as their MSI Resource ID as provided in the figure below.

Note

The support is available for only managed VMs, and not supported for classic VMs and unmanaged VMs. For the [storage accounts that are restricted with firewalls](#), system MSI is only supported.

Cross Region Restore isn't supported with managed identities.

Currently, this is available in all Azure public and national cloud regions.

Track the restore operation

After you trigger the restore operation, the backup service creates a job for tracking. Azure Backup displays notifications about the job in the portal. If they aren't visible, select the **Notifications** symbol, and then select **More events in the activity log** to see the Restore Process Status.

Track restore as follows:

1. To view operations for the job, select the notifications hyperlink. Alternatively, in the vault, select **Backup jobs**, and then select the relevant VM.
2. To monitor restore progress, select any restore job with a status of **In-progress**. This displays the progress bar, which displays information about the restore progress:
 - **Estimated time of restore:** Initially provides the time taken to complete the restore operation. As the operation progresses, the time taken reduces and reaches zero when the restore operation finishes.
 - **Percentage of restore.** Shows the percentage of restore operation that's done.
 - **Number of bytes transferred:** If you're restoring by creating a new VM, it shows the bytes that were transferred against the total number of bytes to be transferred.

Post-restore steps

There are a few things to note after restoring a VM:

- Extensions present during the backup configuration are installed, but not enabled. If you see an issue, reinstall the extensions. In the case of disk replacement, reinstallation of extensions is not required.
- If the backed-up VM had a static IP address, the restored VM will have a dynamic IP address to avoid conflict. You can [add a static IP address to the restored VM](#).
- A restored VM doesn't have an availability set. If you use the restore disk option, then you can [specify an availability set](#) when you create a VM from the disk using the provided template or PowerShell.
- If you use a cloud-init-based Linux distribution, such as Ubuntu, for security reasons the password is blocked after the restore. Use the VMAccess extension on the restored VM to [reset the password](#). We recommend using SSH keys on these distributions, so you don't need to reset the password after the restore.
- If you're unable to access a VM once restored because the VM has a broken relationship with the domain controller, then follow the steps below to bring up the VM:
 - Attach OS disk as a data disk to a recovered VM.
 - Manually install VM agent if Azure Agent is found to be unresponsive by following this [link](#).

- Enable Serial Console access on VM to allow command-line access to VM

Windows Command PromptCopy

```
bcdeedit /store <drive letter>:\boot\bcd /enum
bcdeedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd
/set {bootmgr} displaybootmenu yes
bcdeedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd
/set {bootmgr} timeout 5
bcdeedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd
/set {bootmgr} bootems yes
bcdeedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd
/ems {<<BOOT LOADER IDENTIFIER>>} ON
bcdeedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd
/emssettings EMSPORT:1 EMSBAUDRATE:115200
```

- When the VM is rebuilt use Azure portal to reset local administrator account and password
- Use Serial console access and CMD to disjoin VM from domain

Windows Command PromptCopy

```
cmd /c "netdom remove <<MachineName>> /domain:<<DomainName>>
/userD:<<DomainAdminhere>> /passwordD:<<PasswordHere>>
/reboot:10 /Force"
```

- Once the VM is disjoined and restarted, you'll be able to successfully RDP to the VM with local admin credentials and rejoin VM back to domain successfully.

Backing up restored VMs

- If you restored a VM to the same resource group with the same name as the originally backed-up VM, backup continues on the VM after restore.
- If you restored the VM to a different resource group or you specified a different name for the restored VM, you need to set up backup for the restored VM.

About Site Recovery

- Article
- 02/01/2023
- 19 contributors

Feedback

In this article

1. [What does Site Recovery provide?](#)
2. [What can I replicate?](#)

3. [Next steps](#)

Welcome to the Azure Site Recovery service! This article provides a quick service overview.

As an organization, you need to adopt a business continuity and disaster recovery (BCDR) strategy that keeps your data safe, and your apps and workloads online, when planned and unplanned outages occur.

Azure Recovery Services contributes to your BCDR strategy:

- **Site Recovery service:** Site Recovery helps ensure business continuity by keeping business apps and workloads running during outages. Site Recovery [replicates](#) workloads running on physical and virtual machines (VMs) from a primary site to a secondary location. When an outage occurs at your primary site, you fail over to a secondary location, and access apps from there. After the primary location is running again, you can fail back to it.
- **Backup service:** The [Azure Backup](#) service keeps your data safe and recoverable.

Site Recovery can manage replication for:

- Azure VMs replicating between Azure regions
- Replication from Azure Public Multi-Access Edge Compute (MEC) to the region
- Replication between two Azure Public MECs
- On-premises VMs, Azure Stack VMs, and physical servers

Note

The Azure Site Recovery functionality for Public MEC is in preview state.

What does Site Recovery provide?

Feature	Details
Simple BCDR solution	Using Site Recovery, you can set up and manage replication, failover, and fallback from a single location in the Azure portal.
Azure VM replication	You can set up disaster recovery of Azure VMs from a primary region to a secondary region or from Azure Public MEC to the Azure region or from one Azure Public MEC to another Azure Public MEC connected to the same Azure region.
VMware VM replication	You can replicate VMware VMs to Azure using the improved Azure Site Recovery replication appliance that offers better security and resilience than the configuration server. For more information, see Disaster recovery of VMware VMs .

Feature	Details
On-premises VM replication	You can replicate on-premises VMs and physical servers to Azure, or to a secondary on-premises datacenter. Replication to Azure eliminates the cost and complexity of maintaining a secondary datacenter.
Workload replication	Replicate any workload running on supported Azure VMs, on-premises Hyper-V and VMware VMs, and Windows/Linux physical servers.
Data resilience	Site Recovery orchestrates replication without intercepting application data. When you replicate to Azure, data is stored in Azure storage, with the resilience that provides. When failover occurs, Azure VMs are created based on the replicated data. This also applies to Public MEC to Azure region Azure Site Recovery scenario. In case of Azure Public MEC to Public MEC Azure Site Recovery scenario (the ASR functionality for Public MEC is in preview state), data is stored in the Public MEC.
RTO and RPO targets	Keep recovery time objectives (RTO) and recovery point objectives (RPO) within organizational limits. Site Recovery provides continuous replication for Azure VMs and VMware VMs, and replication frequency as low as 30 seconds for Hyper-V. You can reduce RTO further by integrating with Azure Traffic Manager .
Keep apps consistent over failover	You can replicate using recovery points with application-consistent snapshots. These snapshots capture disk data, all data in memory, and all transactions in process.
Testing without disruption	You can easily run disaster recovery drills, without affecting ongoing replication.
Flexible failovers	You can run planned failovers for expected outages with zero-data loss. Or, unplanned failovers with minimal data loss, depending on replication frequency, for unexpected disasters. You can easily fail back to your primary site when it's available again.
Customized recovery plans	Using recovery plans, you can customize and sequence the failover and recovery of multi-tier applications running on multiple VMs. You group machines together in a recovery plan, and optionally add scripts and manual actions. Recovery plans can be integrated with Azure Automation runbooks. Note: This functionality is currently supported for Region-to-Region replication and will be available on Azure Public MEC soon.
BCDR integration	Site Recovery integrates with other BCDR technologies. For example, you can use Site Recovery to protect the SQL Server backend of corporate workloads, with native support for SQL Server Always On, to manage the failover of availability groups.
Azure automation integration	A rich Azure Automation library provides production-ready, application-specific scripts that can be downloaded and integrated with Site Recovery.

Feature	Details
Network integration	Site Recovery integrates with Azure for application network management. For example, to reserve IP addresses, configure load-balancers, and use Azure Traffic Manager for efficient network switchovers.

What can I replicate?

Supported	Details
Replication scenarios	<p>Replicate Azure VMs from</p> <ol style="list-style-type: none"> One Azure region to another. Azure Public MEC to the Azure region it's connected to. One Azure Public MEC to another Public MEC connected to same Azure region.
	Replicate on-premises VMware VMs, Hyper-V VMs, physical servers (Windows and Linux), Azure Stack VMs to Azure.
	Replicate AWS Windows instances to Azure.
	Replicate on-premises VMware VMs, Hyper-V VMs managed by System Center VMM, and physical servers to a secondary site.
Regions	Review supported regions for Site Recovery.
Replicated machines	Review the replication requirements for Azure VM replication, on-premises VMware VMs and physical servers , and on-premises Hyper-V VMs .
Workloads	You can replicate any workload running on a machine that's supported for replication. And, the Site Recovery team did app-specific tests for a number of apps .

Overview of VM restore points

- Article
- 11/18/2022
- 6 contributors

Feedback

In this article

- [About VM restore points](#)
- [Restore points for VMs inside Virtual Machine Scale Set and Availability Set \(AvSet\)](#)
- [Limitations](#)
- [Troubleshoot VM restore points](#)
- [Next steps](#)

Business continuity and disaster recovery (BCDR) solutions are primarily designed to address site-wide data loss. Solutions that operate at this scale will often manage and execute automated failovers and failbacks across multiple regions. Azure VM restore points can be used to implement granular backup and retention policies.

You can protect your data and guard against extended downtime by creating virtual machine (VM) restore points at regular intervals. There are several backup options available for virtual machines (VMs), depending on your use-case. For more information, see [Backup and restore options for virtual machines in Azure](#).

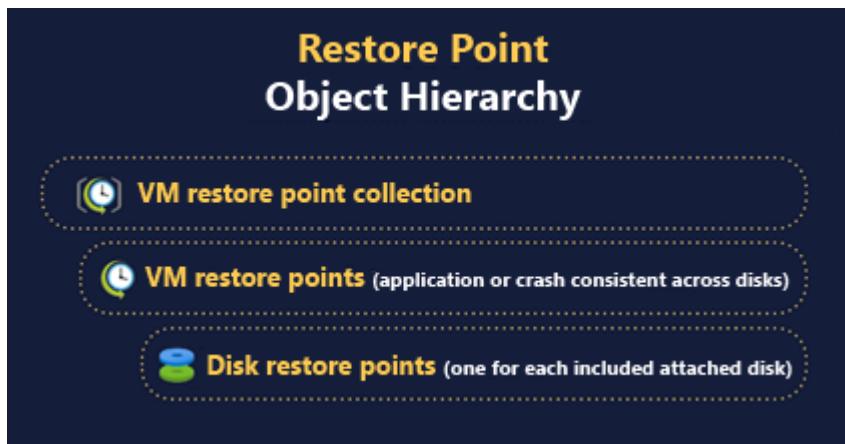
About VM restore points

An individual VM restore point is a resource that stores VM configuration and point-in-time application consistent snapshots of all the managed disks attached to the VM. You can use VM restore points to easily capture multi-disk consistent backups. VM restore points contain a disk restore point for each of the attached disks and a disk restore point consists of a snapshot of an individual managed disk.

VM restore points support application consistency for VMs running Windows operating systems and support file system consistency for VMs running Linux operating system. Application consistent restore points use VSS writers (or pre/post scripts for Linux) to ensure the consistency of the application data before a restore point is created. To get an application consistent restore point, the application running in the VM needs to provide a VSS writer (for Windows), or pre and post scripts (for Linux) to achieve application consistency.

VM restore points are organized into restore point collections. A restore point collection is an Azure Resource Management resource that contains the restore points for a specific VM. If you want to utilize ARM templates for creating restore points and restore point collections, visit the public [Virtual-Machine-Restore-Points](#) repository on GitHub.

The following image illustrates the relationship between restore point collections, VM restore points, and disk restore points.



VM restore points are incremental. The first restore point stores a full copy of all disks attached to the VM. For each successive restore point for a VM, only the incremental changes to your disks are backed up. To reduce your costs, you can optionally exclude any disk when creating a restore point for your VM.

Restore points for VMs inside Virtual Machine Scale Set and Availability Set (AvSet)

Currently, restore points can only be created in one VM at a time, that is, you cannot create a single restore point across multiple VMs. Due to this limitation, we currently support creating restore points for individual VMs with a Virtual Machine Scale Set in Flexible Orchestration mode, or Availability Set. If you want to back up instances within a Virtual Machine Scale Set instance or your Availability Set instance, you must individually create restore points for all the VMs that are part of the instance.

Note

Virtual Machine Scale Set with Uniform orchestration is not supported by restore points. You cannot create restore points of VMs inside a Virtual Machine Scale Set with Uniform orchestration.

Limitations

- Restore points are supported only for managed disks.
- Ultra-disks, Ephemeral OS disks, and Shared disks are not supported.
- Restore points APIs require an API of version 2021-03-01 or later.
- A maximum of 500 VM restore points can be retained at any time for a VM, irrespective of the number of restore point collections.
- Concurrent creation of restore points for a VM is not supported.
- Restore points for Virtual Machine Scale Sets in Uniform orchestration mode are not supported.

- Movement of Virtual Machines (VM) between Resource Groups (RG), or Subscriptions is not supported when the VM has restore points. Moving the VM between Resource Groups or Subscriptions will not update the source VM reference in the restore point and will cause a mismatch of ARM IDs between the actual VM and the restore points.

Note

Public preview of cross-region creation and copying of VM restore points is available, with the following limitations:

- Private links are not supported when copying restore points across regions or creating restore points in a region other than the source VM.
- Customer-managed key encrypted restore points, when copied to a target region or created directly in the target region are created as platform-managed key encrypted restore points.

Troubleshoot VM restore points

Most common restore points failures are attributed to the communication with the VM agent and extension, and can be resolved by following the troubleshooting steps listed in the [troubleshooting](#) article.

Manage VM restore points

- Article
- 01/12/2023
- 3 contributors

Feedback

In this article

1. [Copy a VM restore point between regions](#)
2. [Get restore point copy or replication status](#)
3. [Create a disk using disk restore points](#)
4. [Restore a VM with a restore point](#)

Show 2 more

This article explains how to copy and restore a VM from a VM restore point and track the progress of the copy operation. This article also explains how to create a disk from a disk restore point and to create a shared access signature for a disk.

Copy a VM restore point between regions

The VM restore point APIs can be used to restore a VM in a different region than the source VM. Use the following steps:

Step 1: Create a destination VM restore point collection

To copy an existing VM restore point from one region to another, your first step is to create a restore point collection in the target or destination region. To do this, reference the restore point collection from the source region as detailed in [Create a VM restore point collection](#).

Step 2: Create the destination VM restore point

After the restore point collection is created, trigger the creation of a restore point in the target restore point collection. Ensure that you've referenced the restore point in the source region that you want to copy and specified the source restore point's identifier in the request body. The source VM's location is inferred from the target restore point collection in which the restore point is being created. See the [Restore Points - Create](#) API documentation to create a RestorePoint.

Step 3: Track copy status

To track the status of the copy operation, follow the guidance in the [Get restore point copy or replication status](#) section below. This is only applicable for scenarios where the restore points are copied to a different region than the source VM.

Get restore point copy or replication status

Creation of a cross-region VM restore point is a long running operation. The VM restore point can be used to restore a VM only after the operation is completed for all disk restore points. To track the operation's status, call the [Restore Point - Get](#) API on the target VM restore point and include the `instanceView` parameter. The return will include the percentage of data that has been copied at the time of the request.

During restore point creation, the `ProvisioningState` will appear as `Creating` in the response. If creation fails, `ProvisioningState` is set to `Failed`.

Create a disk using disk restore points

You can use the VM restore points APIs to restore a VM disk, which can then be used to create a new VM. Use the following steps:

Step 1: Retrieve disk restore point identifiers

Call the [Restore Point Collections - Get](#) API on the restore point collection to get access to associated restore points and their IDs. Each VM restore point will in turn contain individual disk restore point identifiers.

Step 2: Create a disk

After you have the list of disk restore point IDs, you can use the [Disks - Create Or Update](#) API to create a disk from the disk restore points. You can choose a zone while creating the disk. The zone can be different from zone in which the disk restore point exists.

Restore a VM with a restore point

To restore a full VM from a VM restore point, you must restore individual disks from each disk restore point. This process is described in the [Create a disk](#) section. After you restore all the disks, create a new VM and attach the restored disks to the new VM. You can also use the [ARM template](#) to restore a full VM along with all the disks.

Get a shared access signature for a disk

To create a Shared Access Signature (SAS) for a disk within a VM restore point, pass the ID of the disk restore points via the `BeginGetAccess` API. If no active SAS exists on the restore point snapshot, a new SAS is created. The new SAS URL is returned in the response. If an active SAS already exists, the SAS duration is extended, and the pre-existing SAS URL is returned in the response.

For more information about granting access to snapshots, see the [Grant Access](#) API documentation.

Move resources to a new resource group or subscription

- Article
- 04/24/2023
- 16 contributors

Feedback

In this article

1. [Changed resource ID](#)
2. [Checklist before moving resources](#)
3. [Scenario for move across subscriptions](#)
4. [Use the portal](#)

Show 6 more

This article shows you how to move Azure resources to either another Azure subscription or another resource group under the same subscription. You can use the Azure portal, Azure PowerShell, Azure CLI, or the REST API to move resources.

Both the source group and the target group are locked during the move operation. Write and delete operations are blocked on the resource groups until the move completes. This lock means you can't add, update, or delete resources in the resource groups. It doesn't mean the resources are frozen. For example, if you move an Azure SQL logical server, its databases and other dependent resources to a new resource group or subscription, applications that use the databases experience no downtime. They can still read and write to the databases. The lock can last for a maximum of four hours, but most moves complete in much less time.

If your move requires setting up new dependent resources, you'll experience an interruption in those services until they've been reconfigured.

Moving a resource only moves it to a new resource group or subscription. It doesn't change the location of the resource.

Note

This article was partially created with the help of artificial intelligence. Before publishing, an author reviewed and revised the content as needed. See [Our principles for using AI-generated content in Microsoft Learn](#).

Changed resource ID

When you move a resource, you change its resource ID. The standard format for a resource ID is `/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/{resourceProviderNamespace}/{resourceType}/{resourceName}`. When you move a resource

to a new resource group or subscription, you change one or more values in that path.

If you use the resource ID anywhere, you'll need to change that value. For example, if you have a [custom dashboard](#) in the portal that references a resource ID, you'll need to update that value. Look for any scripts or templates that need to be updated for the new resource ID.

Checklist before moving resources

There are some important steps to do before moving a resource. By verifying these conditions, you can avoid errors.

1. The source and destination subscriptions must be active. If you have trouble enabling an account that has been disabled, [create an Azure support request](#). Select **Subscription Management** for the issue type.
2. The source and destination subscriptions must exist within the same [Azure Active Directory tenant](#). To check that both subscriptions have the same tenant ID, use Azure PowerShell or Azure CLI.

For Azure PowerShell, use:

Azure PowerShellCopy

Open Cloudshell

```
(Get-AzSubscription -SubscriptionName <your-source-subscription>).TenantId  
(Get-AzSubscription -SubscriptionName <your-destination-subscription>).TenantId
```

For Azure CLI, use:

Azure CLICopy

Open Cloudshell

```
az account show --subscription <your-source-subscription> --query tenantId  
az account show --subscription <your-destination-subscription> --query tenantId
```

If the tenant IDs for the source and destination subscriptions aren't the same, use the following methods to reconcile the tenant IDs:

- [Transfer ownership of an Azure subscription to another account](#)
- [How to associate or add an Azure subscription to Azure Active Directory](#)

3. If you're attempting to move resources to or from a Cloud Solution Provider (CSP) partner, see [Transfer Azure subscriptions between subscribers and CSPs](#).
4. The resources you want to move must support the move operation. For a list of which resources support move, see [Move operation support for resources](#).
5. Some services have specific limitations or requirements when moving resources. If you're moving any of the following services, check that guidance before moving.
 - If you're using Azure Stack Hub, you can't move resources between groups.
 - [App Services move guidance](#)
 - [Azure DevOps Services move guidance](#)
 - [Classic deployment model move guidance](#) - Classic Compute, Classic Storage, Classic Virtual Networks, and Cloud Services
 - [Cloud Services \(extended support\) move guidance](#)
 - [Networking move guidance](#)
 - [Recovery Services move guidance](#)
 - [Virtual Machines move guidance](#)
 - To move an Azure subscription to a new management group, see [Move subscriptions](#).
6. The destination subscription must be registered for the resource provider of the resource being moved. If not, you receive an error stating that the **subscription is not registered for a resource type**. You might see this error when moving a resource to a new subscription, but that subscription has never been used with that resource type.

For PowerShell, use the following commands to get the registration status:

Azure PowerShellCopy

Open Cloudshell

```
Set-AzContext -Subscription <destination-subscription-name-or-id>
Get-AzResourceProvider -ListAvailable | Select-Object
ProviderNamespace, RegistrationState
```

To register a resource provider, use:

Azure PowerShellCopy

Open Cloudshell

```
Register-AzResourceProvider -ProviderNamespace Microsoft.Batch
```

For Azure CLI, use the following commands to get the registration status:

Azure CLICopy

Open Cloudshell

```
az account set -s <destination-subscription-name-or-id>
az provider list --query "[].{Provider:namespace,
Status:registrationState}" --out table
```

To register a resource provider, use:

Azure CLICopy

Open Cloudshell

```
az provider register --namespace Microsoft.Batch
```

7. Before moving the resources, check the subscription quotas for the subscription you're moving the resources to. If moving the resources means the subscription will exceed its limits, you need to review whether you can request an increase in the quota. For a list of limits and how to request an increase, see [Azure subscription and service limits, quotas, and constraints](#).
8. The account moving the resources must have at least the following permissions:
 - **Microsoft.Resources/subscriptions/resourceGroups/moveResources/action** on the source resource group.
 - **Microsoft.Resources/subscriptions/resourceGroups/write** on the destination resource group.
9. If you move a resource that has an Azure role assigned directly to the resource (or a child resource), the role assignment isn't moved and becomes orphaned. After the move, you must re-create the role assignment. Eventually, the orphaned role assignment is automatically removed, but we recommend removing the role assignment before the move.

For information about how to manage role assignments, see [List Azure role assignments](#) and [Assign Azure roles](#).

10. **For a move across subscriptions, the resource and its dependent resources must be located in the same resource group and they must be moved together.** For example, a VM with managed disks would require the VM and the managed disks to be moved together, along with other dependent resources.

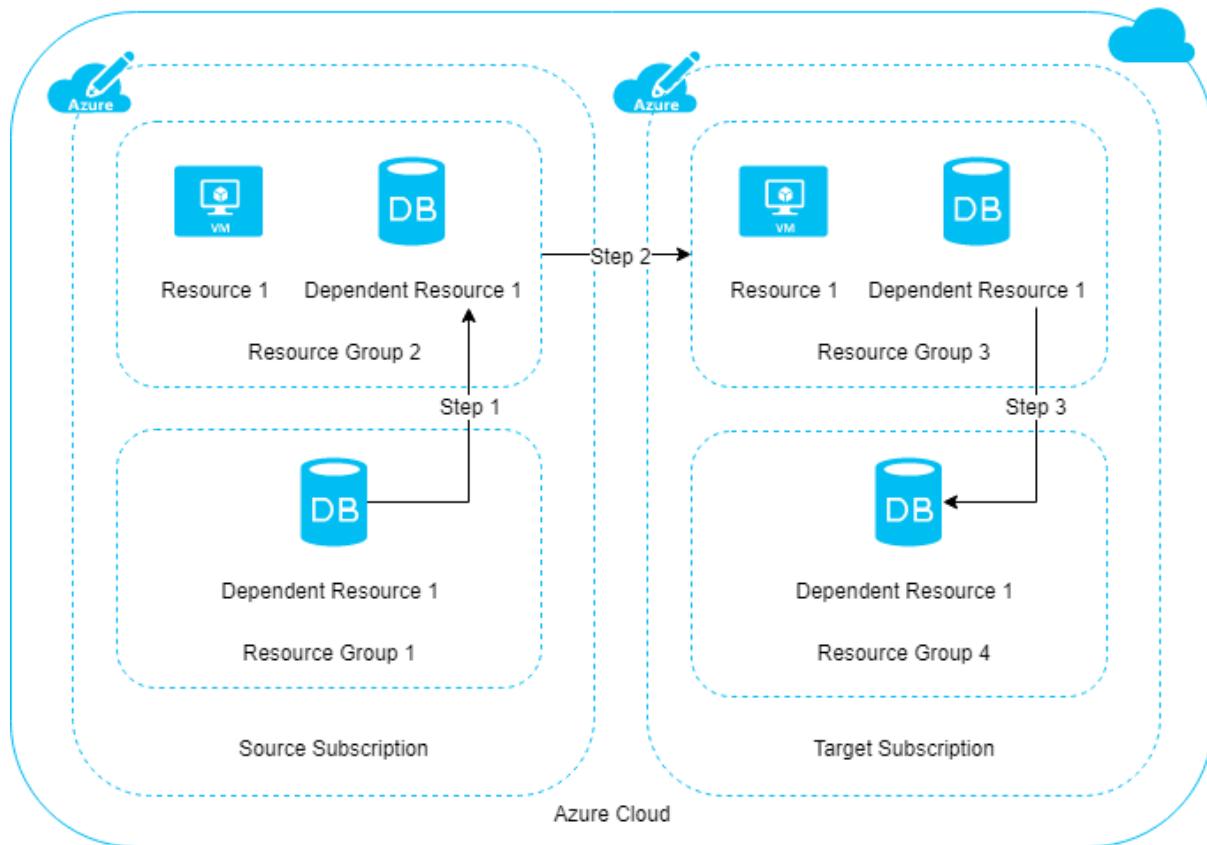
If you're moving a resource to a new subscription, check to see whether the resource has any dependent resources, and whether they're located in the same resource group. If the resources aren't in the same resource

group, check to see whether the resources can be combined into the same resource group. If so, bring all these resources into the same resource group by using a move operation across resource groups.

For more information, see [Scenario for move across subscriptions](#).

Scenario for move across subscriptions

Moving resources from one subscription to another is a three-step process:



For illustration purposes, we have only one dependent resource.

- Step 1: If dependent resources are distributed across different resource groups, first move them into one resource group.
- Step 2: Move the resource and dependent resources together from the source subscription to the target subscription.
- Step 3: Optionally, redistribute the dependent resources to different resource groups within the target subscription.

Use the portal

To move resources, select the resource group that contains those resources.

Select the resources you want to move. To move all of the resources, select the checkbox at the top of list. Or, select resources individually.

The screenshot shows the Microsoft Azure Resource Groups blade. On the left, there's a navigation pane with sections like Overview, Activity log, Access control (IAM), Tags, Events, Settings (Deployments, Security, Policies, Properties, Locks), and Cost Management (Cost analysis, Cost alerts). The main area is titled 'sourceGroup' and shows a list of resources under the 'Essentials' tab. The list includes:

- Name: exampleVM1
- Name: exampleVM1-ip
- Name: exampleVM1-nsg
- Name: examplevm1920
- Name: exampleVM1_OsDisk_1_38427735e90a4270a5888d64e37065de
- Name: sourceGroup-vnet

A red box highlights the first item in the list, 'exampleVM1'. At the bottom right of the list, there are filters for 'Type == all' and 'Location == all'.

Select the **Move** button.

The screenshot shows the 'Move' button in the top right corner of the blade. A context menu is open, listing three options:

- Move to another resource group
- Move to another subscription
- Move to another region

This button gives you three options:

- Move to a new resource group.
- Move to a new subscription.
- Move to a new region. To change regions, see [Move resources across regions \(from resource group\)](#).

Select whether you're moving the resources to a new resource group or a new subscription.

The source resource group is automatically set. Specify the destination resource group. If you're moving to a new subscription, also specify the subscription. Select **Next**.

Home > sourceGroup >

Move resources

sourceGroup

1 Source + target 2 Resources to move 3 Review

To move a resource, select a source and a destination. The source and destination resource groups will both be locked during the move. [Learn more](#)

Source

Subscription	Documentation Testing 1
Resource group	sourceGroup

Target

Subscription	Documentation Testing 1
Resource group *	<input type="text" value="destinationGroup"/> Create new

[Previous](#) [Next](#)

The portal validates that the resources can be moved. Wait for validation to complete.

Home > sourceGroup >

Move resources

sourceGroup

1 Source + target 2 Resources to move 3 Review

Checking whether these resources can be moved. This might take a few minutes.

Add resources Remove from the move list

Name	Type	Resource type	Validation status
exampleVM1	Virtual machine	microsoft.compute/virtualmachines	Pending validation
exampleVM1-ip	Public IP address	microsoft.network/publicipaddresses	Pending validation
exampleVM1-nsg	Network security group	microsoft.network/networksecuritygroups	Pending validation
examplevm1920	Network interface	microsoft.network/networkinterfaces	Pending validation
exampleVM1_OsDisk_1_38427735e90a4271	Disk	microsoft.compute/disks	Pending validation
sourceGroup-vnet	Virtual network	microsoft.network/virtualnetworks	Pending validation

When validation completes successfully, select **Next**.

Acknowledge that you need to update tools and scripts for these resources. To start moving the resources, select **Move**.

Home > sourceGroup >

Move resources

sourceGroup

1 Source + target 2 Resources to move 3 Review

Selection summary

Source subscription	Documentation Testing 1
Source resource group	sourceGroup
Target subscription	Documentation Testing 1
Target resource group	destinationGroup
Number of resources to move	6

I understand that tools and scripts associated with moved resources will not work until I update them to use new resource IDs

Previous

Move

When the move has completed, you're notified of the result.



Notifications

X

[More events in the activity log →](#)

[Dismiss all](#) ▾

Moving resources complete X

Successfully moved 6 resources from resource group 'sourceGroup' in subscription 'Documentation Testing 1' to resource group 'destinationGroup' in subscription 'Documentation Testing 1'

[Feedback](#)

[Related events](#)

12 minutes ago

Use Azure PowerShell

Validate

To test your move scenario without actually moving the resources, use the [Invoke-AzResourceAction](#) command. Use this command only when you need to predetermine the results.

Azure PowerShellCopy

```
$sourceName = "sourceRG"
$destinationName = "destinationRG"
$resourcesToMove = @("app1", "app2")

$sourceResourceGroup = Get-AzResourceGroup -Name $sourceName
$destinationResourceGroup = Get-AzResourceGroup -Name $destinationName

$resources = Get-AzResource -ResourceGroupName $sourceName | Where-Object {
    $_.Name -in $resourcesToMove }

Invoke-AzResourceAction -Action validateMoveResources ` 
    -ResourceId $sourceResourceGroup.ResourceId ` 
    -Parameters @{ resources= $resources.ResourceId;targetResourceGroup = 
    $destinationResourceGroup.ResourceId }
```

If validation passes, you see no output.

If validation fails, you see an error message describing why the resources can't be moved.

Move

To move existing resources to another resource group or subscription, use the [Move-AzResource](#) command. The following example shows how to move several resources to a new resource group.

Azure PowerShellCopy

Open Cloudshell

```
$sourceName = "sourceRG"
$destinationName = "destinationRG"
$resourcesToMove = @("app1", "app2")

$resources = Get-AzResource -ResourceGroupName $sourceName | Where-Object {
    $_.Name -in $resourcesToMove }

Move-AzResource -DestinationResourceGroupName $destinationName -ResourceId
$resources.ResourceId
```

To move to a new subscription, include a value for the `DestinationSubscriptionId` parameter.

Use Azure CLI

Validate

To test your move scenario without actually moving the resources, use the [az resource invoke-action](#) command. Use this command only when you need to predetermine the results. To run this operation, you need the:

- Resource ID of the source resource group
- Resource ID of the target resource group
- Resource ID of each resource to move

In the request body, use \" to escape double quotes.

Azure CLICopy

```
az resource invoke-action --action validateMoveResources \
    --ids "/subscriptions/{subscription-id}/resourceGroups/{source-rg}" \
    --request-body "{ \"resources\": [\"/subscriptions/{subscription-
    id}/resourceGroups/{source-rg}/providers/{resource-provider}/{resource-
    type}/{resource-name}\", \"/subscriptions/{subscription-
    id}/resourceGroups/{source-rg}/providers/{resource-provider}/{resource-
```

```
type}/{resource-name}\", \"/subscriptions/{subscription-id}/resourceGroups/{source-rg}/providers/{resource-provider}/{resource-type}/{resource-name}\"]],\"targetResourceGroup\":\"/subscriptions/{subscription-id}/resourceGroups/{destination-rg}\\" }"
```

If validation passes, you see:

Azure CLICopy

```
{} Finished ..
```

If validation fails, you see an error message describing why the resources can't be moved.

Move

To move existing resources to another resource group or subscription, use the [az resource move](#) command. In the --ids parameter, provide a space-separated list of the resource IDs to move.

The following example shows how to move several resources to a new resource group. It works when using Azure CLI in a **Bash** terminal.

Azure CLICopy

```
webapp=$(az resource show -g OldRG -n ExampleSite --resource-type "Microsoft.Web/sites" --query id --output tsv)
plan=$(az resource show -g OldRG -n ExamplePlan --resource-type "Microsoft.Web/serverfarms" --query id --output tsv)
az resource move --destination-group newgroup --ids $webapp $plan
```

The next example shows how to run the same commands in a **PowerShell** console.

Azure CLICopy

```
$webapp=$(az resource show -g OldRG -n ExampleSite --resource-type "Microsoft.Web/sites" --query id --output tsv)
$plan=$(az resource show -g OldRG -n ExamplePlan --resource-type "Microsoft.Web/serverfarms" --query id --output tsv)
az resource move --destination-group newgroup --ids $webapp $plan
```

To move to a new subscription, provide the --destination-subscription-id parameter.

Use Python

Validate

To test your move scenario without actually moving the resources, use the [ResourceManagementClient.resources.begin_validate_move_resources](#) method. Use this method only when you need to predetermine the results.

PythonCopy

```
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ResourceManagementClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

resource_client = ResourceManagementClient(credential, subscription_id)

source_name = "sourceRG"
destination_name = "destinationRG"
resources_to_move = ["app1", "app2"]

destination_resource_group = resource_client.resource_groups.get(destination_name)

resources = [
    resource for resource in
    resource_client.resources.list_by_resource_group(source_name)
    if resource.name in resources_to_move
]

resource_ids = [resource.id for resource in resources]

validate_move_resources_result =
    resource_client.resources.begin_validate_move_resources(
        source_name,
        {
            "resources": resource_ids,
            "target_resource_group": destination_resource_group.id
        }
    ).result()

print("Validate move resources result: {}".format(validate_move_resources_result))
```

If validation passes, you see no output.

If validation fails, you see an error message describing why the resources can't be moved.

Move

To move existing resources to another resource group or subscription, use the [ResourceManagementClient.resources.begin_move_resources](#) method. The following example shows how to move several resources to a new resource group.

PythonCopy

```

import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ResourceManagementClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

resource_client = ResourceManagementClient(credential, subscription_id)

source_name = "sourceRG"
destination_name = "destinationRG"
resources_to_move = ["app1", "app2"]

destination_resource_group = resource_client.resource_groups.get(destination_name)

resources = [
    resource for resource in
resource_client.resources.list_by_resource_group(source_name)
    if resource.name in resources_to_move
]

resource_ids = [resource.id for resource in resources]

resource_client.resources.begin_move_resources(
    source_name,
    {
        "resources": resource_ids,
        "target_resource_group": destination_resource_group.id
    }
)

```

Use REST API

Validate

The [validate move operation](#) lets you test your move scenario without actually moving the resources. Use this operation to check if the move will succeed. Validation is automatically called when you send a move request. Use this operation only when you need to predetermine the results. To run this operation, you need the:

- Name of the source resource group
- Resource ID of the target resource group
- Resource ID of each resource to move
- The [access token](#) for your account

Send the following request:

HTTPCopy

```

POST https://management.azure.com/subscriptions/<subscription-
id>/resourceGroups/<source-group>/validateMoveResources?api-version=2019-05-10
Authorization: Bearer <access-token>

```

```
Content-type: application/json
```

With a request body:

JSONCopy

```
{
  "resources": ["<resource-id-1>", "<resource-id-2>"],
  "targetResourceGroup": "/subscriptions/<subscription-id>/resourceGroups/<target-group>"
}
```

If the request is formatted correctly, the operation returns:

HTTPCopy

```
Response Code: 202
cache-control: no-cache
pragma: no-cache
expires: -1
location: https://management.azure.com/subscriptions/<subscription-id>/operationresults/<operation-id>?api-version=2018-02-01
retry-after: 15
...
...
```

The 202 status code indicates the validation request was accepted, but it hasn't yet determined if the move operation will succeed. The `location` value contains a URL that you use to check the status of the long-running operation.

To check the status, send the following request:

HTTPCopy

```
GET <location-url>
Authorization: Bearer <access-token>
```

While the operation is still running, you continue to receive the 202 status code. Wait the number of seconds indicated in the `retry-after` value before trying again. If the move operation validates successfully, you receive the 204 status code. If the move validation fails, you receive an error message, such as:

JSONCopy

```
{"error":{"code":"ResourceMoveProviderValidationFailed","message":"<message>"...}}
```

Move

To move existing resources to another resource group or subscription, use the [Move resources](#) operation.

HTTPCopy

```
POST https://management.azure.com/subscriptions/{source-subscription-id}/resourcegroups/{source-resource-group-name}/moveResources?api-version={api-version}
```

In the request body, you specify the target resource group and the resources to move.

JSONCopy

```
{  
  "resources": ["<resource-id-1>", "<resource-id-2>"],  
  "targetResourceGroup": "/subscriptions/<subscription-id>/resourceGroups/<target-group>"  
}
```

Frequently asked questions

Question: My resource move operation, which usually takes a few minutes, has been running for almost an hour. Is there something wrong?

Moving a resource is a complex operation that has different phases. It can involve more than just the resource provider of the resource you're trying to move. Because of the dependencies between resource providers, Azure Resource Manager allows 4 hours for the operation to complete. This time period gives resource providers a chance to recover from transient issues. If your move request is within the four-hour period, the operation keeps trying to complete and may still succeed. The source and destination resource groups are locked during this time to avoid consistency issues.

Question: Why is my resource group locked for four hours during resource move?

A move request is allowed a maximum of four hours to complete. To prevent modifications on the resources being moved, both the source and destination resource groups are locked during the resource move.

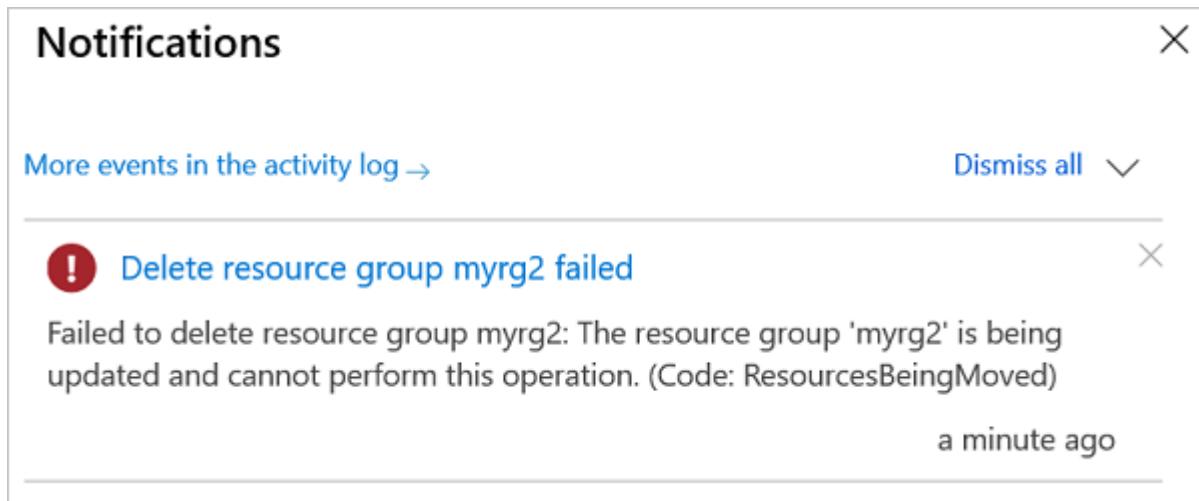
There are two phases in a move request. In the first phase, the resource is moved. In the second phase, notifications are sent to other resource providers that are dependent on the resource being moved. A resource group can be locked for the entire four hours when a resource provider fails either phase. During the allowed time, Resource Manager retries the failed step.

If a resource can't be moved within four hours, Resource Manager unlocks both resource groups. Resources that were successfully moved are in the destination resource group. Resources that failed to move are left the source resource group.

Question: What are the implications of the source and destination resource groups being locked during the resource move?

The lock prevents you from deleting either resource group, creating a new resource in either resource group, or deleting any of the resources involved in the move.

The following image shows an error message from the Azure portal when a user tries to delete a resource group that is part of an ongoing move.



Question: What does the error code "MissingMoveDependentResources" mean?

When you move a resource, its dependent resources must either exist in the destination resource group or subscription, or be included in the move request. You get the MissingMoveDependentResources error code when a dependent resource doesn't meet this requirement. The error message has details about the dependent resource that needs to be included in the move request.

For example, moving a virtual machine could require moving seven resource types with three different resource providers. Those resource providers and types are:

- Microsoft.Compute
 - virtualMachines
 - disks
- Microsoft.Network
 - networkInterfaces
 - publicIPAddresses
 - networkSecurityGroups
 - virtualNetworks
- Microsoft.Storage
 - storageAccounts

Another common example involves moving a virtual network. You may have to move several other resources associated with that virtual network. The move request could require moving public IP addresses, route tables, virtual network gateways, network security groups, and others. In general, a virtual network gateway must always be in the same resource group as its virtual network, they can't be moved separately.

Question: What does the error code "RequestDisallowedByPolicy" mean?

Resource Manager validates your move request before attempting the move. This validation includes checking policies defined on the resources involved in the move. For example, if you're attempting to move a key vault but your organization has a policy to deny the creation of a key vault in the target resource group, validation fails and the move is blocked. The returned error code is **RequestDisallowedByPolicy**.

For more information about policies, see [What is Azure Policy?](#).

Question: Why can't I move some resources in Azure?

Currently, not all resources in Azure support move. For a list of resources that support move, see [Move operation support for resources](#).

Question: How many resources can I move in a single operation?

When possible, break large moves into separate move operations. Resource Manager immediately returns an error when there are more than 800 resources in a single operation. However, moving less than 800 resources may also fail by timing out.

Question: What is the meaning of the error that a resource isn't in succeeded state?

When you get an error message that indicates a resource can't be moved because it isn't in a succeeded state, it may actually be a dependent resource that is blocking the move. Typically, the error code is **MoveCannotProceedWithResourcesNotInSucceededState**.

If the source or target resource group contains a virtual network, the states of all dependent resources for the virtual network are checked during the move. The check includes those resources directly and indirectly dependent on the virtual network. If any of those resources are in a failed state, the move is blocked. For example, if a virtual machine that uses the virtual network has failed, the move is blocked. The move is blocked even when the virtual machine isn't one of the resources being moved and isn't in one of the resource groups for the move.

When you receive this error, you have two options. Either move your resources to a resource group that doesn't have a virtual network, or [contact support](#).

Move resources across regions (from resource group)

- Article
- 03/03/2023
- 4 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Check VM requirements](#)
3. [Select resources to move](#)
4. [Resolve dependencies](#)

Show 11 more

In this article, learn how to move resources in a specific resource group to a different Azure region. In the resource group, you select the resources you want to move. Then, you move them using [Azure Resource Mover](#).

Prerequisites

- You need *Owner* access on the subscription in which resources you want to move are located.
 - The first time you add a resource for a specific source and destination mapping in an Azure subscription, Resource Mover creates a [system-assigned managed identity](#) (formerly known as Managed Service Identity (MSI)) that's trusted by the subscription.
 - To create the identity, and to assign it the required role (Contributor or User Access administrator in the source subscription), the account you use to add resources needs *Owner* permissions on the subscription. [Learn more](#) about Azure roles.
- The subscription needs enough quota to create the source resources in the target region. If it doesn't, request additional limits. [Learn more](#).
- Verify pricing and charges associated with the target region to which you're moving VMs. Use the [pricing calculator](#) to help you.
- Check that the resources you want to move are supported by Resource Mover:
 - Azure VMs and associated disks
 - NICs
 - Availability sets
 - Azure virtual networks
 - Public IP addresses
 - Network security groups (NSGs)
 - Internal and public load balancers
 - Azure SQL databases and elastic pools

Check VM requirements

1. Check that the VMs you want to move are supported.
 - [Verify](#) supported Windows VMs.
 - [Verify](#) supported Linux VMs and kernel versions.
 - Check supported [compute](#), [storage](#), and [networking](#) settings.
2. Make sure VMs have the latest trusted root certificates and an updated certificate revocation list (CRL).
 - On Azure VMs running Windows, install the latest Windows updates.
 - On VMs running Linux, follow the Linux distributor guidance to ensure the machine has the latest certificates and CRL.
3. Allow outbound connectivity from VMs:
 - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to these [URLs](#)
 - If you're using network security group (NSG) rules to control outbound connectivity, create these [service tag rules](#).

Select resources to move

Select resources you want to move. You move resources to a target region in the source region subscription. If you want to change the subscription, you can do that after the resources are moved.

Note

Don't select associated disks or the operation will fail. Associated disks are automatically included in a VM move.

1. In the Azure portal, open the relevant resource group.
2. In the resource group page, select the resources that you want to move.
3. Select **Move > Move to another region**.

The screenshot shows a list of resources in a resource group. The resources listed are:

Name	Type
raynetest-vnet	Virtual network
RayneTestVM-1	Virtual machine
raynetestvm-1-ip	Public IP address
raynetestvm-1-nsg	Network security group
raynetestvm-1286	Network interface

4. In **Source + destination**, select the target region to which you want to move the resources. Then select **Next**.

The screenshot shows the 'Move resources' wizard interface. At the top, there are three tabs: 1 Source + destination (which is selected), 2 Resources to move, and 3 Review + Add. Below the tabs, a note says: 'Select the source subscription and the source region of the resources you want to move, and the destination region you want to move your resources to. [View support matrix](#)'.

Source

Subscription * Region *

Destination

Subscription Region *

Note: You can change the subscription after moving resources to the destination region. Use the 'Move across subscriptions' feature for this. [Learn more](#)

5. In **Resources to move**, select **Next**.
6. In **Select resources**, select resource you want to move. You can only add resources supported for move. Then select **Done**.
7. In **Move resources**, select **Next**.
8. In **Review + Add**, check the source and target details.
9. Confirm that you understand that metadata about the resources being moved will be stored in a resource group created for this purpose, and that you allow Resource Mover to create a system-managed identity to access the subscription resources.
10. Select **Proceed** to begin adding the resources.

Move resources



Move across regions | PREVIEW

Source + destination

Resources to move

Review + Add

Selection summary

Source subscription	ASR PM team subscription 3
Source region	East US 2
Destination subscription	ASR PM team subscription 3
Destination region	Canada Central
Number of resources to move	5

You have now chosen the resources you want to move to the destination region.

The Azure Resource Mover will help you navigate through the following steps in the upcoming screens.

- Validate dependencies:** Validate whether resources you want to move have dependencies on other resources in the source region. After validation, add the dependent resources to the move.
- Prepare:** The preparation process depends on the resource being moved, but might typically include exporting an ARM template, or initiating data replication. This doesn't have any impact on the availability of the resources in the source region.
- Initiate move:** Bring up the resources in the destination region. The process depends on the resource you're moving, but might typically include recreating the resource in the target region, or bringing up a replica copy.

Finish up the move

- Discard (optional):** After verifying the resources in the destination region, you can optionally roll back the move.
- Commit:** If everything's running as expected in the destination region, you can commit the move. This step may result in downtime depending on the source resource type.
- Delete source:** Finally, after everything's up and running in the new region, delete the resources in the source region. This is important to avoid double billing, and to ensure a stable infrastructure.

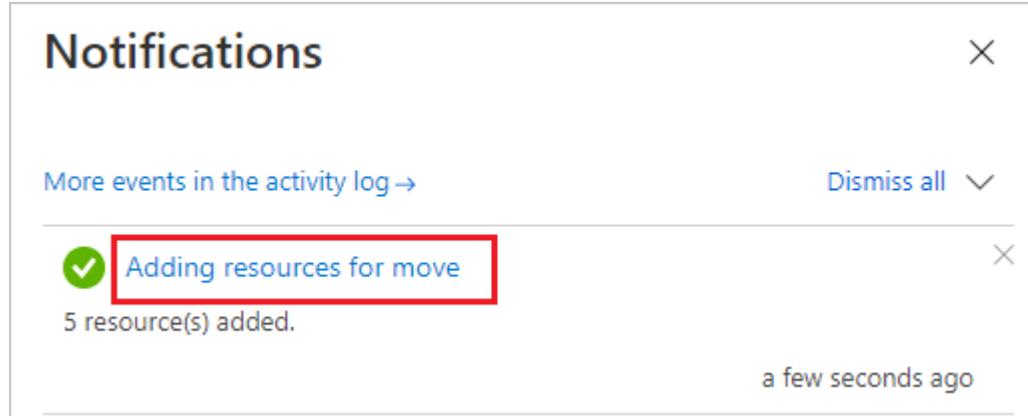
[Learn more](#)

I understand that metadata information related to this move will be stored in Resource group - 'RegionMoveRG-eastus2-canadacentral' which will be created in the region - East US 2. I am granting permissions to create a [System Assigned Managed Identity](#) on my behalf to access the resources in the subscription - ASR PM team subscription 3.

Previous

Proceed

11. The add resource operation starts. When the operation completes, the notifications show that resources were added, and deployment succeeded.
12. In the notifications, select **Adding resources for move**.



13. After selecting the notification, the resources you selected are added to a move collection in the Azure Resource Mover hub. Resource Mover helps you to check dependencies, and then start moving resources to the target region.

Resolve dependencies

Resources you're moving appear in the **Across regions** page, in a *Prepare pending* state. Start validation as follows:

1. Dependencies are *auto validated* at the beginning when you add the resources. If the initial auto validation does not resolve the issue, you will see a **Validate dependencies** ribbon, select it to validate manually.

The screenshot shows the 'Validate dependencies' ribbon highlighted with a red box. Below the ribbon, there's a table listing resources with their names, types, resource groups, destination configurations, and status. The 'Issues' column shows validation errors for each resource. A legend on the right indicates that a yellow triangle icon followed by the text 'Validate dependencies' means there are validation issues.

Name	Type	Resource group	Destination configuration	Status	Issues
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest-eastus2	(new) raynetest-vnet	准备中	Validate dependencies
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest-eastus2	(new) raynetestvm-1286	准备中	Validate dependencies
<input type="checkbox"/> raynetestnm-1-nsg	Network security group	raynetest-eastus2	(new) raynetestnm-1-nsg	准备中	Validate dependencies
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest-eastus2	(new) raynetestvm-1-ip	准备中	Validate dependencies
<input type="checkbox"/> RayneTestVM-1	Virtual machine	raynetest-eastus2	(new) RayneTestVM-1	准备中	Validate dependencies

2. If dependencies are found, select **Add dependencies**.
3. In **Add dependencies**, select the dependent resources > **Add dependencies**. Monitor progress in the notifications.

Add dependencies

Azure Resource Mover - Across regions | PREVIEW

- Some resources in your list may require other resources to be moved along with them. Review and add the dependencies
- Want to assign existing resource in the destination region? Add the resource to the list first and edit the target properties.[Learn more.](#)

Select all [Clear selection](#) | Selected items: 2 | Total items: 2

Name	Type	Resource group
<input checked="" type="checkbox"/>  rg-cleanupservice-nsg1	Network security group	test123
<input checked="" type="checkbox"/>  raynetest-eastus2	Resource group	-

4. Add additional dependencies if needed, and validate dependencies as needed. Dependency validation happens automatically in the background.
5. On the **Across regions** page, verify that resources are now in a *Prepare pending* state, with no issues.

Name	Type	Resource group	Destination configuration	Status
 raynetest-vnet	Virtual network	raynetest-eastus2	(new) raynetest-vnet	 'Prepare' pending
 raynetestvm-1286	Network interface	raynetest-eastus2	(new) raynetestvm-1286	 'Prepare' pending
 raynetestvm-1-nsg	Network security group	raynetest-eastus2	(new) raynetestvm-1-nsg	 'Prepare' pending
 raynetestvm-1-ip	Public IP address	raynetest-eastus2	(new) raynetestvm-1-ip	 'Prepare' pending
 RayneTestVM-1	Virtual machine	raynetest-eastus2	(new) RayneTestVM-1	 'Prepare' pending
 raynetest-eastus2	Resource group	-	(new) raynetest-eastus2-canadacentral	 'Prepare' pending
 rg-cleanupservice-nsg	Network security group	test123	(new) rg-cleanupservice-nsg1	 'Prepare' pending
 test123	Resource group	-	(new) test123-canadacentral	 'Prepare' pending

Move the source resource group

Before you can prepare and move resources, the source resource group must be present in the target region.

Prepare to move the source resource group

Prepare as follows:

1. In **Across regions**, select the source resource group > **Prepare**.
2. In **Prepare resources**, select **Prepare**.

The screenshot shows the 'Prepare' step in the Azure Resource Mover interface. At the top, there are buttons for 'Add resources', 'Remove', 'Refresh', 'Validate dependencies', 'Add dependency', 'Prepare' (which is highlighted with a red box), 'Initiate move', 'Discard move', and 'Commit move'. Below this, the 'source region' is set to 'East US 2', 'Destination region' to 'Canada Central', and 'Subscription' to 'subscription-id'. A search bar and filters for 'Resource type' (11 selected) and 'Move status' (12 selected) are present. The main table lists the resources being moved:

Name	Type	Resource group	Destination configuration
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest-eastus2	(new) raynetest-vnet
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest-eastus2	(new) raynetestvm-1286
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest-eastus2	(new) raynetestvm-1-nsg
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest-eastus2	(new) raynetestvm-1-ip
<input type="checkbox"/> RayneTestVM-1	Virtual machine	raynetest-eastus2	(new) RayneTestVM-1
<input checked="" type="checkbox"/> raynetest-eastus2	Resource group	-	(new) raynetest-eastus2-cana
<input type="checkbox"/> rg-cleanupservice-nsg	Network security group	test123	(new) rg-cleanupservice-nsg1
<input type="checkbox"/> test123	Resource group	-	(new) test123-canadacentral

During the Prepare process, Resource Mover generates Azure Resource Manager (ARM) templates using the resource group settings. Resources inside the resource group aren't affected.

Note

After preparing the resource group, it's in the *Initiate move pending* state. Refresh to show the latest state.

Name	Type	Resource group	Destination configuration	Status
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest-eastus2	(new) raynetest-vnet	‘Prepare’ pending
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest-eastus2	(new) raynetestvm-1286	‘Prepare’ pending
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest-eastus2	(new) raynetestvm-1-nsg	‘Prepare’ pending
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest-eastus2	(new) raynetestvm-1-ip	‘Prepare’ pending
<input type="checkbox"/> RayneTestVM-1	Virtual machine	raynetest-eastus2	(new) RayneTestVM-1	‘Prepare’ pending
<input type="checkbox"/> raynetest-eastus2	Resource group	-	(new) raynetest-eastus2-canadacentral	‘Initiate move’ pending
<input type="checkbox"/> rg-cleanupservice-nsg	Network security group	test123	(new) rg-cleanupservice-nsg1	‘Prepare’ pending
<input type="checkbox"/> test123	Resource group	-	(new) test123-canadacentral	‘Prepare’ pending

Move the source resource group

Initiate the move as follows:

1. In **Across regions**, select the resource group > **Initiate Move**
2. In **Move Resources**, select **Initiate move**. The resource group moves into an *Initiate move in progress* state.
3. After initiating the move, the target resource group is created, based on the generated ARM template. The source resource group moves into a *Commit move pending* state.

<input type="checkbox"/> RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	‘Prepare’ pending
<input type="checkbox"/> raynetest	Resource group	-	(new) raynetest-eastus2	‘Commit move’ pending
<input type="checkbox"/> raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	‘Prepare’ pending
<input type="checkbox"/> raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	‘Prepare’ pending
<input type="checkbox"/> raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	‘Prepare’ pending
<input type="checkbox"/> raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	‘Prepare’ pending

To commit and finish the move process:

1. In **Across regions**, select the resource group > **Commit move**
2. In **Move Resources**, select **Commit**.

Note

After committing the move, the source resource group is in a *Delete source pending* state.

Modify target settings

If you don't want to move a source resource, you can do either of the following:

- Create a resource in the target region with the same name and settings as the resource in the source region.
- Create a new equivalent resource in the target region. Except for the settings you specify, the target resource is created with the same settings as the source.
- Use an existing resource in the target region.

Modify a setting as follows:

1. To modify a setting, select the entry in the **Destination configuration** column for the resource.
2. In the **Destination configuration** page, specify the target settings you want to use. Changes are only made for the resource you're editing. You need to update any dependent resources separately.

The exact settings you modify depend on the resource type. [Learn more](#) about editing target settings.

Prepare resources to move

Now that the source resource group is moved, you can prepare to move the other resources.

1. In **Across regions**, select the resources you want to prepare.

Prepare resources		
Azure Resource Mover - Across regions PREVIEW		
<input checked="" type="checkbox"/> Want to assign an existing resource in the destination region? Edit the target properties before you prepare. <input checked="" type="checkbox"/> Learn more about how different resources are prepared.		
Name	Type	Resource group
RayneTestVM-1	Virtual machine	RayneTest
raynetestvm-1286	Network interface	raynetest
raynetestvm-1-nsg	Network security group	raynetest
raynetest-vnet	Virtual network	raynetest
raynetestvm-1-ip	Public IP address	raynetest

2. Select Prepare.

Note

- During the prepare process, the Azure Site Recovery Mobility agent is installed on VMs, for replication.
- VM data is replicated periodically to the target region. This doesn't affect the source VM.
- Resource Move generates ARM templates for the other source resources.
- After preparing resources, they're in an *Initiate move pending* state.

<input type="checkbox"/>	RayneTestVM-1	Virtual machine	RayneTest	(new) RayneTestVM-1	i 'Initiate move' pending
<input type="checkbox"/>	raynetest	Resource group	-	(new) raynetest-eastus2	i 'Delete source' pending
<input type="checkbox"/>	raynetestvm-1286	Network interface	raynetest	(new) raynetestvm-1286	i 'Initiate move' pending
<input type="checkbox"/>	raynetestvm-1-nsg	Network security group	raynetest	(new) raynetestvm-1-nsg	i 'Initiate move' pending
<input type="checkbox"/>	raynetest-vnet	Virtual network	raynetest	(new) raynetest-vnet	i 'Initiate move' pending
<input type="checkbox"/>	raynetestvm-1-ip	Public IP address	raynetest	(new) raynetestvm-1-ip	i 'Initiate move' pending

Initiate the move

With resources prepared, you can now initiate the move.

- In **Across regions**, select resources with state *Initiate move pending*. Then select **Initiate move**.
- In **Move resources**, select **Initiate move**.

Name	Type
RayneTestVM-1	Virtual machine
raynetestvm-1286	Network interface
raynetestvm-1-nsg	Network security group
raynetest-vnet	Virtual network
raynetestvm-1-ip	Public IP address

Initiate move **Cancel**

3. Track move progress in the notifications bar.

Note

- For VMs, replica VMs are created in the target region. The source VM is shut down, and some downtime occurs (usually minutes).
- Resource Mover recreates other resources using the ARM templates that were prepared. There's usually no downtime.
- For load balancers, NAT rules aren't copied. Create them in the target region after you commit the move.
- For public IP addresses, the DNS name label isn't copied. Recreate the label after you commit the move.
- After preparing resources, they're in an *Commit move pending* state.

Discard or commit?

After the initial move, you can decide whether you want to commit the move, or to discard it.

- **Discard:** You might discard a move if you're testing, and you don't want to actually move the source resource. Discarding the move returns the resource to a state of *Initiate move pending*.
- **Commit:** Commit completes the move to the target region. After committing, a source resource will be in a state of *Delete source pending*, and you can decide if you want to delete it.

Discard the move

You can discard the move as follows:

1. In **Across regions**, select resources with state *Commit move pending*, and select **Discard move**.
2. In **Discard move**, select **Discard**.
3. Track move progress in the notifications bar.
4. When the notifications show that the move was successful, select **Refresh**.

Note

For VMs, After discarding resources, they're in an *Initiate move pending* state.

Commit the move

If you want to complete the move process, commit the move.

1. In **Across regions**, select resources with state *Commit move pending*, and select **Commit move**.
2. In **Commit resources**, select **Commit**.

Commit resources

PREVIEW

Name	Type
 RayneTestVM-1	Virtual machine
 raynetestvm-1286	Network interface
 raynetestvm-1-nsg	Network security group
 raynetest-vnet	Virtual network
 raynetestvm-1-ip	Public IP address

Commit

Cancel

3. Track the commit progress in the notifications bar.

Note

- After committing the move, VMs stop replicating. The source VM isn't impacted by the commit.
- Commit doesn't impact source networking resources.
- After committing the move, resources are in a *Delete source pending* state.

Configure settings after the move

1. Since DNS name labels aren't copied over for public IP addresses, after the move is done, navigate to the target resources and update the label.
2. For internal load balancers, since NAT rules aren't copied over, navigate to the resources created in the target region, and update the NAT rules.
3. The Mobility service isn't uninstalled automatically from VMs. Uninstall it manually, or leave it if you plan to move the server again.

Delete source resources after commit

After the move, you can optionally delete resources in the source region.

1. In **Across Regions**, select the name of each source resource that you want to delete.
2. In the properties page for each resource, select **Delete**.

Delete additional resources created for move

After the move, you can manually delete the move collection, and Site Recovery resources that were created.

- The move collection is hidden by default. To see it you need to turn on hidden resources.
- The cache storage has a lock that must be deleted, before it can be deleted.

Delete as follows:

1. Locate the resources in resource group `RegionMoveRG-<sourcerregion>-<target-region>`, in the source region.
2. Check that all the VM and other source resources in the move collection have been moved/deleted. This ensures that there are no pending resources using them.
3. Delete the resources:
 - The move collection name is `movecollection-<sourcerregion>-<target-region>`.

- The cache storage account name is resmovecache<guid>
- The vault name is ResourceMove-<sourcerregion>-<target-region>-GUID.

Move App Service resources to a new resource group or subscription

- Article
- 04/09/2023
- 4 contributors

Feedback

In this article

1. [Move across subscriptions](#)
2. [Find original resource group](#)
3. [Move hidden resource types in portal](#)
4. [Move with free managed certificates](#)

Show 2 more

This article describes the steps to move App Service resources between resource groups or Azure subscriptions. There are specific requirements for moving App Service resources to a new subscription.

If you want to move App Services to a new region, see [Move an App Service resource to another region](#).

Move across subscriptions

When you move a Web App across subscriptions, the following guidance applies:

- Moving a resource to a new resource group or subscription is a metadata change that shouldn't affect anything about how the resource functions. For example, the inbound IP address for an app service doesn't change when moving the app service.
- The destination resource group must not have any existing App Service resources. App Service resources include:
 - Web Apps
 - App Service plans
 - Uploaded or imported TLS/SSL certificates
 - App Service Environments
- All App Service resources in the resource group must be moved together.

- App Service Environments can't be moved to a new resource group or subscription. However, you can move a web app and app service plan to a new subscription without moving the App Service Environment.
- You can move a certificate bound to a web without deleting the TLS bindings, as long as the certificate is moved with all other resources in the resource group. However, you can't move a free App Service managed certificate. For that scenario, see [Move with free managed certificates](#).
- App Service apps with private endpoints cannot be moved. Delete the private endpoint(s) and recreate it after the move.
- App Service resources can only be moved from the resource group in which they were originally created. If an App Service resource is no longer in its original resource group, move it back to its original resource group. Then, move the resource across subscriptions. For help with finding the original resource group, see the next section.

Find original resource group

If you don't remember the original resource group, you can find it through diagnostics. For your web app, select **Diagnose and solve problems**. Then, select **Configuration and Management**.

movewebapp - Diagnose and solve problems

Search (Ctrl+ /)

Home

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Security

Deployment

Quickstart

Deployment slots

Deployment Center

Settings

Configuration

Application settings (Classic)

Search App Service Diagnostics

App Service Diagnostics

Use App Service Diagnostics to investigate how your app is performing, diagnose issues, and discover how to improve your application. Select the problem category that best matches the information or tool that you're interested in:

Availability and Performance

Is your app experiencing downtime or slowness? Click here to run a health checkup to discover issues that may be affect your app's high availability, by either platform or app issues.

Keywords

Health Check, Downtime, 5xx Errors, 4xx Errors, CPU, Memory

Configuration and Management

Are you having issues with something that you configured specifically for your app? Find out if you misconfigured App Service features, such as backups, deployment slots, and scaling.

Keywords

Scaling, Swaps, Failed Backups, IPs, Migration

Select **Migration Options**.

Home Configuration and Management X

Hello! Welcome to App Service Diagnostics! My name is Genie and I'm here to help you diagnose and solve problems.

Here are some issues related to Configuration and Management that I can help with. Please select the tile that best describes your issue.

Check Backup Failures Check Swap Operations Deprecating APIs IP Address Configuration Migration Operations

Which client IPs got rejected due to IP restriction?

Select the option for recommended steps to move the web app.

Home Configuration and Management X

Hello! Welcome to App Service Diagnostics! My name is Genie and I'm here to help you diagnose and solve problems.

Here are some issues related to Configuration and Management that I can help with. Please select the tile that best describes your issue.

Okay give me a moment while I analyze your app for any issues related to this tile. Once the detectors load, feel free to click to investigate each topic further.

i Recommended steps to move Microsoft.Web resources across the subscription from 'demogroup2' resource group >

i Recommendation to change App Service Plan of the site 'movewebapp' >

You see the recommended actions to take before moving the resources. The information includes the original resource group for the web app.

Recommended steps Recommended steps to move Microsoft.Web resources across the subscription from 'demogroup2' resource group

Recommended steps

- Please select all the Microsoft.Web resources from **demogroup2** resource group for cross subscription migration. Please see the list below:
 - movewebapp** (sites)

NOTE: This site was originally created in "**demogroup1**" resource group. Please move this site back to the "**demogroup1**" resource group to enable it for new migration operation.
 - app-plan** (App Service Plans)

NOTE: This App Service Plan was originally created in "**demogroup1**" resource group. Please move this site back to the "**demogroup1**" resource group to enable it for new migration operation.
- Please ensure destination resource group doesn't have any Microsoft.Web resources before the move operation.

Move hidden resource types in portal

When using the portal to move your App Service resources, you may see an error indicating that you haven't moved all of the resources. If you see this error, check if there are resource types that the portal didn't display. Select **Show hidden types**. Then, select all of the resources to move.

Microsoft Azure

Search resources, services, and docs (G+/)

Home >

webgroup Resource group

Search (Ctrl+ /) Create Edit columns Delete resource group Refresh

Overview

Subscription (change) : Documentation Testing 1
Subscription ID :
Tags (change) : Click here to add tags

Filter for any field... Type == all X Location == all X

Showing 1 to 3 of 3 records.

Show hidden types ⓘ

Name ↑

<input checked="" type="checkbox"/>	ASP-webgroup-92f4
<input checked="" type="checkbox"/>	tfwebapp0826
<input checked="" type="checkbox"/>	tfwebapp0826

Move with free managed certificates

You can't move a free App Service managed certificate. Instead, delete the managed certificate and recreate it after moving the web app. To get instructions for deleting the certificate, use the **Migration Operations** tool.

If your free App Service managed certificate gets created in an unexpected resource group, try moving the app service plan back to its original resource group. Then, recreate the free managed certificate. This issue will be fixed.

Move support

To determine which App Service resources can be moved, see move support status for:

- [Microsoft.AppService](#)
- [Microsoft.CertificateRegistration](#)
- [Microsoft.DomainRegistration](#)
- [Microsoft.Web](#)

Move your Azure Automation account to another subscription

- Article
- 05/29/2023
- 9 contributors

Feedback

In this article

1. [Remove features](#)
2. [Unlink your workspace](#)
3. [Move your Automation account](#)
4. [Enable features](#)

Show 2 more

Azure Automation allows you to move some resources to a new resource group or subscription. You can move resources through the Azure portal, PowerShell, the Azure CLI, or the REST API. To learn more about the process, see [Move resources to a new resource group or subscription](#).

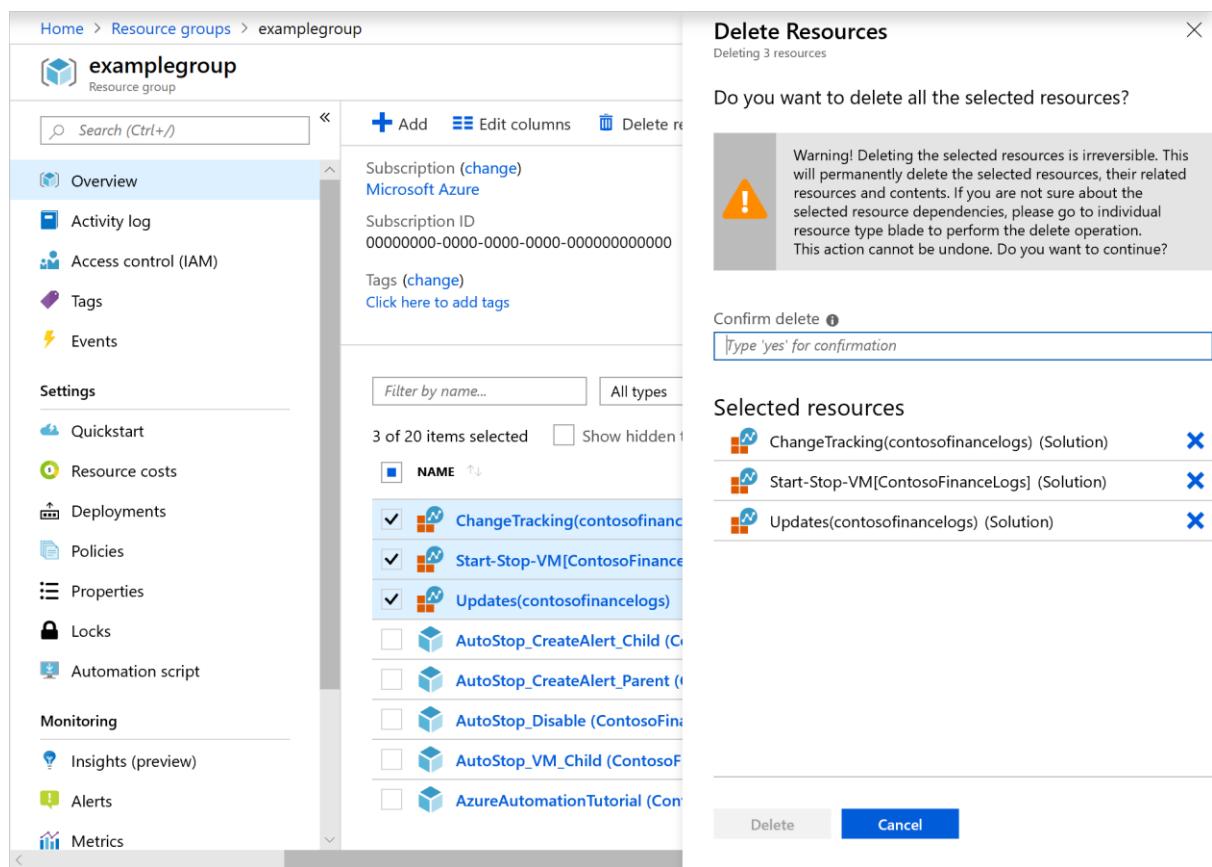
The Automation account is one of the resources that you can move. In this article, you'll learn to move Automation accounts to another resource or subscription. The high-level steps for moving your Automation account are:

1. Disable your features.
2. Unlink your workspace.
3. Move the Automation account.
4. Re-enable your features.

Remove features

To unlink your workspace from your Automation account, you must remove the feature resources in your workspace:

- Change Tracking and Inventory
 - Update Management
 - Start/Stop VMs during off-hours
1. In the Azure portal, locate your resource group.
 2. Find each feature, and select **Delete** on the **Delete Resources** page.



If you prefer, you can delete the resources by using the [Remove-AzResource](#) cmdlet:

Azure PowerShellCopy

Open Cloudshell

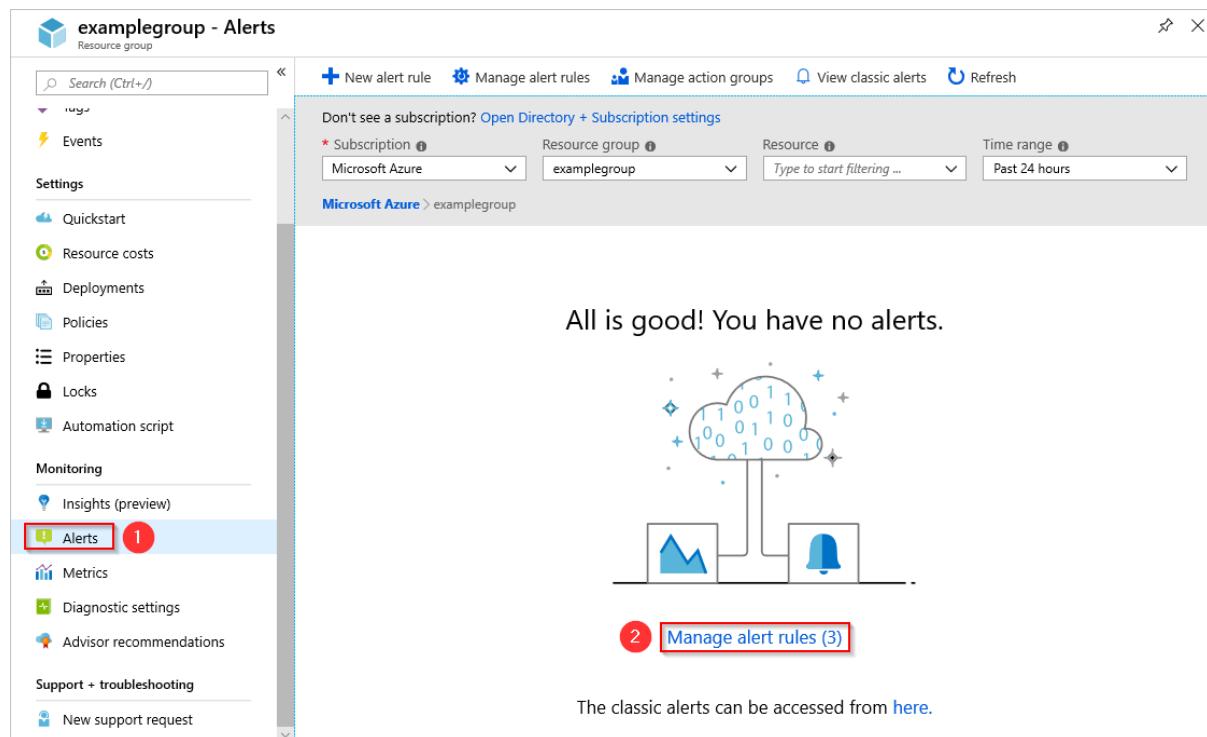
```
$workspaceName = <myWorkspaceName>
$resourceGroupName = <myResourceGroup>
Remove-AzResource -ResourceType 'Microsoft.OperationsManagement/solutions' -
    ResourceName "ChangeTracking($workspaceName)" -ResourceGroupName $resourceGroupName
Remove-AzResource -ResourceType 'Microsoft.OperationsManagement/solutions' -
    ResourceName "Updates($workspaceName)" -ResourceGroupName $resourceGroupName
```

```
Remove-AzResource -ResourceType 'Microsoft.OperationsManagement/solutions' -  
ResourceName "Start-Stop-VM($workspaceName)" -ResourceGroupName $resourceGroupName
```

Remove alert rules for Start/Stop VMs during off-hours

For Start/Stop VMs during off-hours, you also need to remove the alert rules created by the feature.

1. In the Azure portal, go to your resource group and select **Monitoring > Alerts > Manage alert rules**.



2. On the Rules page, you should see a list of the alerts configured in that resource group. The feature creates these rules:
 - AutoStop_VM_Child
 - ScheduledStartStop_Parent
 - SequencedStartStop_Parent
3. Select the rules one at a time, and select **Delete** to remove them.

The screenshot shows the Azure portal's 'Rules' management interface. At the top, there are buttons for 'New alert rule', 'Edit columns', 'Manage action groups', 'View classic alerts', 'Refresh', 'Enable', 'Disable', and 'Delete'. Below this is a 'Delete' confirmation dialog asking 'Are you sure you want to delete?'. Two buttons, 'Yes' and 'No', are present. A note below the dialog says 'Click on "View classic alerts" to view rules configured in Alerts (classic)'. The main table displays three alert rules:

NAME	CONDITION	STATUS	TARGET RESOURCE	TARGET RESOURCE ...	SIGNAL TYPE
<input checked="" type="checkbox"/> AutoStop_VM_Child	AzureDiagnostics where (RunbookName_s == "AutoStop_V...")	Disabled	contosofinancelogs	Log Analytics worksp...	Log Search
<input checked="" type="checkbox"/> ScheduledStartStop_Parent	AzureDiagnostics where (RunbookName_s == "ScheduledS...")	Disabled	contosofinancelogs	Log Analytics worksp...	Log Search
<input checked="" type="checkbox"/> SequencedStartStop_Parent	AzureDiagnostics where (RunbookName_s == "Sequenced...")	Disabled	contosofinancelogs	Log Analytics worksp...	Log Search

Note

If you don't see any alert rules on the Rules page, change the **Status** field to **Disabled** to show disabled alerts.

4. When you remove the alert rules, you must remove the action group created for Start/Stop VMs during off-hours notifications. In the Azure portal, select **Monitor > Alerts > Manage action groups**.
5. Select **StartStop_VM_Notification**.
6. On the action group page, select **Delete**.

The screenshot shows the Azure portal's configuration page for an action group named 'StartStop_VM_Notification'. At the top, there are buttons for Save, Discard, Refresh, and Delete, with the Delete button highlighted by a red box. Below these, the 'Short name' is set to 'StStAlert'. The 'Action group name' is also 'StartStop_VM_Notification'. Under 'Resource group', it says 'examplegroup'. Under 'Subscription', it says 'Microsoft Azure'. In the 'Actions' section, there is a table with one row. The row contains 'EmailInt3m3rhm56gq4' under 'ACTION NAME', 'Email/SMS/Push/V...' under 'ACTION TYPE', 'Subscribed' under 'STATUS', and a 'Edit details' link under 'DETAILS'. A delete icon is also present in the 'ACTIONS' column. Below the table, there are links for 'Privacy Statement' and 'Pricing'.

If you prefer, you can delete your action group by using the [Remove-AzActionGroup](#) cmdlet:

Azure PowerShellCopy

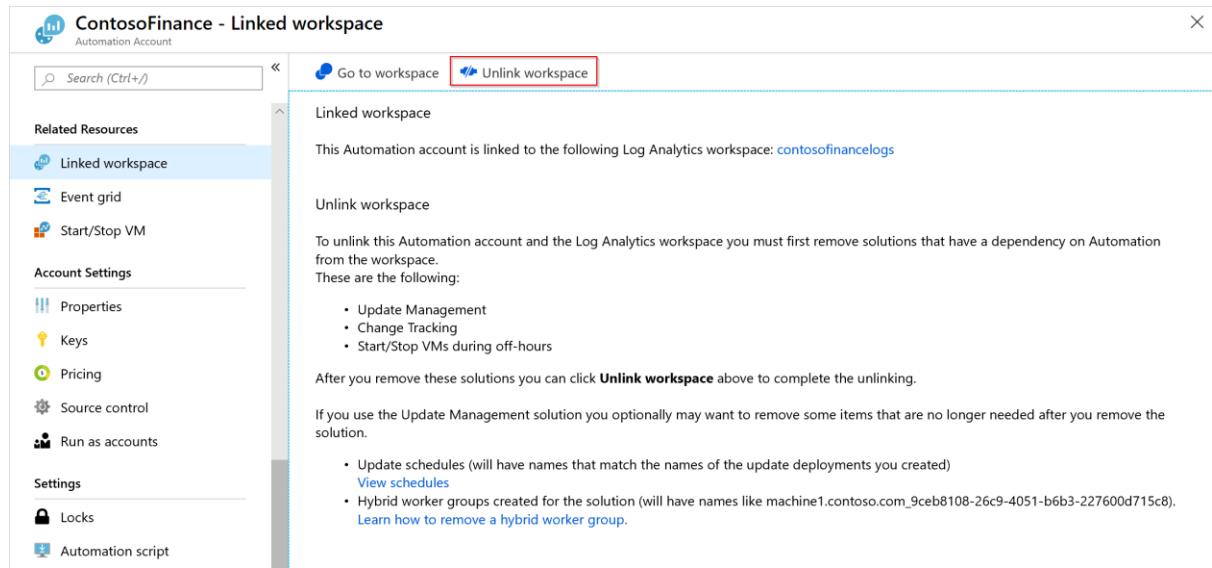
Open Cloudshell

```
Remove-AzActionGroup -ResourceGroupName <myResourceGroup> -Name StartStop_VM_Notification
```

Unlink your workspace

Now you can unlink your workspace:

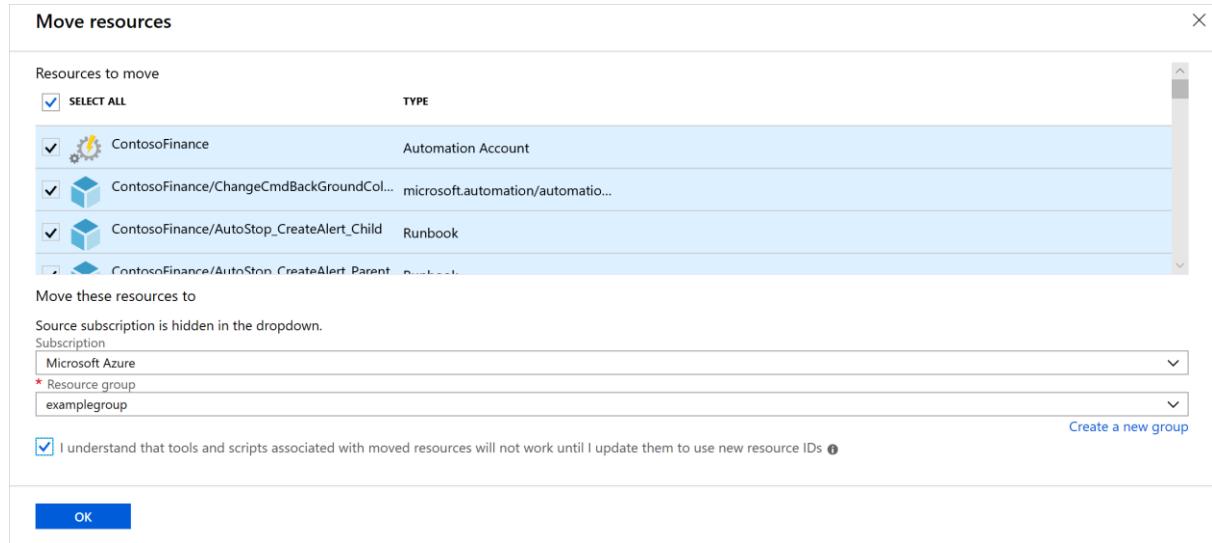
1. In the Azure portal, select **Automation account** > **Related Resources** > **Linked workspace**.
2. Select **Unlink workspace** to unlink the workspace from your Automation account.



Move your Automation account

You can now move your Automation account and its runbooks.

1. In the Azure portal, browse to the resource group of your Automation account. Select **Move > Move to another subscription**.



2. Select the resources in your resource group that you want to move. Ensure that you include your Automation account, runbooks, and Log Analytics workspace resources.

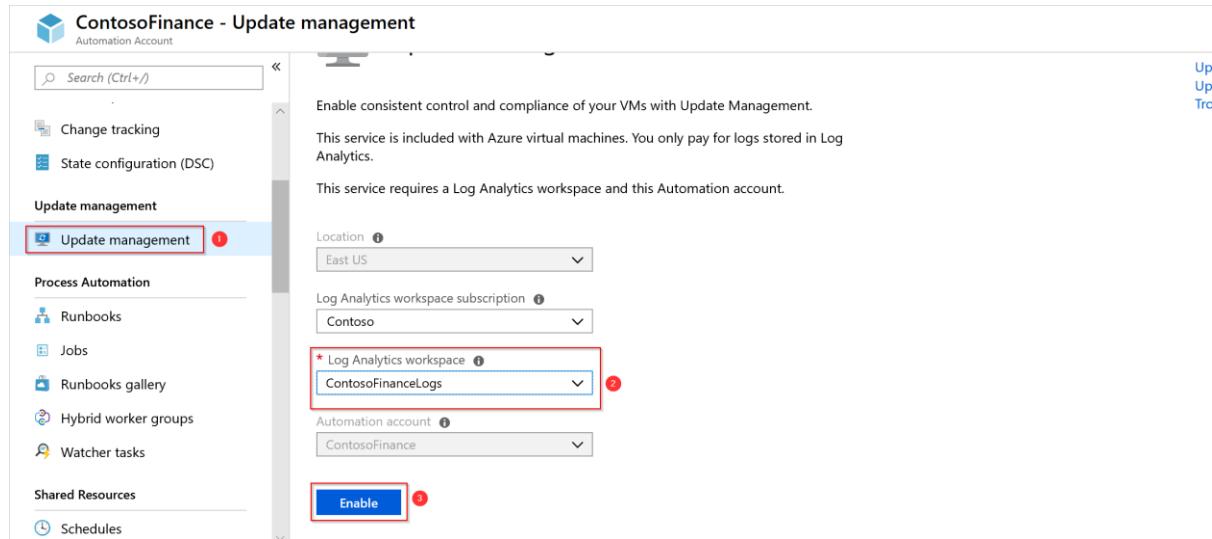
Note

The movement of System assigned managed identity, and User-assigned managed identity takes place automatically with the Automation account.

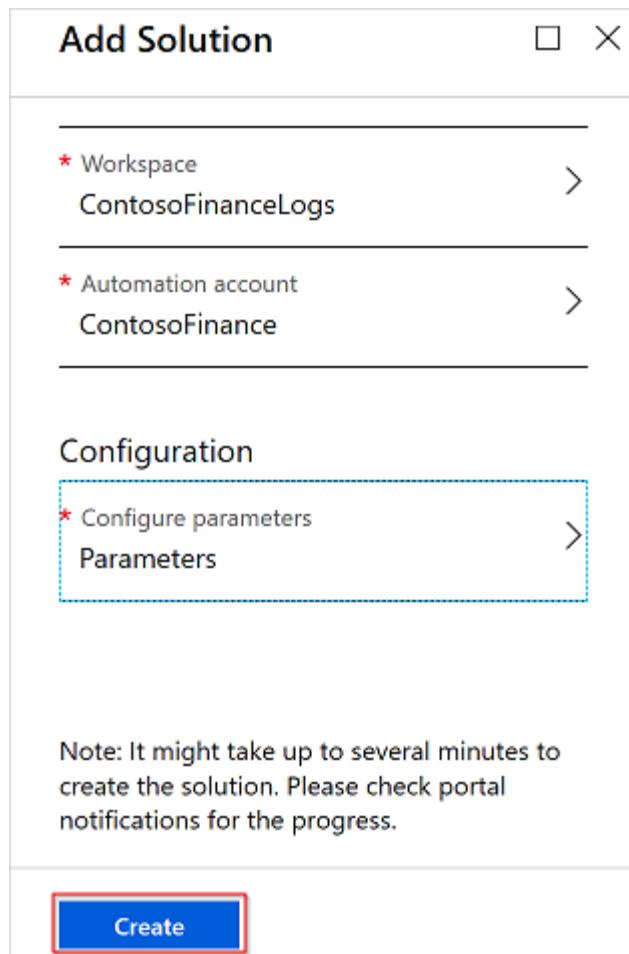
Enable features

You must re-enable the features that you removed before the move:

1. To turn on Change Tracking and Inventory, select **Change Tracking and Inventory** in your Automation account. Choose the Log Analytics workspace that you moved over, and select **Enable**.
2. Repeat step 1 for Update Management.



3. Machines that are enabled with your features are visible when you've connected the existing Log Analytics workspace.
4. On the Add Solution page, choose your Log Analytics workspace and Automation account.



Verify the move

When the move is complete, verify that the capabilities listed below are enabled.

Capability	Tests	Troubleshooting
Runbooks	A runbook can successfully run and connect to Azure resources.	Troubleshoot runbooks
Source control	You can run a manual sync on your source control repository.	Source control integration
Change tracking and inventory	Verify that you see current inventory data from your machines.	Troubleshoot change tracking
Update management	Verify that you see your machines and that they're healthy. Run a test software update deployment.	Troubleshoot update management
Shared resources	Verify that you see all your shared resources, such as credentials and variables .	

Move guidance for classic deployment model resources

- Article
- 04/09/2023
- 2 contributors

Feedback

In this article

1. [Move in the same subscription](#)
2. [Move across subscriptions](#)
3. [Possible error messages in the source subscription validation stage](#)
4. [Next steps](#)

The steps to move resources deployed through the classic model differ based on whether you're moving the resources within a subscription or to a new subscription.

Move in the same subscription

When moving resources from one resource group to another resource group within the same subscription, the following restrictions apply:

- Virtual networks (classic) can't be moved.
- Virtual machines (classic) must be moved with the cloud service.
- Cloud service can only be moved when the move includes all its virtual machines.
- Only one cloud service can be moved at a time.
- Only one storage account (classic) can be moved at a time.
- Storage account (classic) can't be moved in the same operation with a virtual machine or a cloud service.

To move classic resources to a new resource group within the same subscription, use the [standard move operations](#) through the portal, Azure PowerShell, Azure CLI, or REST API. You use the same operations as you use for moving Resource Manager resources.

Move across subscriptions

When moving classic cloud services to a new subscription, the following restrictions apply:

- The source and target subscriptions need to be under the same Azure AD tenant.

- Cloud Service Provider (CSP) subscriptions do not support migrating classic cloud services.
- All classic resources in the subscription must be moved in the same operation.
- The target subscription must not have any other classic resources.
- The move can only be requested through a separate REST API for classic moves. The standard Resource Manager move commands don't work when moving classic resources to a new subscription.

To move classic resources to a new subscription, use the REST operations that are specific to classic resources. To use REST, do the following steps:

1. Check if the source subscription can participate in a cross-subscription move. Use the following operation:

HTTPCopy

```
POST
https://management.azure.com/subscriptions/{sourceSubscriptionId}/providers/Microsoft.ClassicCompute/validateSubscriptionMoveAvailability?api-version=2016-04-01
```

In the request body, include:

JSONCopy

```
{
  "role": "source"
}
```

The response for the validation operation is in the following format:

JSONCopy

```
{
  "status": "{status}",
  "reasons": [
    "reason1",
    "reason2"
  ]
}
```

2. Check if the destination subscription can participate in a cross-subscription move. Use the following operation:

HTTPCopy

```
POST
https://management.azure.com/subscriptions/{destinationSubscriptionId}/providers/Microsoft.ClassicCompute/validateSubscriptionMoveAvailability?api-version=2016-04-01
```

In the request body, include:

JSONCopy

```
{  
  "role": "target"  
}
```

The response is in the same format as the source subscription validation.

3. If both subscriptions pass validation, move all classic resources from one subscription to another subscription with the following operation:

HTTPCopy

```
POST https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.ClassicCompute/moveSubscriptionResources?api-version=2016-04-01
```

In the request body, include:

JSONCopy

```
{  
  "target": "/subscriptions/{target-subscription-id}"  
}
```

The operation may run for several minutes.

Possible error messages in the source subscription validation stage

"Subscription migration for SubscriptionId {subscription ID} cannot continue as IaaS classic to ARM migration is in progress for the following deployment resource: xx in HostedService {classic-cloud-service-name}"

This message means there is a classic cloud service that is ongoing migrating to the cloud service (extended support). Users should abort this ARM migration operation and then retry validation.

"Source subscription {subscription ID} is empty"

The source subscription cannot be empty, disabled, deleted or currently undergoing migration. During the migration period, write operations are not allowed on resources within the subscription.

"Source subscription contains application(s) which doesn't support migration: {application name}"

"Source subscription contains following cloud service(s) which doesn't support migration: {cloud service name}"

The resources mentioned in the error message cannot be migrated, so users should delete these resources before triggering the migration.

More information

The domain name and the public IP are still the same as before migration. Under normal circumstances, there should be no downtime for the cloud service during the migration.

Move guidance for Cloud Services (extended support) deployment model resources

- Article
- 06/21/2023
- 3 contributors

Feedback

In this article

1. [Move in the same subscription](#)
2. [Move across subscriptions](#)

Important

The move feature is under development for Cloud Services (extended support) and not available for Production use yet. The guidance will be updated once the feature is deployed for all customers and regions.

The steps to move resources deployed through the Cloud Services (extended support) model differ based on whether you're moving the resources within a subscription or to a new subscription.

Move in the same subscription

When moving Cloud Services (extended support) resources from one resource group to another resource group within the same subscription, the following restrictions apply:

- Cloud Service must not be in manual mode
- Cloud Service must not be VIP Swappable
- Cloud Service must not have any pending operations
- Cloud Service must not be in migration
- Cloud Service must not be in failed state
- Ensure the Cloud Service has an unexpired SAS blob URI pointing to the cloud service package

Note

Cloud Services and associated networking resources (for example, PublicIPs and network security groups) can be move independently. Load balancers must always exist in the same resource group

To move classic resources to a new resource group within the same subscription, use the [standard move operations](#) through the portal, Azure PowerShell, Azure CLI, or REST API. You use the same operations as you use for moving Resource Manager resources.

Move across subscriptions

When moving Cloud Services (extended support) deployments to a new subscription, the following restrictions apply:

- When performing a cross subscription move, all associated cloud service resources such key vault and network resources must move together.
- If faced with a Move Resource operation error saying that the cloud service can't be moved because of a prior failed operation, create a ticket to resolve the issue.
- Cloud Service must not have any cross-subscription references.

Move networking resources to new resource group or subscription

- Article
- 05/08/2023
- 8 contributors

Feedback

In this article

1. [Dependent resources](#)
2. [Peered virtual network](#)
3. [VPN Gateways](#)
4. [Subnet links](#)

Show 2 more

This article describes how to move virtual networks and other networking resources to a new resource group or Azure subscription.

During the move, your networking resources operate without interruption.

If you want to move networking resources to a new region, see [Tutorial: Move Azure VMs across regions](#).

Dependent resources

When moving a resource, you must also move its dependent networking resources. However, any resource that is associated with a **Standard SKU** public IP address can't be moved across subscriptions. For example, you can't move a VPN Gateway that is associated with a **Standard SKU** public IP address to a new subscription.

To move a virtual machine with a network interface card to a new subscription, you must move all dependent resources. Move the virtual network for the network interface card, all other network interface cards for the virtual network, and the VPN gateways. If a virtual machine is associated with a **Standard SKU** public IP address, [disassociate the public IP address](#) before moving across subscriptions.

If you move the virtual network for an AKS cluster, the AKS cluster stops working. The local network gateways can be in a different resource group.

For more information, see [Scenario for move across subscriptions](#).

Peered virtual network

To move a peered virtual network, you must first disable the virtual network peering. Once disabled, you can move the virtual network. After the move, reenable the virtual network peering.

VPN Gateways

You can't move VPN Gateways across resource groups or subscriptions if they are of Basic SKU. Basic SKU is only meant for test environment usage and doesn't support resource move operation. A virtual network gateway must always be in the same resource group as its virtual network, they can't be moved separately.

Subnet links

You can't move a virtual network to a different subscription if the virtual network contains a subnet with resource navigation links. For example, if an Azure Cache for Redis resource is deployed into a subnet, that subnet has a resource navigation link.

Private endpoints

The following [private-link resources](#) support move:

- Microsoft.aadiam/privateLinkForAzureAD
- Microsoft.DocumentDB/databaseAccounts
- Microsoft.Kusto/clusters
- Microsoft.Search/searchServices
- Microsoft.SignalRService/SignalR
- Microsoft.SignalRService/webPubSub
- Microsoft.Sql/servers
- Microsoft.StorageSync/storageSyncServices
- Microsoft.Synapse/workspaces
- Microsoft.Synapse/privateLinkHubs

All other private-link resources don't support move.

Move a Recovery Services vault across Azure subscriptions and resource groups

- Article
- 02/02/2023
- 15 contributors

Feedback

In this article

1. [Supported regions](#)

2. [Prerequisites for moving Recovery Services vault](#)
3. [Use Azure portal to move Recovery Services vault to different resource group](#)
4. [Use Azure portal to move Recovery Services vault to a different subscription](#)

Show 6 more

This article explains how to move a Recovery Services vault configured for Azure Backup across Azure subscriptions, or to another resource group in the same subscription. You can use the Azure portal or PowerShell to move a Recovery Services vault.

Supported regions

All public regions and sovereign regions are supported, except France South, France Central, Germany Northeast and Germany Central.

Prerequisites for moving Recovery Services vault

- During vault move across resource groups, both the source and target resource groups are locked preventing the write and delete operations. For more information, see this [article](#).
- Only admin subscription has the permissions to move a vault.
- For moving vaults across subscriptions, the target subscription must reside in the same tenant as the source subscription and its state must be enabled. To move a vault to a different Azure AD, see [Transfer subscription to a different directory](#) and [Recovery Service vault FAQs](#).
- You must have permission to perform write operations on the target resource group.
- Moving the vault only changes the resource group. The Recovery Services vault will reside on the same location and it can't be changed.
- You can move only one Recovery Services vault, per region, at a time.
- If a VM doesn't move with the Recovery Services vault across subscriptions, or to a new resource group, the current VM recovery points will remain intact in the vault until they expire.
- Whether the VM is moved with the vault or not, you can always restore the VM from the retained backup history in the vault.
- The Azure Disk Encryption requires that the key vault and VMs reside in the same Azure region and subscription.
- To move a virtual machine with managed disks, see this [article](#).
- The options for moving resources deployed through the Classic model differ depending on whether you're moving the resources within a subscription, or to a new subscription. For more information, see this [article](#).
- Backup policies defined for the vault are retained after the vault moves across subscriptions or to a new resource group.

- You can only move a vault that contains any of the following types of backup items. Any backup items of types not listed below will need to be stopped and the data permanently deleted before moving the vault.
 - Azure Virtual Machines
 - Microsoft Azure Recovery Services (MARS) Agent
 - Microsoft Azure Backup Server (MABS)
 - Data Protection Manager (DPM)
- If you move a vault containing VM backup data, across subscriptions, you must move your VMs to the same subscription, and use the same target VM resource group name (as it was in old subscription) to continue backups.

Note

Moving Recovery Services vaults for Azure Backup across Azure regions isn't supported.

If you've configured any VMs (Azure IaaS, Hyper-V, VMware) or physical machines for disaster recovery using **Azure Site Recovery**, the move operation will be blocked. If you want to move vaults for Azure Site Recovery, review [this article](#) to learn about moving vaults manually.

Use Azure portal to move Recovery Services vault to different resource group

To move a Recovery Services vault and its associated resources to different resource group:

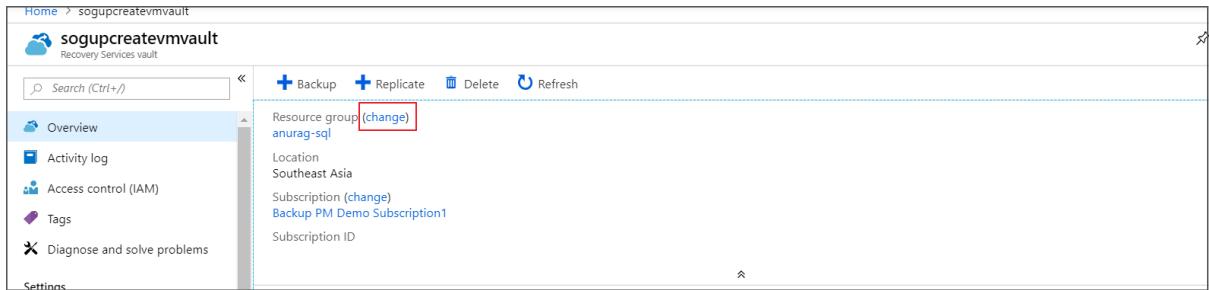
1. Sign in to the [Azure portal](#).
2. Open the list of **Recovery Services vaults** and select the vault you want to move. When the vault dashboard opens, it appears as shown in the following image.

The screenshot shows the Azure Recovery Services vault overview page for the vault 'sogupcreatevmvault'. The left sidebar contains a navigation menu with links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Properties, Locks, Automation script, Backup, Site Recovery, Backup items, Replicated items, and Manage. The main content area features a 'Backup' section with links for Getting started, Backup dashboard, Backup items, and Backup policies. It also features a 'Site Recovery' section with links for Getting started, Site Recovery dashboard, Replicated items, and Manage Recovery Plans. A 'What's new' section at the top right lists several recent updates.

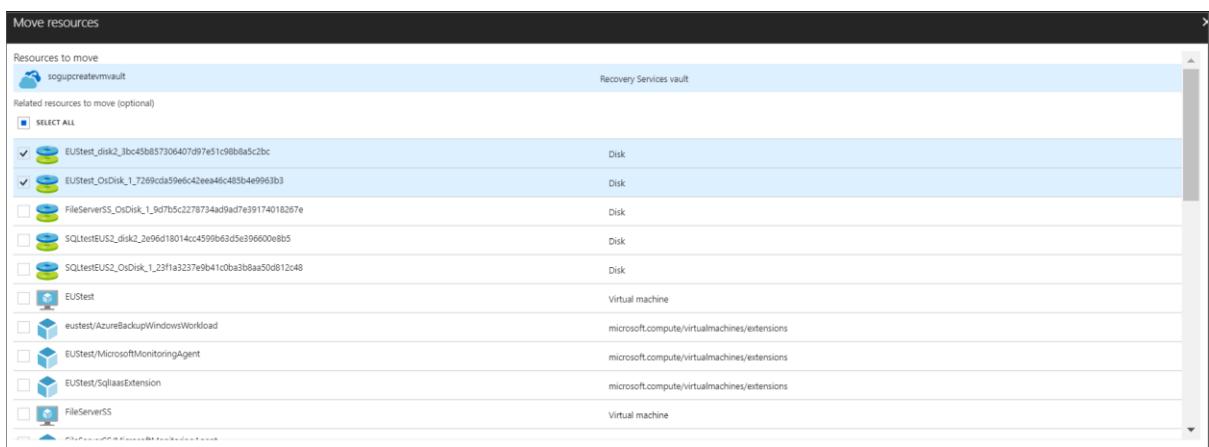
If you don't see the **Essentials** information for your vault, select the drop-down icon. You should now see the Essentials information for your vault.

The screenshot shows the same Azure Recovery Services vault overview page as the previous one, but with a red box highlighting the dropdown arrow next to the 'Resource group' field in the center column. This field currently displays 'Resource group (change)' followed by the value 'anurag-sql'. The rest of the interface remains identical to the first screenshot.

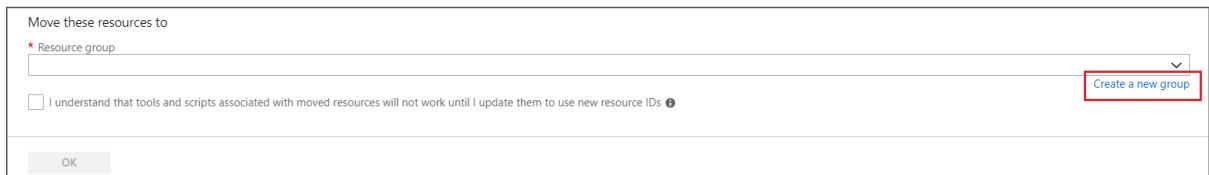
3. In the vault overview menu, select **change** next to the **Resource group**, to open the **Move resources** pane.



- In the **Move resources** pane, for the selected vault it's recommended to move the optional related resources by selecting the checkbox as shown in the following image.



- To add the target resource group, in the **Resource group** drop-down list, select an existing resource group or select **create a new group** option.



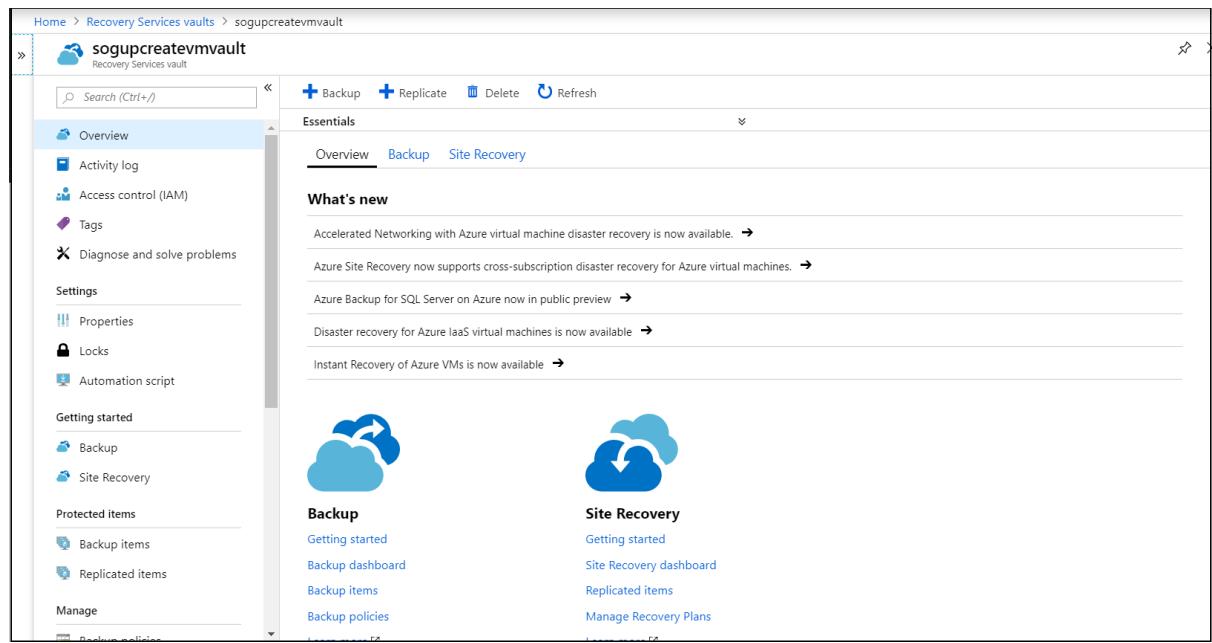
- After adding the resource group, confirm **I understand that tools and scripts associated with moved resources will not work until I update them to use new resource IDs** option and then select **OK** to complete moving the vault.



Use Azure portal to move Recovery Services vault to a different subscription

You can move a Recovery Services vault and its associated resources to a different subscription

1. Sign in to the [Azure portal](#).
2. Open the list of Recovery Services vaults and select the vault you want to move. When the vault dashboard opens, it appears as shown the following image.



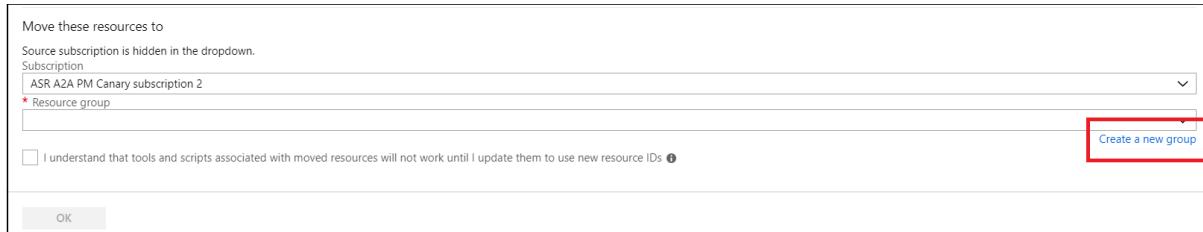
If you don't see the **Essentials** information for your vault, select the drop-down icon. You should now see the Essentials information for your vault.

3. In the vault overview menu, select **change** next to **Subscription**, to open the **Move resources** pane.

4. Select the resources to be moved, here we recommend you to use the **Select All** option to select all the listed optional resources.

Resource Type	Resource Name	Description
Disk	EUSTest_disk2_3bc45b857306407d97e51c98b8a5c2b	
Disk	EUSTest_OsDisk_1_2f69cd59edc42ee46c483b4e9963b3	
Disk	FileServer55_OsDisk_1_9d7b5c2278734ad9ad7e39174018267e	
Disk	SQLTestEUUS2_disk2_2e96d18014cc4599be63e396600e8b5	
Disk	SQLTestEUUS2_OsDisk_1_2f3f1a3237e9b41c0ba3b8aa50d12c48	
Virtual machine	EUSTest	
microsoft.compute/virtualmachines/extensions	eustest/AzureBackupWindowsWorkload	
microsoft.compute/virtualmachines/extensions	EUSTest/MicrosoftMonitoringAgent	
microsoft.compute/virtualmachines/extensions	EUSTest/SqlIaasExtension	

5. Select the target subscription from the **Subscription** drop-down list, where you want the vault to be moved.
6. To add the target resource group, in the **Resource group** drop-down list, select an existing resource group or select **create a new group** option.



7. Select **I understand that tools and scripts associated with moved resources will not work until I update them to use new resource IDs** option to confirm, and then select **OK**.

Note

Cross subscription backup (RS vault and protected VMs are in different subscriptions) isn't a supported scenario. Also, storage redundancy option from local redundant storage (LRS) to global redundant storage (GRS) and vice versa can't be modified during the vault move operation.

Use Azure portal to back up resources in Recovery Services vault after moving across regions

Azure Resource Mover supports the movement of multiple resources across regions. While moving your resources from one region to another, you can ensure that your resources stay protected. As Azure Backup supports protection of several workloads, you may need to take some steps to continue having the same level of protection in the new region.

To understand the detailed steps to achieve this, refer to the sections below.

Note

- Azure Backup currently doesn't support the movement of backup data from one Recovery Services vault to another. To protect your resource in the new region, the resource needs to be registered and backed up to a new/existing vault in the new region. When moving your resources from one region to another, backup data in your existing Recovery Services vaults in the older region can be retained/deleted based on your requirement. If you choose to retain data in the old vaults, you will incur backup charges accordingly.
- After resource move, to ensure continued security for backed-up resources in a vault that was configured with Multi-User Authorization (MUA), the destination vault should be configured with MUA using a Resource Guard in the destination region. This is because the Resource Guard and the vault must be located in the same region; therefore, the Resource Guard for the source vault can't be used to enable MUA on the destination vault.

Back up Azure Virtual Machine after moving across regions

When an Azure Virtual Machine (VM) that's been protected by a Recovery Services vault is moved from one region to another, it can no longer be backed up to the older vault. The backups in the old vault will start failing with the errors **BCMV2VMNotFound** or **ResourceNotFound**. For information on how to protect your VMs in the new region, see the following sections.

Prepare to move Azure VMs

Before you move a VM, ensure the following prerequisites are met:

1. See the [prerequisites associated with VM move](#) and ensure that the VM is eligible for move.
2. [Select the VM on the Backup Items tab](#) of existing vault's dashboard and select **Stop protection** followed by retain/delete data as per your requirement. When the backup data for a VM is stopped with retain data, the recovery points remain forever and don't adhere to any policy. This ensures you always have your backup data ready for restore.

Note

Retaining data in the older vault will incur backup charges. If you no longer wish to retain data to avoid billing, you need to delete the retained backup data using the [Delete data option](#).

3. Ensure that the VMs are turned on. All VMs' disks that need to be available in the destination region are attached and initialized in the VMs.
4. Ensure that VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do so:
 - On Windows VMs, install the latest Windows updates.
 - On Linux VMs, refer to distributor guidance to ensure that machines have the latest certificates and CRL.
5. Allow outbound connectivity from VMs:
 - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to [these URLs](#).
 - If you're using network security group (NSG) rules to control outbound connectivity, create [these service tag rules](#).

Move Azure VMs

Move your VM to the new region using [Azure Resource Mover](#).

Protect Azure VMs using Azure Backup

Start protecting your VM in a new or existing Recovery Services vault in the new region. When you need to restore from your older backups, you can still do it from your old Recovery Services vault if you had chosen to retain the backup data.

The above steps should help ensure that your resources are being backed up in the new region as well.

Back up Azure File Share after moving across regions

Azure Backup offers [a snapshot management solution](#) for your Azure Files today. This means, you don't move the file share data into the Recovery Services vaults. Also, as the snapshots don't move with your Storage Account, you'll effectively have all your backups (snapshots) in the existing region only and protected by the existing vault. However, if you move your Storage Accounts along with the file shares across regions or create new file shares in the new region, see to the following sections to ensure that they are protected by Azure Backup.

Prepare to move Azure File Share

Before you move the Storage Account, ensure the following prerequisites are met:

1. See the [prerequisites to move Storage Account](#).
2. Export and modify a Resource Move template. For more information, see [Prepare Storage Account for region move](#).

Move Azure File Share

To move your Storage Accounts along with the Azure File Shares in them from one region to another, see [Move an Azure Storage account to another region](#).

Note

When Azure File Share is copied across regions, its associated snapshots don't move along with it. In order to move the snapshots data to the new region, you need to move the individual files and directories of the snapshots to the Storage Account in the new region using [AzCopy](#).

Protect Azure File share using Azure Backup

Start protecting the Azure File Share copied into the new Storage Account in a new or existing Recovery Services vault in the new region.

Once the Azure File Share is copied to the new region, you can choose to stop protection and retain/delete the snapshots (and the corresponding recovery points) of the original Azure File Share as per your requirement. This can be done by selecting your file share on the [Backup Items tab](#) of the original vault's dashboard. When the backup data for Azure File Share is stopped with retain data, the recovery points remain forever and don't adhere to any policy.

This ensures that you will always have your snapshots ready for restore from the older vault.

Back up SQL Server/SAP HANA in Azure VM after moving across regions

When you move a VM running SQL or SAP HANA servers to another region, the SQL and SAP HANA databases in those VMs can no longer be backed up in the vault of the earlier region. To protect the SQL and SAP HANA servers running in Azure VM in the new region, see the following sections.

Prepare to move SQL Server/SAP HANA in Azure VM

Before you move SQL Server/SAP HANA running in a VM to a new region, ensure the following prerequisites are met:

1. See the [prerequisites associated with VM move](#) and ensure that the VM is eligible for move.
2. Select the VM on the [Backup Items tab](#) of the existing vault's dashboard and select *the databases* for which backup needs to be stopped. Select **Stop protection** followed by retain/delete data as per your requirement. When the backup data is stopped with retain data, the recovery points remain forever and don't adhere to any policy. This ensures that you always have your backup data ready for restore.

Note

Retaining data in the older vault will incur backup charges. If you no longer wish to retain data to avoid billing, you need to delete the retained backup data using [Delete data option](#).

3. Ensure that the VMs to be moved are turned on. All VMs disks that need to be available in the destination region are attached and initialized in the VMs.

4. Ensure that VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do so:
 - On Windows VMs, install the latest Windows updates.
 - On Linux VMs, refer to the distributor guidance and ensure that machines have the latest certificates and CRL.
5. Allow outbound connectivity from VMs:
 - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to [these URLs](#).
 - If you're using network security group (NSG) rules to control outbound connectivity, create [these service tag rules](#).

Move SQL Server/SAP HANA in Azure VM

Move your VM to the new region using [Azure Resource Mover](#).

Protect SQL Server/SAP HANA in Azure VM using Azure Backup

Start protecting the VM in a new/existing Recovery Services vault in the new region. When you need to restore from your older backups, you can still do it from your old Recovery Services vault.

The above steps should help ensure that your resources are being backed up in the new region as well.

Use PowerShell to move Recovery Services vault

To move a Recovery Services vault to another resource group, use the `Move-AzureRMResource` cmdlet. `Move-AzureRMResource` requires the resource name and type of resource. You can get both from the `Get-AzureRmRecoveryServicesVault` cmdlet.

PowerShellCopy

```
$destinationRG = "<destinationResourceGroupName>"  
$vault = Get-AzureRmRecoveryServicesVault -Name <vaultname> -ResourceGroupName  
<vaultRGname>  
Move-AzureRmResource -DestinationResourceGroupName $destinationRG -ResourceId  
$vault.ID
```

To move the resources to different subscription, include the `-DestinationSubscriptionId` parameter.

PowerShellCopy

```
Move-AzureRmResource -DestinationSubscriptionId "<destinationSubscriptionID>" -  
DestinationResourceGroupName $destinationRG -ResourceId $vault.ID
```

After executing the above cmdlets, you'll be asked to confirm that you want to move the specified resources. Type **Y** to confirm. After a successful validation, the resource moves.

Use CLI to move Recovery Services vault

To move a Recovery Services vault to another resource group, use the following cmdlet:

Azure CLICopy

```
az resource move --destination-group <destinationResourceGroupName> --ids  
<VaultResourceID>
```

To move to a new subscription, provide the --destination-subscription-id parameter.

Post migration

1. Set/verify the access controls for the resource groups.
2. The Backup reporting and monitoring feature needs to be configured again for the vault after the move completes. The previous configuration will be lost during the move operation.

Move an Azure virtual machine to a different recovery service vault.

If you want to move an Azure virtual machine that has backup enabled, then you have two choices. They depend on your business requirements:

- [Don't need to preserve previous backed-up data](#)
- [Must preserve previous backed-up data](#)

Don't need to preserve previous backed-up data

To protect workloads in a new vault, the current protection and data will need to be deleted in the old vault and backup is configured again.

Warning

The following operation is destructive and can't be undone. All backup data and backup items associated with the protected server will be permanently deleted. Proceed with caution.

Stop and delete current protection on the old vault:

1. Disable soft delete in the vault properties. Follow [these steps](#) to disable soft delete.
2. Stop protection and delete backups from the current vault. In the Vault dashboard menu, select **Backup Items**. Items listed here that need to be moved to the new vault must be removed along with their backup data. See how to [delete protected items in the cloud](#) and [delete protected items on premises](#).
3. If you're planning to move AFS (Azure file shares), SQL servers or SAP HANA servers, then you'll need also to unregister them. In the vault dashboard menu, select **Backup Infrastructure**. See how to [unregister the SQL server](#), [unregister a storage account associated with Azure file shares](#), and [unregister an SAP HANA instance](#).
4. Once they're removed from the old vault, continue to configure the backups for your workload in the new vault.

Must preserve previous backed-up data

If you need to keep the current protected data in the old vault and continue the protection in a new vault, there are limited options for some of the workloads:

- For MARS, you can [stop protection with retain data](#) and register the agent in the new vault.
 - Azure Backup service will continue to retain all the existing recovery points of the old vault.
 - You'll need to pay to keep the recovery points in the old vault.
 - You'll be able to restore the backed-up data only for unexpired recovery points in the old vault.
 - A new initial replica of the data will need to be created on the new vault.
- For an Azure VM, you can [stop protection with retain data](#) for the VM in the old vault, move the VM to another resource group, and then protect the VM in the new vault. See [guidance and limitations](#) for moving a VM to another resource group.

A VM can be protected in only one vault at a time. However, the VM in the new resource group can be protected on the new vault as it's considered a different VM.

- Azure Backup service will retain the recovery points that have been backed up on the old vault.
- You'll need to pay to keep the recovery points in the old vault (see [Azure Backup pricing](#) for details).
- You'll be able to restore the VM, if needed, from the old vault.

- The first backup on the new vault of the VM in the new resource will be an initial replica.

Move virtual machines to resource group or subscription

- Article
- 04/09/2023
- 7 contributors

Feedback

In this article

1. [Scenarios not supported](#)
2. [Azure disk encryption](#)
3. [Virtual machines with Marketplace plans](#)
4. [Virtual machines with Azure Backup](#)
5. [Next steps](#)

This article describes how to move a virtual machine to a new resource group or Azure subscription.

If you want to move a virtual machine to a new region, see [Tutorial: Move Azure VMs across regions](#).

Scenarios not supported

The following scenarios aren't yet supported:

- Virtual Machine Scale Sets with Standard SKU Load Balancer or Standard SKU Public IP can't be moved.
- Virtual machines in an existing virtual network can be moved to a new subscription only when the virtual network and all of its dependent resources are also moved.
- Virtual machines created from Marketplace resources with plans attached can't be moved across subscriptions. For a potential workaround, see [Virtual machines with Marketplace plans](#).
- Low-priority virtual machines and low-priority virtual machine scale sets can't be moved across resource groups or subscriptions.
- Virtual machines in an availability set can't be moved individually.

Azure disk encryption

A virtual machine that is integrated with a key vault to implement [Azure Disk Encryption for Linux VMs](#) or [Azure Disk Encryption for Windows VMs](#) can be moved to another resource group when it is in deallocated state.

However, to move such virtual machine to another subscription, you must disable encryption.

- [Azure CLI](#)
- [PowerShell](#)

Azure CLICopy

Open Cloudshell

```
az vm encryption disable --resource-group demoRG --name myVm1 --volume-type all
```

Virtual machines with Marketplace plans

Virtual machines created from Marketplace resources with plans attached can't be moved across subscriptions. To work around this limitation, you can de-provision the virtual machine in the current subscription, and deploy it again in the new subscription. The following steps help you recreate the virtual machine in the new subscription. However, they might not work for all scenarios. If the plan is no longer available in the Marketplace, these steps won't work.

1. Get information about the plan.

- [Azure CLI](#)
- [PowerShell](#)

Azure CLICopy

```
az vm show --resource-group demoRG --name myVm1 --query plan
```

2. Check that the offering still exists in the Marketplace.

- [Azure CLI](#)
- [PowerShell](#)

Azure CLICopy

```
az vm image list-skus --publisher Fabrikam --offer LinuxServer --location centralus
```

3. Either clone the OS disk to the destination subscription, or move the original disk after deleting the virtual machine from source subscription.

4. In the destination subscription, accept the Marketplace terms for your plan. You can accept the terms by running the following PowerShell command:

- o [Azure CLI](#)
- o [PowerShell](#)

Azure CLICopy

```
az vm image terms accept --publisher {publisher} --offer {product/offer} --plan {name/SKU}
```

Or, you can create a new instance of a virtual machine with the plan through the portal. You can delete the virtual machine after accepting the terms in the new subscription.

5. In the destination subscription, recreate the virtual machine from the cloned OS disk using PowerShell, CLI, or an Azure Resource Manager template. Include the marketplace plan that's attached to the disk. The information about the plan should match the plan you purchased in the new subscription. For more information, see [Create the VM](#).

For more information, see [Move a Marketplace Azure Virtual Machine to another subscription](#).

Virtual machines with Azure Backup

To move virtual machines configured with Azure Backup, you must delete the restore points collections (snapshots) from the vault. Restore points already copied to the vault can be retained and moved.

If [soft delete](#) is enabled for your virtual machine, you can't move the virtual machine while those restore points are kept. Either [disable soft delete](#) or wait 14 days after deleting the restore points.

Portal

1. Temporarily stop the backup and keep backup data.
2. To move virtual machines configured with Azure Backup, do the following steps:
 - a. Find the resource group that contains your backups. If you used the default resource group, it has the following naming pattern: `AzureBackupRG_<VM location>_1`. For example, the name is in the format of `AzureBackupRG_westus2_1`.

If you created a custom resource group, select that resource group. If you can't find the resource group, search for **Restore Point Collections** in the portal. Look for the collection with the naming pattern AzureBackup_<VM name>_#####.

- b. Select the resource with type **Restore Point Collection** that has the naming pattern AzureBackup_<VM name>_#####.
 - c. Delete this resource. This operation deletes only the instant recovery points, not the backed-up data in the vault.
 - d. After the delete operation is complete, you can move your virtual machine.
3. Move the VM to the target resource group.
 4. Reconfigure the backup.

Script

1. Find the resource group that contains your backups. If you used the default resource group, it has the following naming pattern: AzureBackupRG_<VM location>_1. For example, the name is in the format of *AzureBackupRG_westus2_1*.

If you created a custom resource group, find that resource group. If you can't find the resource group, use the following command and provide the name of the virtual machine.

- o [Azure CLI](#)
- o [PowerShell](#)

Azure CLICopy

Open Cloudshell

```
az resource list --resource-type Microsoft.Compute/restorePointCollections --query "[?starts_with(name, 'AzureBackup_<vm-name>')].resourceGroup"
```

2. If you're moving only one virtual machine, get the restore point collection for that virtual machine.

- o [Azure CLI](#)
- o [PowerShell](#)

Azure CLICopy

Open Cloudshell

```
RESTOREPOINTCOL=$(az resource list -g AzureBackupRG_<VM location>_1 --resource-type Microsoft.Compute/restorePointCollections --query "[?starts_with(name, 'AzureBackup_<VM name>')].id" --output tsv)
```

Delete this resource. This operation deletes only the instant recovery points, not the backed-up data in the vault.

- [Azure CLI](#)
- [PowerShell](#)

Azure CLICopy

Open Cloudshell

```
az resource delete --ids $RESTOREPOINTCOL
```

3. If you're moving all the virtual machines with back ups in this location, get the restore point collections for those virtual machines.

- [Azure CLI](#)
- [PowerShell](#)

Azure CLICopy

Open Cloudshell

```
RESTOREPOINTCOL=$(az resource list -g AzureBackupRG_<VM location>_1 --resource-type Microsoft.Compute/restorePointCollections)
```

Delete each resource. This operation deletes only the instant recovery points, not the backed-up data in the vault.

- [Azure CLI](#)
- [PowerShell](#)

Azure CLICopy

Open Cloudshell

```
az resource delete --ids $RESTOREPOINTCOL
```

How to move Azure API Management across regions

- Article
- 08/16/2022
- 3 contributors

Feedback

In this article

1. [Availability](#)
2. [Considerations](#)
3. [Prerequisites](#)
4. [Prepare and move](#)

Show 3 more

This article describes how to move an API Management instance to a different Azure region. You might move your instance to another region for many reasons. For example:

- Locate your instance closer to your API consumers
- Deploy features available in specific regions only
- Meet internal policy and governance requirements

To move API Management instances from one Azure region to another, use the service's [backup and restore](#) operations. You can use a different API Management instance name or the existing name.

Note

API Management also supports [multi-region deployment](#), which distributes a single Azure API management service across multiple Azure regions. Multi-region deployment helps reduce request latency perceived by geographically distributed API consumers and improves service availability if one region goes offline.

Availability

Important

This feature is available in the **Premium**, **Standard**, **Basic**, and **Developer** tiers of API Management.

Considerations

- Choose the same API Management pricing tier in the source and target regions.
- Backup and restore won't work when migrating between different cloud types. For that scenario, export the resource [as a template](#). Then, adapt the exported template for the target Azure region and re-create the resource.

Prerequisites

- Review requirements and limitations of the API Management [backup and restore](#) operations.

- See [What is not backed up](#). Record settings and data that you will need to recreate manually after moving the instance.
- Create a [storage account](#) in the source region. You will use this account to back up the source instance.

Prepare and move

Option 1: Use a different API Management instance name

1. In the target region, create a new API Management instance with the same pricing tier as the source API Management instance. Use a different name for the new instance.
2. [Back up](#) the existing API Management instance to the storage account.
3. [Restore](#) the source instance's backup to the new API Management instance.
4. If you have a custom domain pointing to the source region API Management instance, update the custom domain CNAME to point to the new API Management instance.

Option 2: Use the same API Management instance name

Warning

This option deletes the original API Management instance and results in downtime during the migration. Ensure that you have a valid backup before deleting the source instance.

1. [Back up](#) the existing API Management instance to the storage account.
2. Delete the API Management instance in the source region.
3. Create a new API Management instance in the target region with the same name as the one in the source region.
4. [Restore](#) the source instance's backup to the new API Management instance in the target region.

Verify

1. Ensure that the restore operation completes successfully before accessing your API Management instance in the target region.
2. Configure settings that are not automatically moved during the restore operation. Examples: virtual network configuration, managed identities, developer portal content, and custom domain and custom CA certificates.
3. Access your API Management endpoints in the target region. For example, test your APIs, or access the developer portal.

Clean up source resources

If you moved the API Management instance using Option 1, after you successfully restore and configure the target instance, you may delete the source instance.

Move an App Service resource to another region

- Article
- 02/01/2023
- 3 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare](#)
3. [Move](#)
4. [Clean up source resources](#)
5. [Next steps](#)

This article describes how to move App Service resources to a different Azure region. You might move your resources to another region for a number of reasons. For example, to take advantage of a new Azure region, to deploy features or services available in specific regions only, to meet internal policy and governance requirements, or in response to capacity planning requirements.

App Service resources are region-specific and can't be moved across regions. You must create a copy of your existing App Service resources in the target region, then move your content over to the new app. If your source app uses a custom domain, you can [migrate it to the new app in the target region](#) when you're finished.

To make copying your app easier, you can [clone an individual App Service app](#) into an App Service plan in another region, but it does have [limitations](#), especially that it doesn't support Linux apps.

Prerequisites

- Make sure that the App Service app is in the Azure region from which you want to move.
- Make sure that the target region supports App Service and any related service, whose resources you want to move.

Prepare

Identify all the App Service resources that you're currently using. For example:

- App Service apps
- [App Service plans](#)
- [Deployment slots](#)
- [Custom domains purchased in Azure](#)
- [TLS/SSL certificates](#)
- [Azure Virtual Network integration](#)
- [Hybrid connections](#).
- [Managed identities](#)
- [Backup settings](#)

Certain resources, such as imported certificates or hybrid connections, contain integration with other Azure services. For information on how to move those resources across regions, see the documentation for the respective services.

Move

1. [Create a back up of the source app.](#)
2. [Create an app in a new App Service plan, in the target region.](#)
3. [Restore the back up in the target app](#)
4. If you use a custom domain, [bind it preemptively to the target app](#) with asuid. and [enable the domain in the target app](#).
5. Configure everything else in your target app to be the same as the source app and verify your configuration.
6. When you're ready for the custom domain to point to the target app, [remap the domain name](#).

Clean up source resources

Delete the source app and App Service plan. [An App Service plan in the non-free tier carries a charge, even if no app is running in it.](#)

Management of Azure Automation data

- Article
- 05/26/2023
- 16 contributors

Feedback

In this article

1. [TLS 1.2 for Azure Automation](#)

2. [Data retention](#)
3. [Data backup](#)
4. [Data residency](#)
5. [Next steps](#)

This article contains several topics explaining how data is protected and secured in an Azure Automation environment.

TLS 1.2 for Azure Automation

To ensure the security of data in transit to Azure Automation, we strongly encourage you to configure the use of Transport Layer Security (TLS) 1.2. The following are a list of methods or clients that communicate over HTTPS to the Automation service:

- Webhook calls
- Hybrid Runbook Workers, which include machines managed by Update Management and Change Tracking and Inventory.
- DSC nodes

Older versions of TLS/Secure Sockets Layer (SSL) have been found to be vulnerable and while they still currently work to allow backwards compatibility, they are **not recommended**. We do not recommend explicitly setting your agent to only use TLS 1.2 unless its necessary, as it can break platform level security features that allow you to automatically detect and take advantage of newer more secure protocols as they become available, such as TLS 1.3.

For information about TLS 1.2 support with the Log Analytics agent for Windows and Linux, which is a dependency for the Hybrid Runbook Worker role, see [Log Analytics agent overview - TLS 1.2](#).

Platform-specific guidance

Platform/Language	Support	More Information
Linux	Linux distributions tend to rely on OpenSSL for TLS 1.2 support.	Check the OpenSSL Changelog to confirm your version of OpenSSL is supported.
Windows 8.0 - 10	Supported, and enabled by default.	To confirm that you are still using the default settings .
Windows Server 2012 - 2016	Supported, and enabled by default.	To confirm that you are still using the default settings

Platform/Language	Support	More Information
Windows 7 SP1 and Windows Server 2008 R2 SP1 default.	Supported, but not enabled by default.	See the Transport Layer Security (TLS) registry settings page for details on how to enable.

Data retention

When you delete a resource in Azure Automation, it's retained for many days for auditing purposes before permanent removal. You can't see or use the resource during this time. This policy also applies to resources that belong to a deleted Automation account. The retention policy applies to all users and currently can't be customized. However, if you need to keep data for a longer period, you can [forward Azure Automation job data to Azure Monitor logs](#).

The following table summarizes the retention policy for different resources.

Data	Policy
Accounts	An account is permanently removed 30 days after a user deletes it.
Assets	An asset is permanently removed 30 days after a user deletes it, or 30 days after a user deletes an account that holds the asset. Assets include variables, schedules, credentials, certificates, Python 2 packages, and connections.
DSC Nodes	A DSC node is permanently removed 30 days after being unregistered from an Automation account using Azure portal or the Unregister-AzAutomationDscNode cmdlet in Windows PowerShell. A node is also permanently removed 30 days after a user deletes the account that holds the node.
Jobs	A job is deleted and permanently removed 30 days after modification, for example, after the job completes, is stopped, or is suspended.
Modules	A module is permanently removed 30 days after a user deletes it, or 30 days after a user deletes the account that holds the module.
Node Configurations/MOF Files	An old node configuration is permanently removed 30 days after a new node configuration is generated.
Node Reports	A node report is permanently removed 90 days after a new report is generated for that node.
Runbooks	A runbook is permanently removed 30 days after a user deletes the resource, or 30 days after a user deletes the account that holds the resource ¹ .

¹The runbook can be recovered within the 30-day window by filing an Azure support incident with Microsoft Azure Support. Go to the [Azure support site](#) and select **Submit a support request**.

Data backup

When you delete an Automation account in Azure, all objects in the account are deleted. The objects include runbooks, modules, configurations, settings, jobs, and assets. You can [recover](#) a deleted Automation account within 30 days. You can also use the following information to back up the contents of your Automation account before deleting it:

Runbooks

You can export your runbooks to script files using either the Azure portal or the [Get-AzureAutomationRunbookDefinition](#) cmdlet in Windows PowerShell. You can import these script files into another Automation account, as discussed in [Manage runbooks in Azure Automation](#).

Integration modules

You can't export integration modules from Azure Automation, they have to be made available outside of the Automation account.

Assets

You can't export Azure Automation assets: certificates, connections, credentials, schedules, and variables. Instead, you can use the Azure portal and Azure cmdlets to note the details of these assets. Then use these details to create any assets that are used by runbooks that you import into another Automation account.

You can't retrieve the values for encrypted variables or the password fields of credentials using cmdlets. If you don't know these values, you can retrieve them in a runbook. For retrieving variable values, see [Variable assets in Azure Automation](#). To find out more about retrieving credential values, see [Credential assets in Azure Automation](#).

DSC configurations

You can export your DSC configurations to script files using either the Azure portal or the [Export-AzAutomationDscConfiguration](#) cmdlet in Windows PowerShell. You can import and use these configurations in another Automation account.

Data residency

You specify a region during the creation of an Azure Automation account. Service data such as assets, configuration, logs are stored in that region and may transit or be processed in other regions within the same geography. These global endpoints are necessary to provide end-users with a high-performance, low-latency experience regardless of location. Only for the Brazil South (Sao Paulo State) region of Brazil geography, Southeast Asia region (Singapore) and East Asia region (Hongkong) of the Asia Pacific geography, we store Azure Automation data in the same region to accommodate data-residency requirements for these regions.

Back up resources in Recovery Services vault after moving across regions

- Article
- 01/31/2023
- 4 contributors

Feedback

In this article

1. [Back up Azure Virtual Machine after moving across regions](#)
2. [Back up Azure File Share after moving across regions](#)
3. [Back up SQL Server/SAP HANA in Azure VM after moving across regions](#)

Azure Resource Mover supports the movement of multiple resources across regions. While moving your resources from one region to another, you can ensure that your resources stay protected. As Azure Backup supports protection of several workloads, you may need to take some steps to continue having the same level of protection in the new region.

To understand the detailed steps to achieve this, refer to the sections below.

Note

Azure Backup currently doesn't support the movement of backup data from one Recovery Services vault to another. To protect your resource in the new region, the resource needs to be registered and backed up to a new/existing vault in the new region. When moving your resources from one region to another, backup data in

your existing Recovery Services vaults in the older region can be retained/deleted based on your requirement. If you choose to retain data in the old vaults, you will incur backup charges accordingly.

Back up Azure Virtual Machine after moving across regions

When an Azure Virtual Machine (VM) that's been protected by a Recovery Services vault is moved from one region to another, it can no longer be backed up to the older vault. The backups in the old vault will start failing with the errors **BCMV2VMNotFound** or **ResourceNotFound**. For information on how to protect your VMs in the new region, see the following sections.

Prepare to move Azure VMs

Before you move a VM, ensure the following prerequisites are met:

1. See the [prerequisites associated with VM move](#) and ensure that the VM is eligible for move.
2. [Select the VM on the Backup Items tab](#) of existing vault's dashboard and select **Stop protection** followed by retain/delete data as per your requirement. When the backup data for a VM is stopped with retain data, the recovery points remain forever and don't adhere to any policy. This ensures you always have your backup data ready for restore.

Note

Retaining data in the older vault will incur backup charges. If you no longer wish to retain data to avoid billing, you need to delete the retained backup data using the [Delete data option](#).

3. Ensure that the VMs are turned on. All VMs' disks that need to be available in the destination region are attached and initialized in the VMs.
4. Ensure that VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do so:
 - On Windows VMs, install the latest Windows updates.
 - On Linux VMs, refer to distributor guidance to ensure that machines have the latest certificates and CRL.
5. Allow outbound connectivity from VMs:
 - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to [these URLs](#).
 - If you're using network security group (NSG) rules to control outbound connectivity, create [these service tag rules](#).

Move Azure VMs

Move your VM to the new region using [Azure Resource Mover](#).

Protect Azure VMs using Azure Backup

Start protecting your VM in a new or existing Recovery Services vault in the new region. When you need to restore from your older backups, you can still do it from your old Recovery Services vault if you had chosen to retain the backup data.

The above steps should help ensure that your resources are being backed up in the new region as well.

Back up Azure File Share after moving across regions

Azure Backup offers [a snapshot management solution](#) for your Azure Files today. This means, you don't move the file share data into the Recovery Services vaults. Also, as the snapshots don't move with your Storage Account, you'll effectively have all your backups (snapshots) in the existing region only and protected by the existing vault. However, if you move your Storage Accounts along with the file shares across regions or create new file shares in the new region, see to the following sections to ensure that they are protected by Azure Backup.

Prepare to move Azure File Share

Before you move the Storage Account, ensure the following prerequisites are met:

1. See the [prerequisites to move Storage Account](#).
2. Export and modify a Resource Move template. For more information, see [Prepare Storage Account for region move](#).

Move Azure File Share

To move your Storage Accounts along with the Azure File Shares in them from one region to another, see [Move an Azure Storage account to another region](#).

Note

When Azure File Share is copied across regions, its associated snapshots don't move along with it. In order to move the snapshots data to the new region, you need to move the individual files and directories of the snapshots to the Storage Account in the new region using [AzCopy](#).

Protect Azure File share using Azure Backup

Start protecting the Azure File Share copied into the new Storage Account in a new or existing Recovery Services vault in the new region.

Once the Azure File Share is copied to the new region, you can choose to stop protection and retain/delete the snapshots (and the corresponding recovery points) of the original Azure File Share as per your requirement. This can be done by selecting your file share on the [Backup Items tab](#) of the original vault's dashboard. When the backup data for Azure File Share is stopped with retain data, the recovery points remain forever and don't adhere to any policy.

This ensures that you will always have your snapshots ready for restore from the older vault.

Back up SQL Server/SAP HANA in Azure VM after moving across regions

When you move a VM running SQL or SAP HANA servers to another region, the SQL and SAP HANA databases in those VMs can no longer be backed up in the vault of the earlier region. To protect the SQL and SAP HANA servers running in Azure VM in the new region, see the follow sections.

Prepare to move SQL Server/SAP HANA in Azure VM

Before you move SQL Server/SAP HANA running in a VM to a new region, ensure the following prerequisites are met:

1. See the [prerequisites associated with VM move](#) and ensure that the VM is eligible for move.
2. Select the VM on the [Backup Items tab](#) of the existing vault's dashboard and select *the databases* for which backup needs to be stopped. Select **Stop protection** followed by retain/delete data as per your requirement. When the backup data is stopped with retain data, the recovery points remain forever and don't adhere to any policy. This ensures that you always have your backup data ready for restore.

Note

Retaining data in the older vault will incur backup charges. If you no longer wish to retain data to avoid billing, you need to delete the retained backup data using [Delete data option](#).

3. Ensure that the VMs to be moved are turned on. All VMs disks that need to be available in the destination region are attached and initialized in the VMs.
4. Ensure that VMs have the latest trusted root certificates, and an updated certificate revocation list (CRL). To do so:
 - On Windows VMs, install the latest Windows updates.

- On Linux VMs, refer to the distributor guidance and ensure that machines have the latest certificates and CRL.
5. Allow outbound connectivity from VMs:
 - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to [these URLs](#).
 - If you're using network security group (NSG) rules to control outbound connectivity, create [these service tag rules](#).

Move SQL Server/SAP HANA in Azure VM

Move your VM to the new region using [Azure Resource Mover](#).

Protect SQL Server/SAP HANA in Azure VM using Azure Backup

Start protecting the VM in a new/existing Recovery Services vault in the new region. When you need to restore from your older backups, you can still do it from your old Recovery Services vault.

The above steps should help ensure that your resources are being backed up in the new region as well.

Move an Azure Batch account to another region

- Article
- 02/28/2023
- 5 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare the template](#)
3. [Move the account](#)
4. [Discard or clean up](#)
5. [Next steps](#)

There are scenarios where you might want to move an existing [Azure Batch account](#) from one region to another. For example, you might want to move for disaster recovery planning. This article explains how to move a Batch account between regions using the Azure portal.

Moving Batch accounts directly from one region to another isn't possible. You can use an Azure Resource Manager template (ARM template) to export the existing configuration of your Batch account instead. Then, stage the resource in another region. First, export the Batch account to a template. Next, modify the parameters to match the destination region. Deploy the modified template to the new region. Last, recreate jobs and other features in the account.

For more information on Resource Manager and templates, see [Quickstart: Create and deploy Azure Resource Manager templates by using the Azure portal](#).

Prerequisites

- Make sure that the services and features that your Batch account uses are supported in the new target region.
- It's recommended to move any Azure resources associated with your Batch account to the new target region. For example, follow the steps in [Move an Azure Storage account to another region](#) to move an associated autostorage account. If you prefer, you can leave resources in the original region, however, performance is typically better when your Batch account is in the same region as your other Azure resources used by your workload. This article assumes you've already migrated your storage account or any other regional Azure resources to be aligned with your Batch account.

Prepare the template

To get started, you need to export and then modify an ARM template.

Export a template

Export an ARM template that contains settings and information for your Batch account.

1. Sign in to the [Azure portal](#).
2. Select **All resources** and then select your Batch account.
3. Select > **Automation** > **Export template**.
4. Choose **Download** in the **Export template** pane.
5. Locate the .zip file that you downloaded from the portal. Unzip that file into a folder of your choice.

This zip file contains the .json files that make up the template. The file also includes scripts to deploy the template.

Modify the template

Load and modify the template so you can create a new Batch account in the target region.

1. In the Azure portal, select **Create a resource**.
2. In **Search the Marketplace**, type **template deployment**, and then press **ENTER**.
3. Select **Template deployment (deploy using custom templates)**.
4. Select **Create**.
5. Select **Build your own template in the editor**.
6. Select **Load file**, and then select the **template.json** file that you downloaded in the last section.
7. In the uploaded **template.json** file, name the target Batch account by entering a new **defaultValue** for the Batch account name. This example sets the **defaultValue** of the Batch account name to `mytargetaccount` and replaces the string in **defaultValue** with the resource ID for `mytargetstorageaccount`.

JSONCopy

```
{  
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
    "contentVersion": "1.0.0.0",  
    "parameters": {  
        "batchAccounts_mysourceaccount_name": {  
            "defaultValue": "mytargetaccount",  
            "type": "String"  
        }  
    },
```

8. Next, update the **defaultValue** of the storage account with your migrated storage account's resource ID. To get this value, navigate to the storage account in the Azure portal, select **JSON View** near the top of the screen, and then copy the value shown under **Resource ID**. This example uses the resource ID for a storage account named `mytargetstorageaccount` in the resource group `mytargetresourcegroup`.

JSONCopy

```
"storageAccounts_myresourcestorageaccount_externalid": {  
    "defaultValue":  
        "/subscriptions/{subscriptionID}/resourceGroups/mytargetresourcegroup/providers/Microsoft.Storage/storageAccounts/mytargetstorageaccount",  
        "type": "String"  
    }  
},
```

- Finally, edit the **location** property to use your target region. This example sets the target region to **centralus**.

JSONCopy

```
{  
  "resources": [  
    {  
      "type": "Microsoft.Batch/batchAccounts",  
      "apiVersion": "2021-01-01",  
      "name":  
        "[parameters('batchAccounts_mysourceaccount_name')]",  
      "location": "centralus",
```

To obtain region location codes, see [Azure Locations](#). The code for a region is the region name with no spaces. For example, **Central US = centralus**.

Move the account

Deploy the template to create a new Batch account in the target region.

- Now that you've made your modifications, select **Save** below the **template.json** file.
- Enter or select the property values:
 - Subscription**: Select an Azure subscription.
 - Resource group**: Select the resource group that you created when moving the associated storage account.
 - Region**: Select the Azure region where you want to move the account.
- Select **Review and create**, then select **Create**.

Configure the new Batch account

Some features don't export to a template, so you have to recreate them in the new Batch account. These features include:

- Jobs (and tasks)
- Job schedules
- Certificates
- Application packages

Be sure to configure features in the new account as needed. You can look at how you've configured these features in your source Batch account for reference.

Important

New Batch accounts are entirely separate from any prior existing Batch accounts, even within the same region. These newly created Batch accounts will have [default service and core quotas](#) associated with them. For User Subscription pool allocation mode Batch accounts, core quotas from the subscription will apply. You will need to ensure that these new Batch accounts have sufficient quota before migrating your workload.

Discard or clean up

Confirm that your new Batch account is successfully working in the new region. Also make sure to restore the necessary features. Then, you can delete the source Batch account.

1. In the Azure portal, expand the menu on the left side to open the menu of services, and choose **Batch accounts**.
2. Locate the Batch account to delete, and right-click the **More** button (...) on the right side of the listing. Be sure that you're selecting the original source Batch account, not the new one you created.
3. Select **Delete**, then confirm.

Move Azure Cache for Redis instances to different regions

- Article
- 02/07/2023
- 3 contributors

Feedback

In this article

1. [Passive geo-replication \(Premium\)](#)
2. [Create a new cache \(All tiers\)](#)
3. [Export and import data with an RDB file \(Premium, Enterprise, Enterprise Flash\)](#)
4. [Dual-write to two caches \(Basic, Standard, and Premium\)](#)

Show 2 more

In this article, you learn how to move Azure Cache for Redis instances to a different Azure region. You might move your resources to another region for many reasons:

- To take advantage of a new Azure region.
- To deploy features or services available in specific regions only.
- To meet internal policy and governance requirements.

- To respond to capacity planning requirements.

If you're looking to migrate to Azure Cache for Redis from on-premises, cloud-based VMs, or another hosting service, we recommend you see [Migrate to Azure Cache for Redis](#).

The tier of Azure Cache for Redis you use determines the option that's best for you.

Cache Tier	Options
Premium	Geo-replication, create a new cache, dual-write to two caches, export and import data via RDB file, or migrate programmatically
Basic or Standard	Create a new cache, dual-write to two caches, or migrate programmatically
Enterprise or Enterprise Flash	Create a new cache or export and import data with an RDB file, or migrate programmatically

Passive geo-replication (Premium)

Prerequisites

To configure geo-replication between two caches, the following prerequisites must be met:

- Both caches are [Premium tier](#) caches.
- Both caches are in the same Azure subscription.
- The secondary linked cache is either the same cache size or a larger cache size than the primary linked cache.
- Both caches already exist and are running.

Prepare

To move your cache instance to another region, you need to [create a second premium cache instance](#) in the desired region. Once both caches are running, you can set up geo-replication between the two cache instances.

Note

Data transfer between Azure regions is charged at standard [bandwidth rates](#).

Some features aren't supported with geo-replication:

- Zone Redundancy isn't supported with geo-replication.
- Persistence isn't supported with geo-replication.

Conditions for geo-replications support:

- Clustering is supported if both caches have clustering enabled and have the same number of shards.
- Caches in different VNets are supported with caveats. See [Can I use geo-replication with my caches in a VNet?](#) for more information.

After geo-replication is configured, the following restrictions apply to your linked cache pair:

- The secondary linked cache is read-only. You can read from it, but you can't write any data to it.
 - If you choose to read from the Geo-Secondary instance when a full data sync is happening between the Geo-Primary and the Geo-Secondary, such as when either Geo-Primary or Geo-Secondary is updated and on some reboot scenarios as well, the Geo-Secondary instance throws errors on any Redis operation against it until the full data sync between Geo-Primary and Geo-Secondary is complete.
 - Applications reading from Geo-Secondary should be built to fall back to the Geo-Primary whenever the Geo-Secondary is throwing such errors.
- Any data that was in the secondary linked cache before the link was added is removed. If the geo-replication is later removed however, the replicated data remains in the secondary linked cache.
- You can't [scale](#) either cache while the caches are linked.
- You can't [change the number of shards](#) if the cache has clustering enabled.
- You can't enable persistence on either cache.
- You can [Export](#) from either cache.
- You can't [Import](#) into the secondary linked cache.
- You can't delete either linked cache, or the resource group that contains them, until you unlink the caches. For more information, see [Why did the operation fail when I tried to delete my linked cache?](#)
- If the caches are in different regions, network egress costs apply to the data moved across regions. For more information, see [How much does it cost to replicate my data across Azure regions?](#)
- Failover is not automatic. You must start the failover from the primary to the secondary linked cache. For more information on how to fail over a client application, see [Initiate a failover from geo-primary to geo-secondary](#).

Move

1. To link two caches together for geo-replication, first select **Geo-replication** from the Resource menu of the cache that you intend to be the primary linked cache. Next, select **Add cache replication link** from **Geo-replication** on the left.

2. Select the name of your intended secondary cache from the **Compatible caches** list. If your secondary cache isn't displayed in the list, verify that the [Geo-replication prerequisites](#) for the secondary cache are met. To filter the caches by region, select the region in the map to display only those caches in the **Compatible caches** list.

You can also start the linking process or view details about the secondary cache by using the context menu.

3. Select **Link** to link the two caches together and begin the replication process.

Verify

1. You can view the progress of the replication process using **Geo-replication** on the left.

You can also view the linking status on the left, using **Overview**, for both the primary and secondary caches.

Once the replication process is complete, the **Link status** changes to **Succeeded**.

The primary linked cache remains available for use during the linking process. The secondary linked cache isn't available until the linking process completes.

Clean up source resources

Once your new cache in the targeted region is populated with all necessary data, remove the link between the two caches and delete the original instance.

1. To remove the link between two caches and stop geo-replication, select **Unlink caches** from the **Geo-replication** on the left.

When the unlinking process completes, the secondary cache is available for both reads and writes.

Note

When the geo-replication link is removed, the replicated data from the primary linked cache remains in the secondary cache.

1. Delete the original instance.

Create a new cache (All tiers)

Prerequisites

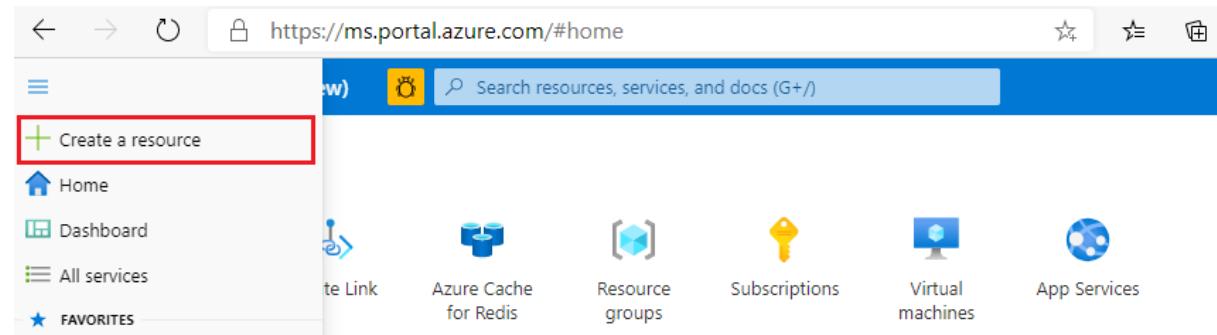
- Azure subscription - [create one for free](#)

Prepare

If you don't need to maintain your data during the move, the easiest way to move regions is to create a new cache instance in the targeted region and connect your application to it. For example, if you use Redis as a look-aside cache of database records, you can easily rebuild the cache from scratch.

Move

1. To create a cache, sign in to the [Azure portal](#) and select **Create a resource**.



2. On the **New** page, select **Databases** and then select **Azure Cache for Redis**.

New

 Search the Marketplace

Azure Marketplace [See all](#)

Get started

Recently created

AI + Machine Learning

Analytics

Blockchain

Compute

Containers

Databases

Developer Tools

DevOps

Identity

Integration

Internet of Things

IT & Management Tools

Media

Migration

Mixed Reality

Monitoring & Diagnostics

Networking

Security

Software as a Service (SaaS)

Storage

Web

Featured [See all](#)



Azure SQL Managed Instance

[Quickstarts + tutorials](#)



SQL Database

[Quickstarts + tutorials](#)



Azure Synapse Analytics (formerly
SQL DW)

[Quickstarts + tutorials](#)



Azure Database for MariaDB

[Learn more](#)



Azure Database for MySQL

[Quickstarts + tutorials](#)



Azure Database for PostgreSQL

[Quickstarts + tutorials](#)



Azure Cosmos DB

[Quickstarts + tutorials](#)



SQL Server 2017 Enterprise Windows
Server 2016

[Learn more](#)



Azure Cache for Redis

[Quickstarts + tutorials](#)



Azure Database Migration Service

[Learn more](#)

3. On the **New Redis Cache** page, configure the settings for your new cache.

Setting	Choose a value	Description
Subscription	Drop down and select your subscription.	The subscription under which to create this new Azure Cache for Redis instance.
Resource group	Drop down and select a resource group, or select Create new and enter a new resource group name.	Name for the resource group in which to create your cache and other resources. By putting all your app resources in one resource group, you can easily manage or delete them together.
DNS name	Enter a unique name. The cache name must be a string between 1 and 63 characters that contain only numbers, letters, or hyphens. The name must start and end with a number or letter, and can't contain consecutive hyphens. Your cache instance's <i>host name</i> will be <DNS name>.redis.cache.windows.net.	
Location	Drop down and select a location.	Select a region near other services that will use your cache.
Cache type	Drop down and select a tier .	The tier determines the size, performance, and features that are available for the cache. For more information, see Azure Cache for Redis Overview .

4. Select the **Networking** tab or select the **Networking** button at the bottom of the page.
5. In the **Networking** tab, select your connectivity method.
6. Select the **Next: Advanced** tab or select the **Next: Advanced** button on the bottom of the page.
7. In the **Advanced** tab for a basic or standard cache instance, select the enable toggle if you want to enable a non-TLS port. You can also select which Redis version you would like use, either 4 or 6.

New Redis Cache ...

The screenshot shows the 'Advanced' tab selected in the top navigation bar. Under the 'Non-TLS port' section, there is a checkbox labeled 'Enable'. Below it, a section for 'Redis version' has two options: '4' and '6'. The option '6' is selected and highlighted with a red box.

Redis version	<input type="radio"/> 4
	<input checked="" type="radio"/> 6

8. In the **Advanced** tab for premium cache instance, configure the settings for non-TLS port, clustering, and data persistence. You can also select which Redis version you would like use, either 4 or 6.
9. Select the **Next: Tags** tab or select the **Next: Tags** button at the bottom of the page.
10. Optionally, in the **Tags** tab, enter the name and value if you wish to categorize the resource.
11. Select **Review + create**. You're taken to the Review + create tab where Azure validates your configuration.
12. After the green Validation passed message appears, select **Create**.

It takes a while for the cache to create. You can monitor progress on the Azure Cache for Redis **Overview** page. When **Status** shows as **Running**, the cache is ready to use.

Finally, update your application to use the new instances.

Clean up source resources

Once your new cache in the targeted region is running, delete the original instance.

Export and import data with an RDB file (Premium, Enterprise, Enterprise Flash)

Open-source Redis defines a standard mechanism for taking a snapshot of a cache's in-memory dataset and saving it to a file. This file, called RDB, can be read by another Redis cache. [Azure Cache for Redis Premium and Enterprise](#) supports importing data into a cache instance with RDB files. You can use an RDB file to transfer data from an existing cache to Azure Cache for Redis.

Important

RDB file format can change between Redis versions and might not maintain backward-compatibility. The Redis version of the cache you're exporting from should be the same or lower than the version of your new cache instance.

Prerequisites

- Both caches are [Premium tier or Enterprise tier](#) caches.
- The second cache is either the same cache size or a larger cache size than the original cache.
- The Redis version of the cache you're exporting from should be the same or lower than the version of your new cache instance.

Prepare

To move your cache instance to another region, you'll need to create [a second premium cache instance](#) or [a second enterprise cache instance](#) in the desired region.

Move

1. For more information on how to import and export data in Azure Cache for Redis. see [Import and Export data in Azure Cache for Redis](#).
2. Update your application to use the new cache instance.

Verify

You can monitor the progress of the import operation by following the notifications from the Azure portal, or by viewing the events in the [audit log](#).

Clean up source resources

Once your new cache in the targeted region is running, delete the original instance.

Dual-write to two caches (Basic, Standard, and Premium)

Rather than moving data directly between caches, you can use your application to write data to both an existing cache and a new one you're setting up. The application initially reads data from the existing cache initially. When the new cache has the necessary data, you switch the application to that cache and retire the old one. Let's say, for example, you use Redis as a session store and the application sessions are valid for seven days. After writing to the two caches for a week, you'll be certain the

new cache contains all non-expired session information. You can safely rely on it from that point onward without concern over data loss.

Prerequisites

- The second cache is either the same cache size or a larger cache size than the original cache.

Prepare

To move your cache instance to another region, you'll need to [create a second cache instance](#) in the desired region.

Move

General steps to implement this option are:

1. Modify application code to write to both the new and the original instances.
2. Continue reading data from the original instance until the new instance is sufficiently populated with data.
3. Update the application code to reading and writing from the new instance only.

Clean up source resources

Once your new cache in the targeted region is running, delete the original instance.

Migrate programmatically (All tiers)

You can create a custom migration process by programmatically reading data from an existing cache and writing them into Azure Cache for Redis. This [open-source tool](#) can be used to copy data from one Azure Cache for Redis instance to another instance in a different Azure Cache region. A [compiled version](#) is available as well. You may also find the source code to be a useful guide for writing your own migration tool.

Note

This tool isn't officially supported by Microsoft.

Prerequisites

- The second cache is either the same cache size or a larger cache size than the original cache.

Prepare

- Create a VM in the region where the existing cache is located. If your dataset is large, choose a relatively powerful VM to reduce copying time.
- To move your cache instance to another region, you'll need to [create a second cache instance](#) in the desired region.

Move

After creating a VM in the region where the existing cache is located and creating a new cache in the desired region, the general steps to implement this option are:

1. Flush data from the new cache to ensure that it's empty. This step is required because the copy tool itself doesn't overwrite any existing key in the target cache.

Important

Make sure to NOT flush from the source cache.

2. Use an application such as the open-source tool above to automate the copying of data from the source cache to the target. Remember that the copy process could take a while to complete depending on the size of your dataset.

Clean up source resources

Once your new cache in the targeted region is running, delete the original instance.

Move your Azure Cognitive Search service to another Azure region

- Article
- 01/31/2023
- 4 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare and move](#)
3. [Discard or clean up](#)
4. [Next steps](#)

Occasionally, customers ask about moving a search service to another region. Currently, there is no built-in mechanism or tooling to help with that task, but this article can help you understand the manual steps for recreating indexes and other objects on a new search service in a different region.

Note

In the Azure portal, all services have an **Export template** command. In the case of Azure Cognitive Search, this command produces a basic definition of a service (name, location, tier, replica, and partition count), but does not recognize the content of your service, nor does it carry over keys, roles, or logs. Although the command exists, we don't recommend using it for moving a search service.

Prerequisites

- Ensure that the services and features that your account uses are supported in the target region.
- For preview features, ensure that your subscription is approved for the target region.

Prepare and move

1. Identify dependencies and related services to understand the full impact of relocating a service, in case you need to move more than just Azure Cognitive Search.

Azure Storage is used for logging, creating a knowledge store, and is a commonly used external data source for AI enrichment and indexing. Cognitive Services is a dependency in AI enrichment. Both Cognitive Services and your search service are required to be in the same region if you are using AI enrichment.

2. Create an inventory of all objects on the service so that you know what to move: indexes, synonym maps, indexers, data sources, skillsets. If you

enabled logging, create and archive any reports you might need for a historical record.

3. Check pricing and availability in the new region to ensure availability of Azure Cognitive Search plus any related services in the new region. The majority of features are available in all regions, but some preview features have restricted availability.
4. Create a service in the new region and republish from source code any existing indexes, synonym maps, indexers, data sources, and skillsets. Remember that service names must be unique so you cannot reuse the existing name. Check each skillset to see if connections to Cognitive Services are still valid in terms of the same-region requirement. Also, if knowledge stores are created, check the connection strings for Azure Storage if you are using a different service.
5. Reload indexes and knowledge stores, if applicable. You'll either use application code to push JSON data into an index, or rerun indexers to pull documents in from external sources.
6. Enable logging, and if you are using them, re-create security roles.
7. Update client applications and test suites to use the new service name and API keys, and test all applications.

Discard or clean up

Delete the old service once the new service is fully tested and operational. Deleting the service automatically deletes all content associated with the service.

Manually move a container registry to another region

- Article
- 01/13/2023
- 8 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Considerations](#)
3. [Export template from source registry](#)
4. [Redeploy target registry in new region](#)

Show 4 more

You might need to move an Azure container registry from one Azure region to another. For example, you may run a development pipeline or host a new deployment target in a different region, and want to provide a nearby registry.

While [Azure Resource Mover](#) can't currently automate a move for an Azure container registry, you can manually move a container registry to a different region:

- Export registry settings to a Resource Manager template
- Use the template to deploy a registry in a different Azure region
- Import registry content from the source registry to the target registry

Note

If you need to distribute identical container images across multiple Azure regions, Azure Container Registry also supports [geo-replication](#). By geo-replicating a registry (Premium service tier required), you can serve multiple regions with identical image and tag names from a single registry.

Prerequisites

Azure CLI

- Use the Bash environment in [Azure Cloud Shell](#). For more information, see [Quickstart for Bash in Azure Cloud Shell](#).
-  [Launch Cloud Shell](#)
- If you prefer to run CLI reference commands locally, [install](#) the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see [How to run the Azure CLI in a Docker container](#).
 - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see [Sign in with the Azure CLI](#).
 - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
 - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).

Considerations

- Use steps in this article to move the registry to a different region in the same subscription. More configuration may be needed to move a registry to a different Azure subscription in the same Active Directory tenant.
- Exporting and using a Resource Manager template can help re-create many registry settings. You can edit the template to configure more settings, or update the target registry after creation.
- Currently, Azure Container Registry doesn't support a registry move to a different Active Directory tenant. This limitation applies to both registries encrypted with a [customer-managed key](#) and unencrypted registries.
- If you are unable to move a registry as outlined in this article, create a new registry, manually recreate settings, and [Import registry content in the target registry](#).
- You can find the steps to move resources of registry to a new resource group in the same subscription or move resources to a [new subscription](#).

Export template from source registry

Use the Azure portal, Azure CLI, Azure PowerShell, or other Azure tools to export a Resource Manager template. To use the Azure portal:

1. In the [Azure portal](#), navigate to your source registry.
2. In the menu, under **Automation**, select **Export template > Download**.

Redeploy target registry in new region

Modify template

Inspect the registry properties in the template JSON file you downloaded, and make necessary changes. At a minimum:

- Change the registry name's `defaultValue` to the desired name of the target registry
- Update the `location` to the desired Azure region for the target registry

JSONCopy

```
{  
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
    "contentVersion": "1.0.0.0",  
    "parameters": {  
        "registries_myregistry_name": {  
            "defaultValue": "myregistry",
```

```
        "type": "String"
    }
},
"variables": {},
"resources": [
{
    "type": "Microsoft.ContainerRegistry/registries",
    "apiVersion": "2020-11-01-preview",
    "name": "[parameters('myregistry_name')]",
    "location": "centralus",
    [...]
```

For more information, see [Use exported template from the Azure portal](#) and the [template reference](#).

Important

If you want to encrypt the target registry using a customer-managed key, make sure to update the template with settings for the required managed identity, key vault, and key. You can only enable the customer-managed key when you deploy the registry.

For more information, see [Encrypt registry using customer-managed key](./tutorial-enable-customer-managed-keys.md## Enable a customer-managed key by using a Resource Manager template).

Create resource group

Create a resource group for the target registry using the [az group create](#). The following example creates a resource group named *myResourceGroup* in the *eastus* location.

Azure CLICopy

```
az group create --name myResourceGroup --location eastus
```

Deploy target registry in new region

Use the [az deployment group create](#) command to deploy the target registry, using the template:

Azure CLICopy

```
az deployment group create --resource-group myResourceGroup \
    --template-file template.json --name mydeployment
```

Note

If you see errors during deployment, you might need to update certain configurations in the template file and retry the command.

Import registry content in target registry

After creating the registry in the target region, use the [az acr import](#) command, or the equivalent PowerShell command `Import-AzContainerImage`, to import images and other artifacts you want to preserve from the source registry to the target registry. For command examples, see [Import container images to a container registry](#).

- Use the Azure CLI commands [az acr repository list](#) and [az acr repository show-tags](#), or Azure PowerShell equivalents, to help enumerate the contents of your source registry.
- Run the import command for individual artifacts, or script it to run over a list of artifacts.

The following sample Azure CLI script enumerates the source repositories and tags and then imports the artifacts to a target registry in the same Azure subscription. Modify as needed to import specific repositories or tags. To import from a registry in a different subscription or tenant, see examples in [Import container images to a container registry](#).

Azure CLICopy

```
#!/bin/bash
# Modify registry names for your environment
SOURCE_REG=myregistry
TARGET_REG=targetregistry

# Get list of source repositories
REPO_LIST=$(az acr repository list \
    --name $SOURCE_REG --output tsv)

# Enumerate tags and import to target registry
for repo in $REPO_LIST; do
    TAGS_LIST=$(az acr repository show-tags --name $SOURCE_REG --repository $repo \
    --output tsv);
    for tag in $TAGS_LIST; do
        echo "Importing $repo:$tag";
        az acr import --name $TARGET_REG --source
$SOURCE_REG.azurecr.io/$repo":"$tag;
    done
done
```

Verify target registry

Confirm the following information in your target registry:

- Registry settings such as the registry name, service tier, public access, and replications
- Repositories and tags for content that you want to preserve.

Additional configuration

- If needed, manually configure settings in the target registry such as private endpoints, IP access rules, and managed identities.
- Update development and deployment systems to use the target registry instead of the source registry.
- Update any client firewall rules to allow access to the target registry.

Delete original registry

After you have successfully deployed the target registry, migrated content, and verified registry settings, you may delete the source registry.

Introduction to Container registries in Azure

- Article
- 10/12/2022
- 22 contributors

Feedback

In this article

1. [Use cases](#)
2. [Key features](#)
3. [Next steps](#)

Azure Container Registry is a managed registry service based on the open-source Docker Registry 2.0. Create and maintain Azure container registries to store and manage your container images and related artifacts.

Use Azure container registries with your existing container development and deployment pipelines, or use Azure Container Registry Tasks to build container images in Azure. Build on demand, or fully automate builds with triggers such as source code commits and base image updates.

Learn more about Docker and Registry concepts, see the [Docker overview](#), and [About registries, repositories, and images](#).

Use cases

Pull images from an Azure container registry to various deployment targets:

- **Scalable orchestration systems** that manage containerized applications across clusters of hosts, including [Kubernetes](#), [DC/OS](#), and [Docker Swarm](#).
- **Azure services** that support building and running applications at scale, including [Azure Kubernetes Service \(AKS\)](#), [App Service](#), [Batch](#), [Service Fabric](#), and others.

Developers can also push to a container registry as part of a container development workflow. For example, target a container registry from a continuous integration and delivery tool such as [Azure Pipelines](#) or [Jenkins](#).

Configure ACR Tasks to automatically rebuild application images when their base images are updated, or automate image builds when your team commits code to a Git repository. Create multi-step tasks to automate building, testing, and patching multiple container images in parallel in the cloud.

Azure provides tooling including the Azure CLI, the Azure portal, and API support to manage your Azure container registries. Optionally install the [Docker Extension for Visual Studio Code](#) and the [Azure Account](#) extension to work with your Azure container registries. Pull and push images to an Azure container registry, or run ACR Tasks, all within Visual Studio Code.

Key features

- **Registry service tiers** - Create one or more container registries in your Azure subscription. Registries are available in three tiers: [Basic](#), [Standard](#), and [Premium](#), each of which supports webhook integration, registry authentication with Azure Active Directory, and delete functionality. Take advantage of local, network-close storage of your container images by creating a registry in the same Azure location as your deployments. Use the [geo-replication](#) feature of Premium registries for advanced replication and container image distribution scenarios.
- **Security and access** - You log in to a registry using the Azure CLI or the standard `docker login` command. Azure Container Registry transfers container images over HTTPS, and supports TLS to secure client connections.

Important

Starting January 13, 2020, Azure Container Registry will require all secure connections from servers and applications to use TLS 1.2. Enable TLS 1.2

by using any recent docker client (version 18.03.0 or later). Support for TLS 1.0 and 1.1 will be retired.

You [control access](#) to a container registry using an Azure identity, an Azure Active Directory-backed [service principal](#), or a provided admin account. Use Azure role-based access control (Azure RBAC) to assign users or systems fine-grained permissions to a registry.

Security features of the Premium service tier include [content trust](#) for image tag signing, and [firewalls and virtual networks \(preview\)](#) to restrict access to the registry. Microsoft Defender for Cloud optionally integrates with Azure Container Registry to [scan images](#) whenever an image is pushed to a registry.

- **Supported images and artifacts** - Grouped in a repository, each image is a read-only snapshot of a Docker-compatible container. Azure container registries can include both Windows and Linux images. You control image names for all your container deployments. Use standard [Docker commands](#) to push images into a repository, or pull an image from a repository. In addition to Docker container images, Azure Container Registry stores [related content formats](#) such as [Helm charts](#) and images built to the [Open Container Initiative \(OCI\) Image Format Specification](#).
- **Automated image builds** - Use [Azure Container Registry Tasks](#) (ACR Tasks) to streamline building, testing, pushing, and deploying images in Azure. For example, use ACR Tasks to extend your development inner-loop to the cloud by offloading `docker build` operations to Azure. Configure build tasks to automate your container OS and framework patching pipeline, and build images automatically when your team commits code to source control.

[Multi-step tasks](#) provide step-based task definition and execution for building, testing, and patching container images in the cloud. Task steps define individual container image build and push operations. They can also define the execution of one or more containers, with each step using the container as its execution environment.

What is a connected registry?

- Article
- 10/26/2022
- 4 contributors

Feedback

In this article

1. [Available regions](#)
2. [Scenarios](#)
3. [How does the connected registry work?](#)
4. [Client access](#)

Show 2 more

In this article, you learn about the *connected registry* feature of [Azure Container Registry](#). A connected registry is an on-premises or remote replica that synchronizes container images and other OCI artifacts with your cloud-based Azure container registry. Use a connected registry to help speed up access to registry artifacts on-premises and to build advanced scenarios, for example using [nested IoT Edge](#).

Note

The connected registry is a preview feature of the **Premium** container registry service tier, and subject to [limitations](#). For information about registry service tiers and limits, see [Azure Container Registry service tiers](#).

Available regions

- Canada Central
- East Asia
- East US
- North Europe
- Norway East
- Southeast Asia
- West Central US
- West Europe

Scenarios

A cloud-based Azure container registry provides [features](#) including geo-replication, integrated security, Azure-managed storage, and integration with Azure development and deployment pipelines. At the same time, customers are extending their cloud investments to their on-premises and field solutions.

To run with the required performance and reliability in on-premises or remote environments, container workloads need container images and related artifacts to be available nearby. The connected registry provides a performant, on-premises registry solution that regularly synchronizes content with a cloud-based Azure container registry.

Scenarios for a connected registry include:

- Connected factories
- Point-of-sale retail locations
- Shipping, oil-drilling, mining, and other occasionally connected environments

How does the connected registry work?

The following image shows a typical deployment model for the connected registry.

Deployment

Each connected registry is a resource you manage using a cloud-based Azure container registry. The top parent in the connected registry hierarchy is an Azure container registry in an Azure cloud or in a private deployment of [Azure Stack Hub](#).

Use Azure tools to install the connected registry on a server or device on your premises, or an environment that supports container workloads on-premises such as [Azure IoT Edge](#).

The connected registry's *activation status* indicates whether it's deployed on-premises.

- **Active** - The connected registry is currently deployed on-premises. It can't be deployed again until it is deactivated.
- **Inactive** - The connected registry is not deployed on-premises. It can be deployed at this time.

Content synchronization

The connected registry regularly accesses the cloud registry to synchronize container images and OCI artifacts.

It can also be configured to synchronize a subset of the repositories from the cloud registry or to synchronize only during certain intervals to reduce traffic between the cloud and the premises.

Modes

A connected registry can work in one of two modes: *ReadWrite* or *ReadOnly*

- **ReadWrite mode** - The default mode allows clients to pull and push artifacts (read and write) to the connected registry. Artifacts that are pushed to the connected registry will be synchronized with the cloud registry.

The ReadWrite mode is useful when a local development environment is in place. The images are pushed to the local connected registry and from there synchronized to the cloud.

- **ReadOnly mode** - When the connected registry is in ReadOnly mode, clients may only pull (read) artifacts. This configuration is used for nested IoT Edge scenarios, or other scenarios where clients need to pull a container image to operate.

Registry hierarchy

Each connected registry must be connected to a parent. The top parent is the cloud registry. For hierarchical scenarios such as [nested IoT Edge](#), you can nest connected registries in either mode. The parent connected to the cloud registry can operate in either mode.

Child registries must be compatible with their parent capabilities. Thus, both ReadWrite and ReadOnly mode connected registries can be children of a connected registry operating in ReadWrite mode, but only a ReadOnly mode registry can be a child of a connected registry operating in ReadOnly mode.

Client access

On-premises clients use standard tools such as the Docker CLI to push or pull content from a connected registry. To manage client access, you create Azure container registry [tokens](#) for access to each connected registry. You can scope the client tokens for pull or push access to one or more repositories in the registry.

Each connected registry also needs to regularly communicate with its parent registry. For this purpose, the registry is issued a synchronization token (*sync token*) by the cloud registry. This token is used to authenticate with its parent registry for synchronization and management operations.

For more information, see [Manage access to a connected registry](#).

Limitations

- Number of tokens and scope maps is [limited](#) to 20,000 each for a single container registry. This indirectly limits the number of connected registries for a cloud registry, because every connected registry needs a sync and client token.
- Number of repository permissions in a scope map is limited to 500.
- Number of clients for the connected registry is currently limited to 20.
- [Image locking](#) through repository/manifest/tag metadata is not currently supported for connected registries.
- [Repository delete](#) is not supported on the connected registry using `ReadOnly` mode.
- [Resource logs](#) for connected registries are currently not supported.
- Connected registry is coupled with the registry's home region data endpoint. Automatic migration for [geo-replication](#) is not supported.
- Deletion of a connected registry needs manual removal of the containers on-premises as well as removal of the respective scope map or tokens in the cloud.
- Connected registry sync limitations are as follows:
 - For continuous sync:
 - `minMessageTt1` is 1 day
 - `maxMessageTt1` is 90 days
 - For occasionally connected scenarios, where you want to specify sync window:
 - `minSyncWindow` is 1 hr
 - `maxSyncWindow` is 7 days

Understand access to a connected registry

- Article
- 10/12/2022
- 5 contributors

Feedback

In this article

1. [Client tokens](#)
2. [Sync token](#)
3. [Registry endpoints](#)
4. [Next steps](#)

To access and manage a [connected registry](#), currently only ACR [token-based authentication](#) is supported. As shown in the following image, two different types of tokens are used by each connected registry:

- **Client tokens** - One or more tokens that on-premises clients use to authenticate with a connected registry and push or pull images and artifacts to or from it.
- **Sync token** - A token used by each connected registry to access its parent and synchronize content.

Important

Store token passwords for each connected registry in a safe location. After they are created, token passwords can't be retrieved. You can regenerate token passwords at any time.

Client tokens

To manage client access to a connected registry, you create tokens scoped for actions on one or more repositories. After creating a token, configure the connected registry to accept the token by using the [az acr connected-registry update](#) command. A client can then use the token credentials to access a connected registry endpoint - for example, to use Docker CLI commands to pull or push images to the connected registry.

Your options for configuring client token actions depend on whether the connected registry allows both push and pull operations or functions as a pull-only mirror.

- A connected registry in the default [ReadWrite mode](#) allows both pull and push operations, so you can create a token that allows actions to both *read* and *write* repository content in that registry.
- For a connected registry in [ReadOnly mode](#), client tokens can only allow actions to *read* repository content.

Manage client tokens

Update client tokens, passwords, or scope maps as needed by using [az acr token](#) and [az acr scope-map](#) commands. Client token updates are propagated automatically to the connected registries that accept the token.

Sync token

Each connected registry uses a sync token to authenticate with its immediate parent - which could be another connected registry or the cloud registry. The connected

registry automatically uses this token when synchronizing content with the parent or performing other updates.

- The sync token and passwords are generated automatically when you create the connected registry resource. Run the [az acr connected-registry install renew-credentials](#) command to regenerate the passwords.
- Include sync token credentials in the configuration used to deploy the connected registry on-premises.
- By default, the sync token is granted permission to synchronize selected repositories with its parent. You must provide an existing sync token or one or more repositories to sync when you create the connected registry resource.
- It also has permissions to read and write synchronization messages on a gateway used to communicate with the connected registry's parent. These messages control the synchronization schedule and manage other updates between the connected registry and its parent.

Manage sync token

Update sync tokens, passwords, or scope maps as needed by using [az acr token](#) and [az acr scope-map](#) commands. Sync token updates are propagated automatically to the connected registry. Follow the standard practices of rotating passwords when updating the sync token.

Note

The sync token cannot be deleted until the connected registry associated with the token is deleted. You can disable a connected registry by setting the status of the sync token to `disabled`.

Registry endpoints

Token credentials for connected registries are scoped to access specific registry endpoints:

- A client token accesses the connected registry's endpoint. The connected registry endpoint is the login server URI, which is typically the IP address of the server or device that hosts it.
- A sync token accesses the endpoint of the parent registry, which is either another connected registry endpoint or the cloud registry itself. When scoped to access the cloud registry, the sync token needs to reach two registry endpoints:
 - The fully qualified login server name, for example, `contoso.azurecr.io`. This endpoint is used for authentication.

- A fully qualified regional [data endpoint](#) for the cloud registry, for example, contoso.westus2.data.azurecr.io. This endpoint is used to exchange messages with the connected registry for synchronization purposes.

Pull images from a connected registry on IoT Edge device

- Article
- 01/13/2023
- 6 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Create a scope map](#)
3. [Create a client token](#)
4. [Update the connected registry with the client token](#)

Show 2 more

To pull images from a [connected registry](#), configure a [client token](#) and pass the token credentials to access registry content.

Prerequisites

- Use the Bash environment in [Azure Cloud Shell](#). For more information, see [Quickstart for Bash in Azure Cloud Shell](#).
A blue rectangular button with a white 'A' icon on the left and the text 'Launch Cloud Shell' in white on the right.
- If you prefer to run CLI reference commands locally, [install](#) the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see [How to run the Azure CLI in a Docker container](#).
 - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see [Sign in with the Azure CLI](#).
 - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).

- Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).
- Connected registry resource in Azure. For deployment steps, see [Quickstart: Create a connected registry using the Azure CLI](#).
- Connected registry instance deployed on an IoT Edge device. For deployment steps, see [Quickstart: Deploy a connected registry to an IoT Edge device](#) or [Tutorial: Deploy a connected registry to nested IoT Edge devices](#). In the commands in this article, the connected registry name is stored in the environment variable `$CONNECTED_REGISTRY_RW`.

Create a scope map

Use the [az acr scope-map create](#) command to create a scope map for read access to the `hello-world` repository:

Azure CLICopy

```
# Use the REGISTRY_NAME variable in the following Azure CLI commands to identify
# the registry
REGISTRY_NAME=<container-registry-name>

az acr scope-map create \
--name hello-world-scopemap \
--registry $REGISTRY_NAME \
--repository hello-world content/read \
--description "Scope map for the connected registry."
```

Create a client token

Use the [az acr token create](#) command to create a client token and associate it with the newly created scope map:

Azure CLICopy

```
az acr token create \
--name myconnectedregistry-client-token \
--registry $REGISTRY_NAME \
--scope-map hello-world-scopemap
```

The command will return details about the newly generated token including passwords.

Important

Make sure that you save the generated passwords. Those are one-time passwords and cannot be retrieved. You can generate new passwords using the [az acr token credential generate](#) command.

Update the connected registry with the client token

Use the az acr connected-registry update command to update the connected registry with the newly created client token.

Azure CLICopy

```
az acr connected-registry update \
--name $CONNECTED_REGISTRY_RW \
--registry $REGISTRY_NAME \
--add-client-token myconnectedregistry-client-token
```

Pull an image from the connected registry

From a machine with access to the IoT Edge device, use the following example command to sign into the connected registry, using the client token credentials. For best practices to manage login credentials, see the [docker login](#) command reference.

Caution

If you set up your connected registry as an insecure registry, update the insecure registries list in the Docker daemon configuration to include the IP address (or FQDN) and port of your connected registry on the IoT Edge device. This configuration should only be used for testing purposes. For more information, see [Test an insecure registry](#).

Copy

```
docker login --username myconnectedregistry-client-token \
--password <token_password> <IP_address_or_FQDN_of_connected_registry>:<port>
```

For IoT Edge scenarios, be sure to include the port used to reach the connected registry on the device. Example:

Copy

```
docker login --username myconnectedregistry-client-token \
--password xxxxxxxxxxxx 192.0.2.13:8000
```

Then, use the following command to pull the `hello-world` image:

Copy

```
docker pull <IP_address_or_FQDN_of_connected_registry>:<port>/hello-world
```

Move an Azure Cosmos DB account to another region

- Article
- 02/24/2023
- 5 contributors

Feedback

In this article

1. [Move data from one region to another](#)
2. [Migrate Azure Cosmos DB account metadata](#)
3. [Next steps](#)

APPLIES TO: NoSQL MongoDB Cassandra Gremlin Table

This article describes how to either:

- Move a region where data is replicated in Azure Cosmos DB.
- Migrate account (Azure Resource Manager) metadata and data from one region to another.

Move data from one region to another

Azure Cosmos DB supports data replication natively, so moving data from one region to another is simple. You can accomplish it by using the Azure portal, Azure PowerShell, or the Azure CLI. It involves the following steps:

1. Add a new region to the account.

To add a new region to an Azure Cosmos DB account, see [Add/remove regions to an Azure Cosmos DB account](#).

2. Perform a manual failover to the new region.

When the region that's being removed is currently the write region for the account, you'll need to start a failover to the new region added in the previous step. This is a zero-downtime operation. If you're moving a read region in a multiple-region account, you can skip this step.

To start a failover, see [Perform manual failover on an Azure Cosmos DB account](#).

3. Remove the original region.

To remove a region from an Azure Cosmos DB account, see [Add/remove regions from your Azure Cosmos DB account](#).

Note

If you perform a failover operation or add/remove a new region while an [asynchronous throughput scaling operation](#) is in progress, the throughput scale-up operation will be paused. It will resume automatically when the failover or add/remove region operation is complete.

Migrate Azure Cosmos DB account metadata

Azure Cosmos DB does not natively support migrating account metadata from one region to another. To migrate both the account metadata and customer data from one region to another, you must create a new account in the desired region and then copy the data manually.

Important

It is not necessary to migrate the account metadata if the data is stored or moved to a different region. The region in which the account metadata resides has no impact on the performance, security or any other operational aspects of your Azure Cosmos DB account.

A near-zero-downtime migration for the API for NoSQL requires the use of the [change feed](#) or a tool that uses it. If you're migrating from the API for MongoDB, Cassandra, or another API, or to learn more about options for migrating data between accounts, see [Options to migrate your on-premises or cloud data to Azure Cosmos DB](#).

The following steps demonstrate how to migrate an Azure Cosmos DB account for the API for NoSQL and its data from one region to another:

1. Create a new Azure Cosmos DB account in the desired region.

To create a new account via the Azure portal, PowerShell, or the Azure CLI, see [Create an Azure Cosmos DB account](#).

2. Create a new database and container.

To create a new database and container, see [Create an Azure Cosmos DB container](#).

3. Migrate data by using the Azure Cosmos DB Spark Connector live migration sample.

To migrate data with near zero downtime, see [Live Migrate Azure Cosmos DB SQL API Containers data with Spark Connector](#).

4. Update the application connection string.

With the Live Data Migration sample still running, update the connection information in the new deployment of your application. You can retrieve the endpoints and keys for your application from the Azure portal.

5. Redirect requests to the new application.

After the new application is connected to Azure Cosmos DB, you can redirect client requests to your new deployment.

6. Delete any resources that you no longer need.

With requests now fully redirected to the new instance, you can delete the old Azure Cosmos DB account and stop the Live Data Migrator sample.

Move an Azure Database for MariaDB server to another region by using the Azure portal

- Article
- 06/26/2022
- 5 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare to move](#)
3. [Move](#)
4. [Clean up source server](#)
5. [Next steps](#)

There are various scenarios for moving an existing Azure Database for MariaDB server from one region to another. For example, you might want to move a production server to another region as part of your disaster recovery planning.

You can use an Azure Database for MariaDB [cross-region read replica](#) to complete the move to another region. To do so, first create a read replica in the target region. Next, stop replication to the read replica server to make it a standalone server that accepts both read and write traffic.

Note

This article focuses on moving your server to a different region. If you want to move your server to a different resource group or subscription, refer to the [move](#) article.

Prerequisites

- The read replica feature is only available for Azure Database for MariaDB servers in the General Purpose or Memory Optimized pricing tiers. Ensure the source server is in one of these pricing tiers.
- Make sure that your Azure Database for MariaDB source server is in the Azure region that you want to move from.

Prepare to move

To create a cross-region read replica server in the target region using the Azure portal, use the following steps:

1. Sign into the [Azure portal](#).
2. Select the existing Azure Database for MariaDB server that you want to use as the source server. This action opens the **Overview** page.
3. Select **Replication** from the menu, under **SETTINGS**.
4. Select **Add Replica**.
5. Enter a name for the replica server.
6. Select the location for the replica server. The default location is the same as the source server's. Verify that you've selected the target location where you want the replica to be deployed.
7. Select **OK** to confirm creation of the replica. During replica creation, data is copied from the source server to the replica. Create time may last several minutes or more, in proportion to the size of the source server.

Note

When you create a replica, it doesn't inherit the VNet service endpoints of the source server. These rules must be set up independently for the replica.

Move

Important

The standalone server can't be made into a replica again. Before you stop replication on a read replica, ensure the replica has all the data that you require.

Stopping replication to the replica server, causes it to become a standalone server. To stop replication to the replica from the Azure portal, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for MariaDB source server.
2. Select **Replication** from the menu, under **SETTINGS**.
3. Select the replica server.
4. Select **Stop replication**.
5. Confirm you want to stop replication by selecting **OK**.

Clean up source server

You may want to delete the source Azure Database for MariaDB server. To do so, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for MariaDB source server.
2. In the **Overview** window, select **Delete**.
3. Type in the name of the source server to confirm you want to delete.
4. Select **Delete**.

Move an Azure Database for MySQL server to another region by using the Azure portal

- Article
- 09/29/2022
- 3 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare to move](#)
3. [Move](#)
4. [Clean up source server](#)

5. [Next steps](#)

APPLIES TO:  Azure Database for MySQL - Single Server

Important

Azure Database for MySQL - Single Server is on the retirement path. We strongly recommend for you to upgrade to Azure Database for MySQL - Flexible Server. For more information about migrating to Azure Database for MySQL - Flexible Server, see [What's happening to Azure Database for MySQL Single Server?](#)

There are various scenarios for moving an existing Azure Database for MySQL server from one region to another. For example, you might want to move a production server to another region as part of your disaster recovery planning.

You can use an Azure Database for MySQL [cross-region read replica](#) to complete the move to another region. To do so, first create a read replica in the target region. Next, stop replication to the read replica server to make it a standalone server that accepts both read and write traffic.

Note

This article focuses on moving your server to a different region. If you want to move your server to a different resource group or subscription, refer to the [move](#) article.

Prerequisites

- The read replica feature is only available for Azure Database for MySQL servers in the General Purpose or Memory Optimized pricing tiers. Ensure the source server is in one of these pricing tiers.
- Make sure that your Azure Database for MySQL source server is in the Azure region that you want to move from.

Prepare to move

To create a cross-region read replica server in the target region using the Azure portal, use the following steps:

1. Sign into the [Azure portal](#).
2. Select the existing Azure Database for MySQL server that you want to use as the source server. This action opens the **Overview** page.
3. Select **Replication** from the menu, under **SETTINGS**.
4. Select **Add Replica**.
5. Enter a name for the replica server.

6. Select the location for the replica server. The default location is the same as the source server's. Verify that you've selected the target location where you want the replica to be deployed.
7. Select **OK** to confirm creation of the replica. During replica creation, data is copied from the source server to the replica. Create time may last several minutes or more, in proportion to the size of the source server.

Note

When you create a replica, it doesn't inherit the VNet service endpoints of the source server. These rules must be set up independently for the replica.

Move

Important

The standalone server can't be made into a replica again. Before you stop replication on a read replica, ensure the replica has all the data that you require.

Stopping replication to the replica server, causes it to become a standalone server. To stop replication to the replica from the Azure portal, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for MySQL source server.
2. Select **Replication** from the menu, under **SETTINGS**.
3. Select the replica server.
4. Select **Stop replication**.
5. Confirm you want to stop replication by clicking **OK**.

Clean up source server

You may want to delete the source Azure Database for MySQL server. To do so, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for MySQL source server.
2. In the **Overview** window, select **Delete**.
3. Type in the name of the source server to confirm you want to delete.
4. Select **Delete**.

Move an Azure Database for MySQL server to another region by using the Azure portal

- Article
- 09/29/2022
- 3 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare to move](#)
3. [Move](#)
4. [Clean up source server](#)
5. [Next steps](#)

APPLIES TO:  Azure Database for MySQL - Single Server

Important

Azure Database for MySQL - Single Server is on the retirement path. We strongly recommend for you to upgrade to Azure Database for MySQL - Flexible Server. For more information about migrating to Azure Database for MySQL - Flexible Server, see [What's happening to Azure Database for MySQL Single Server?](#)

There are various scenarios for moving an existing Azure Database for MySQL server from one region to another. For example, you might want to move a production server to another region as part of your disaster recovery planning.

You can use an Azure Database for MySQL [cross-region read replica](#) to complete the move to another region. To do so, first create a read replica in the target region. Next, stop replication to the read replica server to make it a standalone server that accepts both read and write traffic.

Note

This article focuses on moving your server to a different region. If you want to move your server to a different resource group or subscription, refer to the [move](#) article.

Prerequisites

- The read replica feature is only available for Azure Database for MySQL servers in the General Purpose or Memory Optimized pricing tiers.
Ensure the source server is in one of these pricing tiers.
- Make sure that your Azure Database for MySQL source server is in the Azure region that you want to move from.

Prepare to move

To create a cross-region read replica server in the target region using the Azure portal, use the following steps:

1. Sign into the [Azure portal](#).
2. Select the existing Azure Database for MySQL server that you want to use as the source server. This action opens the **Overview** page.
3. Select **Replication** from the menu, under **SETTINGS**.
4. Select **Add Replica**.
5. Enter a name for the replica server.
6. Select the location for the replica server. The default location is the same as the source server's. Verify that you've selected the target location where you want the replica to be deployed.
7. Select **OK** to confirm creation of the replica. During replica creation, data is copied from the source server to the replica. Create time may last several minutes or more, in proportion to the size of the source server.

Note

When you create a replica, it doesn't inherit the VNet service endpoints of the source server. These rules must be set up independently for the replica.

Move

Important

The standalone server can't be made into a replica again. Before you stop replication on a read replica, ensure the replica has all the data that you require.

Stopping replication to the replica server, causes it to become a standalone server. To stop replication to the replica from the Azure portal, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for MySQL source server.
2. Select **Replication** from the menu, under **SETTINGS**.
3. Select the replica server.

4. Select **Stop replication**.
5. Confirm you want to stop replication by clicking **OK**.

Clean up source server

You may want to delete the source Azure Database for MySQL server. To do so, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for MySQL source server.
2. In the **Overview** window, select **Delete**.
3. Type in the name of the source server to confirm you want to delete.
4. Select **Delete**.

Move an Azure Database for Azure Database for PostgreSQL - Single Server to another region by using the Azure portal

- Article
- 03/29/2023
- 3 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare to move](#)
3. [Move](#)
4. [Clean up source server](#)
5. [Next steps](#)

APPLIES TO:  Azure Database for PostgreSQL - Single Server

Important

Azure Database for PostgreSQL - Single Server is on the retirement path. We strongly recommend for you to upgrade to Azure Database for PostgreSQL - Flexible Server. For more information about migrating to Azure Database for PostgreSQL - Flexible Server, see [What's happening to Azure Database for PostgreSQL Single Server?](#)

There are various scenarios for moving an existing Azure Database for PostgreSQL server from one region to another. For example, you might want to move a production server to another region as part of your disaster recovery planning.

You can use an Azure Database for PostgreSQL [cross-region read replica](#) to complete the move to another region. To do so, first create a read replica in the target region. Next, stop replication to the read replica server to make it a standalone server that accepts both read and write traffic.

Note

This article focuses on moving your server to a different region. If you want to move your server to a different resource group or subscription, refer to the [move](#) article.

Prerequisites

- The cross-region read replica feature is only available for Azure Database for PostgreSQL - Single Server in the General Purpose or Memory Optimized pricing tiers. Ensure the source server is in one of these pricing tiers.
- Make sure that your Azure Database for PostgreSQL source server is in the Azure region that you want to move from.

Prepare to move

To prepare the source server for replication using the Azure portal, use the following steps:

1. Sign into the [Azure portal](#).
2. Select the existing Azure Database for PostgreSQL server that you want to use as the source server. This action opens the **Overview** page.
3. From the server's menu, select **Replication**. If Azure replication support is set to at least **Replica**, you can create read replicas.
4. If Azure replication support is not set to at least **Replica**, set it. Select **Save**.
5. Restart the server to apply the change by selecting **Yes**.
6. You will receive two Azure portal notifications once the operation is complete. There is one notification for updating the server parameter. There is another notification for the server restart that follows immediately.
7. Refresh the Azure portal page to update the Replication toolbar. You can now create read replicas for this server.

To create a cross-region read replica server in the target region using the Azure portal, use the following steps:

1. Select the existing Azure Database for PostgreSQL server that you want to use as the source server.
2. Select **Replication** from the menu, under **SETTINGS**.
3. Select **Add Replica**.
4. Enter a name for the replica server.
5. Select the location for the replica server. The default location is the same as the primary server's. Verify that you've selected the target location where you want the replica to be deployed.
6. Select **OK** to confirm creation of the replica. During replica creation, data is copied from the source server to the replica. Create time may last several minutes or more, in proportion to the size of the source server.

Note

When you create a replica, it doesn't inherit the firewall rules and VNet service endpoints of the primary server. These rules must be set up independently for the replica.

Move

Important

The standalone server can't be made into a replica again. Before you stop replication on a read replica, ensure the replica has all the data that you require.

To stop replication to the replica from the Azure portal, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for PostgreSQL source server.
2. Select **Replication** from the menu, under **SETTINGS**.
3. Select the replica server.
4. Select **Stop replication**.
5. Confirm you want to stop replication by selecting **OK**.

Clean up source server

You may want to delete the source Azure Database for PostgreSQL server. To do so, use the following steps:

1. Once the replica has been created, locate and select your Azure Database for PostgreSQL source server.
2. In the **Overview** window, select **Delete**.
3. Type in the name of the source server to confirm you want to delete.
4. Select **Delete**.

Move an Azure Event Hubs namespace to another region

- Article
- 06/17/2021
- 3 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare](#)
3. [Move](#)
4. [Discard or clean up](#)
5. [Next steps](#)

This article shows you how to export an Azure Resource Manager template for an existing Event Hubs namespace and then use the template to create a namespace with same configuration settings in another region. However, this process doesn't move events that aren't processed yet. You need to process the events from the original namespace before deleting it.

If you have other resources in the Azure resource group that contains the Event Hubs namespace, you may want to export the template at the resource group level so that all related resources can be moved to the new region in one step. The steps in this article show you how to export a **namespace** to the template. The steps for exporting a **resource group** to the template are similar.

Prerequisites

- Ensure that the services and features that your account uses are supported in the target region.
- If you have **capture feature** enabled for event hubs in the namespace, move [Azure Storage or Azure Data Lake Store Gen 2](#) or [Azure Data Lake Store Gen 1](#) accounts before moving the Event Hubs namespace. You can also move the resource group that contains both Storage and Event Hubs namespaces to the other region by following steps similar to the ones described in this article.
- If the Event Hubs namespace is in an **Event Hubs cluster**, [move the dedicated cluster](#) to the **target region** before you go through steps in this article. You can also use the [quickstart template on GitHub](#) to create an Event Hubs cluster. In the template, remove the namespace portion of the JSON to create only the cluster.

Prepare

To get started, export a Resource Manager template. This template contains settings that describe your Event Hubs namespace.

1. Sign in to the [Azure portal](#).
2. Select **All resources** and then select your Event Hubs namespace.
3. On the **Event Hubs Namespace** page, select **Export template** under **Automation** in the left menu.
4. Choose **Download** in the **Export template** page.

5. Locate the .zip file that you downloaded from the portal, and unzip that file to a folder of your choice.

This zip file contains the .json files that include the template and scripts to deploy the template.

Move

Deploy the template to create an Event Hubs namespace in the target region.

1. In the Azure portal, select **Create a resource**.
2. In **Search the Marketplace**, type **template deployment**, and select **Template deployment (deploy using custom templates)**.
3. Select **Build your own template in the editor**.
4. Select **Load file**, and then follow the instructions to load the **template.json** file that you downloaded in the last section.
5. Update the value of the **location** property to point to the new region. To obtain location codes, see [Azure locations](#). The code for a region is the region name with no spaces, for example, West US is equal to westus.
6. Select **Save** to save the template.
7. On the **Custom deployment** page, follow these steps:
 - a. Select an Azure **subscription**.
 - b. Select an existing **resource group** or create one. If the source namespace was in an Event Hubs cluster, select the resource group that contains cluster in the target region.
 - c. Select the target **location** or region. If you selected an existing resource group, this setting is read-only.
 - d. In the **SETTINGS** section, do the following steps:
 - i. Enter the new **namespace name**.

ii.If your source namespace was in an **Event Hubs cluster**, enter names of **resource group** and **Event Hubs cluster** as part of **external ID**.

Copy

```
/subscriptions/<AZURE SUBSCRIPTION ID>/resourceGroups/<CLUSTER'S RESOURCE GROUP>/providers/Microsoft.EventHub/clusters/<CLUSTER NAME>
```

iii.If event hub in your namespace uses a Storage account for capturing events, specify the resource group name and the storage account for `StorageAccounts_<original storage account name>_external` field.

Copy

```
/subscriptions/0000000000-0000-0000-0000-0000000000/resourceGroups/<STORAGE'S RESOURCE GROUP>/providers/Microsoft.Storage/storageAccounts/<STORAGE ACCOUNT NAME>
```

- e. Select **Review + create** at the bottom of the page.
- f. On the **Review + create** page, review settings, and then select **Create**.

Discard or clean up

After the deployment, if you want to start over, you can delete the **target Event Hubs namespace**, and repeat the steps described in the [Prepare](#) and [Move](#) sections of this article.

To commit the changes and complete the move of an Event Hubs namespace, delete the **Event Hubs namespace** in the original region. Make sure that you processed all the events in the namespace before deleting the namespace.

To delete an Event Hubs namespace (source or target) by using the Azure portal:

1. In the search window at the top of Azure portal, type **Event Hubs**, and select **Event Hubs** from search results. You see the Event Hubs namespaces in a list.
2. Select the target namespace to delete, and select **Delete** from the toolbar.

3. On the **Delete Namespace** page, confirm the deletion by typing the **namespace name**, and then select **Delete**.

Move your function app between regions in Azure Functions

- Article
- 02/01/2023
- 2 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare](#)
3. [Move](#)
4. [Clean up source resources](#)
5. [Next steps](#)

This article describes how to move Azure Functions resources to a different Azure region. You might move your resources to another region for one of the following reasons:

- Take advantage of a new Azure region
- Deploy features or services that are available only in specific regions
- Meet internal policy and governance requirements
- Respond to capacity planning requirements

Azure Functions resources are region-specific and can't be moved across regions. You must create a copy of your existing function app resources in the target region, then redeploy your functions code over to the new app.

If minimal downtime is a requirement, consider running your function app in both regions to implement a disaster recovery architecture:

- [Azure Functions geo-disaster recovery](#)
- [Disaster recovery and geo-distribution in Azure Durable Functions](#)

Prerequisites

- Make sure that the target region supports Azure Functions and any related service whose resources you want to move
- Have access to the original source code for the functions you're migrating

Prepare

Identify all the function app resources used on the source region, which may include the following:

- Function app
- [Hosting plan](#)
- [Deployment slots](#)
- [Custom domains purchased in Azure](#)
- [TLS/SSL certificates and settings](#)
- [Configured networking options](#)
- [Managed identities](#)
- [Configured application settings](#) - users with the enough access can copy all the source application settings by using the Advanced Edit feature in the portal
- [Scaling configurations](#)

Your functions may connect to other resources by using triggers or bindings. For information on how to move those resources across regions, see the documentation for the respective services.

You should be able to also [export a template from existing resources](#).

Move

Deploy the function app to the target region and review the configured resources.

Redeploy function app

If you have access to the deployment and automation resources that created the function app in the source region, re-run the same deployment steps in the target region to create and redeploy your app.

If you only have access to the source code but not the deployment and automation resources you can deploy and configure the function app on the target region using any of the available [deployment technologies](#) or using one of the [continuous deployment methods](#).

Review configured resources

Review and configure the resources identified in the [Prepare](#) step above in the target region if they weren't configured during the deploy.

Move considerations

- If your deployment resources and automation doesn't create a function app, [create an app of the same type in a new hosting plan](#) in the target region
- Function app names are globally unique in Azure, so the app in the target region can't have the same name as the one in the source region
- References and application settings that connect your function app to dependencies need to be reviewed and, when needed, updated. For example, when you move a database that your functions call, you must also update the application settings or configuration to connect to the database in the target region. Some application settings such as the Application Insights instrumentation key or the Azure storage account used by the function app can be already be configured on the target region and do not need to be updated
- Remember to verify your configuration and test your functions in the target region
- If you had custom domain configured, [remap the domain name](#)
- For Functions running on Dedicated plans also review the [App Service Migration Plan](#) in case the plan is shared with web apps

Clean up source resources

After the move is complete, delete the function app and hosting plan from the source region. You pay for function apps in Premium or Dedicated plans, even when the app itself isn't running.

How to manually migrate an Azure IoT hub using an Azure Resource Manager template

- Article
- 04/16/2023
- 3 contributors

Feedback

In this article

1. [Compare automatic and manual migration steps](#)
2. [Things to consider](#)
3. [Methodology](#)
4. [How to handle message routing](#)

Show 7 more

Use the Azure portal, Azure Resource Manager templates, and Azure IoT Hub service SDKs to migrate an IoT hub to a new region, a new tier, or a new configuration.

The steps in this article are useful if you want to:

- Upgrade from the free tier to a basic or standard tier IoT hub.
- Move an IoT hub to a new region.
- Export IoT hub state information to have as a backup.
- Increase the number of [partitions](#) for an IoT hub.
- Set up a hub for a development, rather than production, environment.
- Enable a custom implementation of multi-hub high availability. For more information, see the [How to achieve cross region HA section of IoT Hub high availability and disaster recovery](#).

To migrate a hub, you need a subscription with administrative access to the original hub. You can put the new hub in a new resource group and region, in the same subscription as the original hub, or even in a new subscription. You just can't use the same name because the hub name has to be globally unique.

Compare automatic and manual migration steps

The outcome of this article is similar to [How to automatically migrate an IoT hub using the Azure CLI](#), but with a different process. Before you begin, decide which process is right for your scenario.

- The manual process (this article):
 - Migrates your device registry and your routing and endpoint information. You have to manually recreate other configuration details in the new IoT hub.
 - Is faster for migrating large numbers of devices (for example, more than 100,000).
 - Uses an Azure Storage account to transfer the device registry.
 - Scrubs connection strings for routing and file upload endpoints from the ARM template output, and you need to manually add them back in.
- The Azure CLI process:
 - Migrates your device registry, your routing and endpoint information, and other configuration details like IoT Edge deployments or automatic device management configurations.
 - Is easier for migrating small numbers of devices (for example, up to 10,000).
 - Doesn't require an Azure Storage account.
 - Collects connection strings for routing and file upload endpoints and includes them in the ARM template output.

Things to consider

There are several things to consider before migrating an IoT hub.

- Make sure that all of the features available in the original location are also available in the new location. Some services are in preview, and not all features are available everywhere.
- Don't remove the original resources before creating and verifying the migrated version. Once you remove a hub, it's gone forever, and there's no way to recover it to check the settings or data to make sure the hub is replicated correctly.
- Data for the original IoT hub isn't migrated. This data includes device messages, cloud-to-device (C2D) commands, and job-related information such as schedules and history. Metrics and logging results are also not migrated.
- You need to schedule downtime for the migration. Cloning the devices to the new hub takes time. If you use the Import/Export method, benchmark testing has revealed that it could take around two hours to move 500,000 devices, and four hours to move a million devices.
- You can copy devices to the new hub without shutting down or changing the devices.
 - If the devices were originally provisioned using DPS, update their enrollments to point to the new IoT hub. Then, reprovision the devices to update the connection information stored in each device.
 - Otherwise, you have to use the import/export method to move the devices, and then the devices have to be modified to use the new hub. For example, you can set up your device to consume the IoT Hub host name from the twin desired properties. The device takes that IoT Hub host name, disconnect the device from the old hub, and reconnect it to the new one.
- You need to update any certificates so you can use them with the new resources. Also, you probably have the hub defined in a DNS table somewhere and need to update that DNS information.

Methodology

This is the general method we recommend for migrating an IoT hub.

1. Export the hub and its settings to a Resource Manager template.
2. Make the necessary changes to the template, such as updating all occurrences of the name and the location for the migrated hub. For any resources in the template used for message routing endpoints, update the key in the template for that resource.
3. Import the template into a new resource group in the new location. This step creates the new IoT hub.

4. Debug as needed.
5. Add anything that wasn't exported to the template.

For example, consumer groups aren't exported to the template. You need to add the consumer groups to the template manually or use the [Azure portal](#) after the hub is created.

6. Copy the devices from the original hub to the new hub. This process is covered in the section [Manage the devices registered to the IoT hub](#).

How to handle message routing

If your hub uses [message routing](#), exporting the template for the hub includes the routing configuration, but it doesn't include the resources themselves. If you're migrating the IoT hub to a new region, you must choose whether to move the routing resources to the new location as well or to leave them in place and continue to use them "as is". There may be a small performance hit from routing messages to endpoint resources in a different region.

If the hub uses message routing, you have two choices.

- Move the resources used for the routing endpoints to the new location.
 1. Create the new resources yourself either manually in the [Azure portal](#) or by using Resource Manager templates.
 2. Rename all of the resources when you create them in the new location, as they require globally unique names.
 3. Update the resource names and the resource keys in the new hub's template before creating the new hub. The resources should be present when the new hub is created.
- Don't move the resources used for the routing endpoints. Use them "in place".
 1. In the step where you edit the template, you need to retrieve the keys for each routing resource and put them in the template before you create the new hub.
 2. The hub still references the original routing resources and routes messages to them as configured. You'll have a small performance hit because the hub and the routing endpoint resources aren't in the same location.

Prepare to migrate the hub to another region

This section provides specific instructions for migrating the hub.

Export the original hub to a resource template

1. Sign into the [Azure portal](#).
2. Navigate to the IoT hub that you want to move.
3. Select **Export template** from the list of properties and settings for the hub.
4. Select **Download** to download the template. Save the file somewhere you can find it again.

View the template

Go to the downloaded template, which is contained in a zip file. Extract the zip file and find the file called `template.json`.

The following example is for a generic hub with no routing configuration. It's an S1 tier hub (with 1 unit) called **ContosoHub** in region **westus**:

JSONCopy

```
{  
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
    "IotHubs_ContosoHubConnectionString": {  
      "type": "SecureString"  
    },  
    "IotHubs_ContosoHub_containerName": {  
      "type": "SecureString"  
    },  
    "IotHubs_ContosoHub_name": {  
      "defaultValue": "ContosoHub",  
      "type": "String"  
    }  
  },  
  "variables": {},  
  "resources": [  
    {  
      "type": "Microsoft.Devices/IotHubs",  
      "apiVersion": "2021-07-01",  
      "name": "[parameters('IotHubs_ContosoHub_name')]",  
      "location": "westus",  
      "sku": {  
        "name": "S1",  
        "tier": "Standard",  
      }  
    }  
  ]  
}
```

```

        "capacity": 1
    },
    "identity": {
        "type": "None"
    },
    "properties": {
        "ipFilterRules": [],
        "eventHubEndpoints": {
            "events": {
                "retentionTimeInDays": 1,
                "partitionCount": 4
            }
        },
        "routing": {
            "endpoints": {
                "serviceBusQueues": [],
                "serviceBusTopics": [],
                "eventHubs": [],
                "storageContainers": []
            },
            "routes": [],
            "fallbackRoute": {
                "name": "$fallback",
                "source": "DeviceMessages",
                "condition": "true",
                "endpointNames": [
                    "events"
                ],
                "isEnabled": true
            }
        },
        "storageEndpoints": {
            "$default": {
                "sasTtlAsIso8601": "PT1H",
                "connectionString":
                "[parameters('IotHubs_ContosoHubConnectionString')]",
                "containerName":
                "[parameters('IotHubs_ContosoHub_containerName')]"
            }
        },
        "messagingEndpoints": {
            "fileNotifications": {
                "lockDurationAsIso8601": "PT1M",
                "ttlAsIso8601": "PT1H",
                "maxDeliveryCount": 10
            }
        },
        "enableFileUploadNotifications": false,
        "cloudToDevice": {
            "maxDeliveryCount": 10,
            "defaultTtlAsIso8601": "PT1H",
            "feedback": {
                "lockDurationAsIso8601": "PT1M",
                "ttlAsIso8601": "PT1H",
                "maxDeliveryCount": 10
            }
        },
        "features": "None",
        "disableLocalAuth": false,

```

```
        "allowedFqdnList": []
    }
}
]
```

Edit the template

You have to make some changes before you can use the template to create the new hub in the new region. Use [Visual Studio Code](#) or a text editor to edit the template.

Edit the hub name and location

1. Remove the container name parameter section at the top. **ContosoHub** doesn't have an associated container.

JSONCopy

```
"parameters": {
...
    "IotHubs_ContosoHub_containerName": {
        "type": "SecureString"
    },
...
},
```

2. Remove the **storageEndpoints** property.

JSONCopy

```
"properties": {
...
    "storageEndpoints": {
        "$default": {
            "sasTtlAsIso8601": "PT1H",
            "connectionString":
                "[parameters('IotHubs_ContosoHubConnectionString')]",
            "containerName":
                "[parameters('IotHubs_ContosoHub_containerName')]"
        }
    },
...
}
```

3. If you're moving the hub to a new region, change the **location** property under **resources**.

JSONCopy

```
"location": "westus",
```

Update the routing endpoint resources

When you export the Resource Manager template for a hub that has routing configured, you see that the keys for those resources aren't provided in the exported template. Their placement is denoted by asterisks. You must fill them in by going to those resources in the portal and retrieving the keys **before** you import the new hub's template and create the hub.

If you moved the routing resources as well, update the name, ID, and resource group of each endpoint as well.

1. Retrieve the keys required for any of the routing resources and put them in the template. You can retrieve the key(s) from the resource in the [Azure portal](#).

- For example, if you're routing messages to a storage container, find the storage account in the portal. Under the Settings section, select **Access keys**, then copy one of the keys. Here's what the key looks like when you first export the template:

JSONCopy

```
"connectionString": "DefaultEndpointsProtocol=https;  
AccountName=fabrikamstorage1234;AccountKey=****",  
"containerName": "fabrikamresults",
```

After you retrieve the account key for the storage account, put it in the template in the AccountKey=**** clause in the place of the asterisks.

- For service bus queues, get the Shared Access Key matching the SharedAccessKeyName. Here's the key and the SharedAccessKeyName in the json:

JSONCopy

```
"connectionString":  
"Endpoint=sb://fabrikamsbnamespace1234.servicebus.windows.  
net:5671/;  
SharedAccessKeyName=iothubroutes_FabrikamResources;  
SharedAccessKey=****;  
EntityPath=fabrikamsbqueue1234",
```

- The same applies for the Service Bus Topics and Event Hubs connections.

Create the new hub by loading the template

Create the new hub using the edited template. If you have routing resources that are going to move, the resources should be set up in the new location and the references in the template updated to match. If you aren't moving the routing resources, they should be in the template with the updated keys.

1. Sign into the [Azure portal](#).
2. Select **Create a resource**.
3. In the search box, search for and select **template deployment (deploy using custom templates)**. On the screen for the template deployment, select **Create**.
4. On the **Custom deployment** page, select **Build your own template in the editor**, which enables you to upload your template from a file.
5. Select **Load file**.
6. Browse for the new template you edited and select it, then select **Open**. It loads your template in the edit window. Select **Save**.
7. Fill in the following fields on the custom deployment page.

Subscription: Select the subscription to use.

Resource group: Select an existing resource group or create a new one.

Region: If you selected an existing resource group, the region is filled in for you to match the location of the resource group. If you created a new resource group, this is its location.

Connection string: Fill in the connection string for your hub.

Hub name: Give the new hub a name.

8. Select the **Review + create** button.

9. Select the **Create** button. The portal validates your template and deploys your new hub. If you have routing configuration data, it is included in the new hub, but points at the resources in the prior location.

Manage the devices registered to the IoT hub

Now that you have your new hub up and running, you need to copy all of the devices from the original hub to the new one.

There are multiple ways to copy the devices. You either originally used [Device Provisioning Service \(DPS\)](#) to provision the devices, or you didn't. If you did, this process isn't difficult. If you didn't, this process can be complicated.

If you didn't use DPS to provision your devices, you can skip the next section and start with [Use Import/Export to move the devices to the new hub](#).

Use DPS to reprovision the devices in the new hub

To use DPS to move the devices to the new location, see [How to reprovision devices](#). When you're finished, you can view the devices in the [Azure portal](#) and verify they are in the new location.

Go to the new hub using the [Azure portal](#). Select your hub, then select **IoT Devices**. You see the devices that were reprovisioned to the new hub. You can also view the properties for the new hub.

If you have implemented routing, test and make sure your messages are routed to the resources correctly.

Roll back the changes after using DPS

If you want to roll back the changes, reprovision the devices from the new hub to the old one.

You're now finished migrating your hub and its devices. You can skip to [Clean-up](#).

Use import-export to move the devices to the new hub

The application targets .NET Core, so you can run it on either Windows or Linux. You can download the sample, retrieve your connection strings, set the flags for which bits you want to run, and run it. You can do this without ever opening the code.

Download the sample

1. Use the IoT C# samples here: [Azure IoT SDK for C#](#). Download the zip file and unzip it on your computer.
2. The pertinent code is in ./iothub/service/samples/how to guides/ImportExportDevicesSample. You don't need to view or edit the code in order to run the application.
3. To run the application, specify three connection strings and five options. You pass this data in as command-line arguments or use environment variables, or use a combination of the two. We're going to pass the options in as command line arguments, and the connection strings as environment variables.

The reason for this is because the connection strings are long and ungainly, and unlikely to change, but you might want to change the options and run the application more than once. To change the value of an environment variable, you have to close the command window and Visual Studio or Visual Studio Code, whichever you're using.

Options

Here are the five options you specify when you run the application:

- **addDevices** (argument 1) - set this option to `True` if you want to add virtual devices that are generated for you. These devices are added to the source hub. Also, set **numToAdd** (argument 2) to specify how many devices you want to add. The maximum number of devices you can register to a hub is one million. The purpose of this option is for testing. You can generate a specific number of devices, and then copy them to another hub.
- **copyDevices** (argument 3) - set this option to `True` to copy the devices from one hub to another.
- **deleteSourceDevices** (argument 4) - set this option to `True` to delete all of the devices registered to the source hub. We recommend waiting until you are certain all of the devices have been transferred before you run this. Once you delete the devices, you can't get them back.

- **deleteDestDevices** (argument 5) - set this option to `True` to delete all of the devices registered to the destination hub. You might want to do this if you want to copy the devices more than once.

The basic command is `dotnet run`, which tells .NET to build the local csproj file and then run it. You add your command-line arguments to the end before you run it.

Your command-line will look like these examples:

ConsoleCopy

```
// Format: dotnet run add-devices num-to-add copy-devices delete-source-devices delete-destination-devices

// Add 1000 devices, don't copy them to the other hub, or delete them.
// The first argument is true, numToAdd is 50, and the other arguments are false.
dotnet run true 1000 false false false

// Copy the devices you just added to the other hub; don't delete anything.
// The first argument is false, numToAdd is 0, copy-devices is true, and the delete arguments are both false
dotnet run false 0 true false false
```

Use environment variables for the connection strings

1. To run the sample, you need the connection strings to the old and new IoT hubs, and to a storage account you can use for temporary work files. We will store the values for these in environment variables.
2. To get the connection string values, sign in to the [Azure portal](#).
3. Put the connection strings somewhere you can retrieve them, such as NotePad. If you copy the following, you can paste the connection strings in directly where they go. Don't add spaces around the equal sign, or it changes the variable name. Also, you don't need double-quotes around the connection strings. If you put quotes around the storage account connection string, the script fails.

Set the environment variables in Windows:

ConsoleCopy

```
SET IOTHUB_CONN_STRING=<put connection string to original IoT hub here>
SET DEST_IOTHUB_CONN_STRING=<put connection string to destination IoT hub here>
SET STORAGE_ACCT_CONN_STRING=<put connection string to the storage account here>
```

Set the environment variables in Linux:

ConsoleCopy

```
export IOTHUB_CONN_STRING=<put connection string to original IoT hub here>
export DEST_IOTHUB_CONN_STRING=<put connection string to destination IoT hub here>
export STORAGE_ACCT_CONN_STRING=<put connection string to the storage account here>
```

4. For the IoT hub connection strings, go to each hub in the portal. You can search in **Resources** for the hub. If you know the Resource Group, you can go to **Resource groups**, select your resource group, and then select the hub from the list of assets in that resource group.
5. Select **Shared access policies** from the Settings for the hub, then select **iothubowner** and copy one of the connection strings. Do the same for the destination hub. Add them to the appropriate SET commands.
6. For the storage account connection string, find the storage account in **Resources** or under its **Resource group** and open it.
7. Under the Settings section, select **Access keys** and copy one of the connection strings. Put the connection string in your text file for the appropriate SET command.

Now you have the environment variables in a file with the SET commands, and you know what your command-line arguments are. Let's run the sample.

Run the sample application and using command-line arguments

1. Open a command prompt window. Select Windows and type in `command prompt` to get the command prompt window.
2. Copy the commands that set the environment variables, one at a time, and paste them into the command prompt window and select Enter. When you're finished, type `SET` in the command prompt window to see your environment variables and their values. Once you've copied these into the command prompt window, you don't have to copy them again, unless you open a new command prompt window.
3. In the command prompt window, change directories until you are in `./ImportExportDevicesSample` (where the `ImportExportDevicesSample.csproj` file exists). Then type the following, and include your command-line arguments.

ConsoleCopy

```
// Format: dotnet run add-devices num-to-add copy-devices delete-source-devices delete-destination-devices
dotnet run arg1 arg2 arg3 arg4 arg5
```

The dotnet command builds and runs the application. Because you're passing in the options when you run the application, you can change the values of them each time you run the application. For example, you may want to run it once and create new devices, then run it again and copy those devices to a new hub, and so on. You can also perform all the steps in the same run, although we recommend not deleting any devices until you're certain you're finished with the migration. Here's an example that creates 1000 devices and then copies them to the other hub.

ConsoleCopy

```
// Format: dotnet run add-devices num-to-add copy-devices delete-source-devices delete-destination-devices

// Add 1000 devices, don't copy them to the other hub or delete them.
dotnet run true 1000 false false false

// Do not add any devices. Copy the ones you just created to the other
hub; don't delete anything.
dotnet run false 0 true false false
```

After you verify that the devices were copied successfully, you can remove the devices from the source hub like this:

ConsoleCopy

```
// Format: dotnet run add-devices num-to-add copy-devices delete-source-devices delete-destination-devices
// Delete the devices from the source hub.
dotnet run false 0 false true false
```

Run the sample application using Visual Studio

1. If you want to run the application in Visual Studio, change your current directory to the folder where the azureiot.sln file resides. Then run this command in the command prompt window to open the solution in Visual Studio. You must do this in the same command window where you set the environment variables, so those variables are known.

ConsoleCopy

```
azureiot.sln
```

2. Right-click on the project *ImportExportDevicesSample* and select **Set as startup project**.
3. Set the variables at the top of Program.cs in the ImportExportDevicesSample folder for the five options.

C#Copy

```
// Add randomly created devices to the source hub.  
private static bool addDevices = true;  
//If you ask to add devices, this will be the number added.  
private static int numToAdd = 0;  
// Copy the devices from the source hub to the destination hub.  
private static bool copyDevices = false;  
// Delete all of the devices from the source hub. (It uses the  
IoTHubConnectionString).  
private static bool deleteSourceDevices = false;  
// Delete all of the devices from the destination hub. (Uses the  
DestIoTHubConnectionString).  
private static bool deleteDestDevices = false;
```

4. Select F5 to run the application. After it finishes running, you can view the results.

View the results

You can view the devices in the [Azure portal](#) and verify they are in the new location.

1. Go to the new hub using the [Azure portal](#). Select your hub, then select **IoT Devices**. You see the devices you copied from the old hub to the new hub. You can also view the properties for the new hub.
2. Check for import/export errors by going to the Azure storage account in the [Azure portal](#) and looking in the devicefiles container for the `ImportErrors.log`. If this file is empty (the size is 0), there were no errors. If you try to import the same device more than once, it rejects the device the second time and adds an error message to the log file.

Commit the changes

At this point, you have copied your hub to the new location and migrated the devices to the new hub. Now you need to make changes so the devices work with the new hub.

To commit the changes, here are the steps you need to perform:

- Update each device to change the IoT Hub host name to point the IoT Hub host name to the new hub. You should do this using the same method you used when you first provisioned the device.
- Change any applications you have that refer to the old hub to point to the new hub.
- After you're finished, the new hub should be up and running. The old hub should have no active devices and be in a disconnected state.

Roll back the changes

If you decide to roll back the changes, here are the steps to perform:

- Update each device to change the IoT Hub Hostname to point the IoT Hub Hostname for the old hub. You should do this using the same method you used when you first provisioned the device.
- Change any applications you have that refer to the new hub to point to the old hub. For example, if you're using Azure Analytics, you may need to reconfigure your [Azure Stream Analytics input](#).
- Delete the new hub.
- If you have routing resources, the configuration on the old hub should still point to the correct routing configuration, and should work with those resources after the hub is restarted.

Check the results

To check the results, change your IoT solution to point to your hub in the new location and run it. In other words, perform the same actions with the new hub that you performed with the previous hub and make sure they work correctly.

If you have implemented routing, test and make sure your messages are routed to the resources correctly.

Clean up

Don't clean up until you're certain the new hub is up and running and the devices are working correctly. Also be sure to test the routing if you're using that feature. When you're ready, clean up the old resources by performing these steps:

- If you haven't already, delete the old hub. This removes all of the active devices from the hub.
- If you have routing resources that you moved to the new location, you can delete the old routing resources.

Move Microsoft.Resources resources to new region

- Article
- 04/20/2021
- 2 contributors

Feedback

In this article

1. [Move template specs to new region](#)
2. [Move deployment scripts to new region](#)
3. [Next steps](#)

You may need to move an existing resource to a new region. This article shows how to move two resource types - templateSpecs and deploymentScripts - that are in the Microsoft.Resources namespace.

Move template specs to new region

If you have a [template spec](#) in one region and want to move it to new region, you can export the template spec and redeploy it.

1. Use the command to export an existing template spec. For the parameter values, provide the values that match the template spec you want to export.

For Azure PowerShell, use:

Azure PowerShellCopy

```
Export-AzTemplateSpec ` 
    -ResourceGroupName demoRG ` 
    -Name demoTemplateSpec ` 
    -Version 1.0 ` 
    -OutputFolder c:\export
```

For Azure CLI, use:

Azure CLICopy

```
az template-specs export \ 
    --resource-group demoRG \ 
    --name demoTemplateSpec \ 
    --version 1.0 \ 
    --output-folder c:\export
```

2. Use the exported template spec to create a new template spec. The following examples show westus for the new region but you can provide the region you want.

For Azure PowerShell, use:

Azure PowerShellCopy

```
New-AzTemplateSpec ` 
    -Name movedTemplateSpec ` 
    -Version 1.0 `
```

```
-ResourceGroupName newRG  
-Location westus  
-TemplateJsonFile c:\export\1.0.json
```

For Azure CLI, use:

Azure CLICopy

```
az template-specs create \  
  --name movedTemplateSpec \  
  --version "1.0" \  
  --resource-group newRG \  
  --location "westus" \  
  --template-file "c:\export\demoTemplateSpec.json"
```

Move deployment scripts to new region

1. Select the resource group that contains the deployment script you want to move to a new region.
2. [Export the template](#). When exporting, select the deployment script and any other required resources.
3. In the exported template, delete the following properties:
 - tenantId
 - principalId
 - clientId
4. The exported template has a hardcoded value for the region of the deployment script.

JSONCopy

```
"location": "westus2",
```

Change the template to allow a parameter for setting the location. For more information, see [Set resource location in ARM template](#)

JSONCopy

```
"location": "[parameters('location')]",
```

5. [Deploy the exported template](#) and specify a new region for the deployment script.

Move an Azure Storage account to another region

- Article
- 04/03/2023
- 10 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare](#)
3. [Move](#)
4. [Discard or clean up](#)
5. [Next steps](#)

To move a storage account, create a copy of your storage account in another region. Then, move your data to that account by using AzCopy, or another tool of your choice.

In this article, you'll learn how to:

- Export a template.
- Modify the template by adding the target region and storage account name.
- Deploy the template to create the new storage account.
- Configure the new storage account.
- Move data to the new storage account.
- Delete the resources in the source region.

Prerequisites

- Ensure that the services and features that your account uses are supported in the target region.
- For preview features, ensure that your subscription is allowlisted for the target region.

Prepare

To get started, export, and then modify a Resource Manager template.

Export a template

This template contains settings that describe your storage account.

- [Portal](#)
- [PowerShell](#)

To export a template by using Azure portal:

1. Sign in to the [Azure portal](#).
2. Select **All resources** and then select your storage account.
3. Select > **Automation** > **Export template**.
4. Choose **Download** in the **Export template** blade.
5. Locate the .zip file that you downloaded from the portal, and unzip that file to a folder of your choice.

This zip file contains the .json files that comprise the template and scripts to deploy the template.

Modify the template

Modify the template by changing the storage account name and region.

- [Portal](#)
- [PowerShell](#)

To deploy the template by using Azure portal:

1. In the Azure portal, select **Create a resource**.
2. In **Search the Marketplace**, type **template deployment**, and then press **ENTER**.
3. Select **Template deployment**.
4. Select **Create**.
5. Select **Build your own template in the editor**.
6. Select **Load file**, and then follow the instructions to load the **template.json** file that you downloaded in the last section.
7. In the **template.json** file, name the target storage account by setting the default value of the storage account name. This example sets the default value of the storage account name to `mytargetaccount`.

JSONCopy

```
"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
"contentVersion": "1.0.0.0",
"parameters": {
    "storageAccounts_mysourceaccount_name": {
        "defaultValue": "mytargetaccount",
        "type": "String"
    }
}
```

},

8. Edit the **location** property in the **template.json** file to the target region. This example sets the target region to **centralus**.

JSONCopy

```
"resources": [{}  
    "type": "Microsoft.Storage/storageAccounts",  
    "apiVersion": "2019-04-01",  
    "name": "[parameters('storageAccounts_mysourceaccount_name')]",  
    "location": "centralus"  
]
```

To obtain region location codes, see [Azure Locations](#). The code for a region is the region name with no spaces, **Central US** = **centralus**.

Move

Deploy the template to create a new storage account in the target region.

- [Portal](#)
- [PowerShell](#)

1. Save the **template.json** file.
2. Enter or select the property values:
 - **Subscription**: Select an Azure subscription.
 - **Resource group**: Select **Create new** and give the resource group a name.
 - **Location**: Select an Azure location.
3. Click the **I agree to the terms and conditions stated above** checkbox, and then click the **Select Purchase** button.

Tip

If you receive an error which states that the XML specified is not syntactically valid, compare the JSON in your template with the schemas described in the [Azure Resource Manager documentation](#).

Configure the new storage account

Some features won't export to a template, so you'll have to add them to the new storage account.

The following table lists these features along with guidance for adding them to your new storage account.

Feature	Guidance
Lifecycle management policies	Manage the Azure Blob storage lifecycle
Static websites	Host a static website in Azure Storage
Event subscriptions	Reacting to Blob storage events
Alerts	Create, view, and manage activity log alerts by using Azure Monitor
Content Delivery Network (CDN)	Use Azure CDN to access blobs with custom domains over HTTPS

Note

If you set up a CDN for the source storage account, just change the origin of your existing CDN to the primary blob service endpoint (or the primary static website endpoint) of your new account.

Move data to the new storage account

AzCopy is the preferred tool to move your data over. It's optimized for performance. One way that it's faster, is that data is copied directly between storage servers, so AzCopy doesn't use the network bandwidth of your computer. Use AzCopy at the command line or as part of a custom script. See [Get started with AzCopy](#).

You can also use Azure Data Factory to move your data over. It provides an intuitive user interface. To use Azure Data Factory, see any of these links:

- [Copy data to or from Azure Blob storage by using Azure Data Factory](#)
- [Copy data to or from Azure Data Lake Storage Gen2 using Azure Data Factory](#)
- [Copy data from or to Azure Files by using Azure Data Factory](#)
- [Copy data to and from Azure Table storage by using Azure Data Factory](#)

Discard or clean up

After the deployment, if you want to start over, you can delete the target storage account, and repeat the steps described in the [Prepare](#) and [Move](#) sections of this article.

To commit the changes and complete the move of a storage account, delete the source storage account.

- [Portal](#)
- [PowerShell](#)

To remove a storage account by using the Azure portal:

1. In the Azure portal, expand the menu on the left side to open the menu of services, and choose **Storage accounts** to display the list of your storage accounts.
2. Locate the target storage account to delete, and right-click the **More** button (...) on the right side of the listing.
3. Select **Delete**, and confirm.

Move resources to new region - Azure SQL Database & Azure SQL Managed Instance

- Article
- 03/04/2023
- 14 contributors

Feedback

In this article

1. [Overview](#)
2. [Move a database](#)
3. [Move elastic pools](#)
4. [Move a managed instance](#)
5. [Next steps](#)

Applies to: Azure SQL Database Azure SQL Managed Instance

This article teaches you a generic workflow for how to move your database or managed instance to a new region.

Overview

There are various scenarios in which you'd want to move your existing database or managed instance from one region to another. For example, you're expanding your

business to a new region and want to optimize it for the new customer base. Or you need to move the operations to a different region for compliance reasons. Or Azure released a new region that provides a better proximity and improves the customer experience.

This article provides a general workflow for moving resources to a different region. The workflow consists of the following steps:

1. Verify the prerequisites for the move.
2. Prepare to move the resources in scope.
3. Monitor the preparation process.
4. Test the move process.
5. Initiate the actual move.
6. Remove the resources from the source region.

Note

This article applies to migrations within the Azure public cloud or within the same sovereign cloud.

Note

To move Azure SQL databases and elastic pools to a different Azure region, you can also use Azure Resource Mover (Recommended). Refer [this tutorial](#) for detailed steps to do the same.

Note

This article uses the Azure Az PowerShell module, which is the recommended PowerShell module for interacting with Azure. To get started with the Az PowerShell module, see [Install Azure PowerShell](#). To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

Move a database

Verify prerequisites

1. Create a target server for each source server.
2. Configure the firewall with the right exceptions by using [PowerShell](#).
3. Configure the servers with the correct logins. If you're not the subscription administrator or SQL server administrator, work with the administrator to assign the permissions that you need. For more information, see [How to manage Azure SQL Database security after disaster recovery](#).

4. If your databases are encrypted with transparent data encryption (TDE) and bring your own encryption key (BYOK or Customer-Managed Key) in Azure Key Vault, ensure that the correct encryption material is provisioned in the target regions.
 - The simplest way to do this is to add the encryption key from the existing key vault (that is being used as TDE Protector on source server) to the target server and then set the key as the TDE Protector on the target server

Note

A server or managed instance in one region can now be connected to a key vault in any other region.

- As a best practice to ensure the target server has access to older encryption keys (required for restoring database backups), run the [Get-AzSqlServerKeyVaultKey](#) cmdlet on the source server or [Get-AzSqlInstanceKeyVaultKey](#) cmdlet on the source managed instance to return the list of available keys and add those keys to the target server.
 - For more information and best practices on configuring customer-managed TDE on the target server, see [Azure SQL transparent data encryption with customer-managed keys in Azure Key Vault](#).
 - To move the key vault to the new region, see [Move an Azure key vault across regions](#)
5. If database-level audit is enabled, disable it and enable server-level auditing instead. After failover, database-level auditing will require the cross-region traffic, which isn't desired or possible after the move.
 6. For server-level audits, ensure that:
 - The storage container, Log Analytics, or event hub with the existing audit logs is moved to the target region.
 - Auditing is configured on the target server. For more information, see [Get started with SQL Database auditing](#).
 7. If your instance has a long-term retention policy (LTR), the existing LTR backups will remain associated with the current server. Because the target server is different, you'll be able to access the older LTR backups in the source region by using the source server, even if the server is deleted.

Note

This will be insufficient for moving between the sovereign cloud and a public region. Such a migration will require moving the LTR backups to the target server, which is not currently supported.

Prepare resources

1. Create a [failover group](#) between the server of the source and the server of the target.
2. Add the databases you want to move to the failover group.

Replication of all added databases will be initiated automatically. For more information, see [Using failover groups with SQL Database](#).

Monitor the preparation process

You can periodically call [Get-AzSqlDatabaseFailoverGroup](#) to monitor replication of your databases from the source to the target. The output object of `Get-AzSqlDatabaseFailoverGroup` includes a property for the **ReplicationState**:

- **ReplicationState = 2** (CATCH_UP) indicates the database is synchronized and can be safely failed over.
- **ReplicationState = 0** (SEEDING) indicates that the database is not yet seeded, and an attempt to fail over will fail.

Test synchronization

After **ReplicationState** is 2, connect to each database or subset of databases using the secondary endpoint `<fog-name>.secondary.database.windows.net` and perform any query against the databases to ensure connectivity, proper security configuration, and data replication.

Initiate the move

1. Connect to the target server using the secondary endpoint `<fog-name>.secondary.database.windows.net`.
2. Use [Switch-AzSqlDatabaseFailoverGroup](#) to switch the secondary managed instance to be the primary with full synchronization. This operation will succeed or it will roll back.
3. Verify that the command has completed successfully by using `nslookup <fog-name>.secondary.database.windows.net` to ascertain that the DNS CNAME entry points to the target region IP address. If the switch command fails, the CNAME won't be updated.

Remove the source databases

Once the move completes, remove the resources in the source region to avoid unnecessary charges.

1. Delete the failover group using [Remove-AzSqlDatabaseFailoverGroup](#).

2. Delete each source database using [Remove-AzSqlDatabase](#) for each of the databases on the source server. This will automatically terminate geo-replication links.
3. Delete the source server using [Remove-AzSqlServer](#).
4. Remove the key vault, audit storage containers, event hub, Azure Active Directory (Azure AD) instance, and other dependent resources to stop being billed for them.

Move elastic pools

Verify prerequisites

1. Create a target server for each source server.
2. Configure the firewall with the right exceptions using [PowerShell](#).
3. Configure the servers with the correct logins. If you're not the subscription administrator or server administrator, work with the administrator to assign the permissions that you need. For more information, see [How to manage Azure SQL Database security after disaster recovery](#).
4. If your databases are encrypted with transparent data encryption and use your own encryption key in Azure Key Vault, ensure that the correct encryption material is provisioned in the target region.
5. Create a target elastic pool for each source elastic pool, making sure the pool is created in the same service tier, with the same name and the same size.
6. If a database-level audit is enabled, disable it and enable server-level auditing instead. After failover, database-level auditing will require cross-region traffic, which is not desired, or possible after the move.
7. For server-level audits, ensure that:
 - The storage container, Log Analytics, or event hub with the existing audit logs is moved to the target region.
 - Audit configuration is configured at the target server. For more information, see [SQL Database auditing](#).
8. If your instance has a long-term retention policy (LTR), the existing LTR backups will remain associated with the current server. Because the target server is different, you'll be able to access the older LTR backups in the source region using the source server, even if the server is deleted.

Note

This will be insufficient for moving between the sovereign cloud and a public region. Such a migration will require moving the LTR backups to the target server, which is not currently supported.

Prepare to move

1. Create a separate [failover group](#) between each elastic pool on the source server and its counterpart elastic pool on the target server.
2. Add all the databases in the pool to the failover group.

Replication of the added databases will be initiated automatically. For more information, see [Using failover groups with SQL Database](#).

Note

While it is possible to create a failover group that includes multiple elastic pools, we strongly recommend that you create a separate failover group for each pool. If you have a large number of databases across multiple elastic pools that you need to move, you can run the preparation steps in parallel and then initiate the move step in parallel. This process will scale better and will take less time compared to having multiple elastic pools in the same failover group.

Monitor the preparation process

You can periodically call [Get-AzSqlDatabaseFailoverGroup](#) to monitor replication of your databases from the source to the target. The output object of `Get-AzSqlDatabaseFailoverGroup` includes a property for the **ReplicationState**:

- **ReplicationState = 2** (CATCH_UP) indicates the database is synchronized and can be safely failed over.
- **ReplicationState = 0** (SEEDING) indicates that the database is not yet seeded, and an attempt to fail over will fail.

Test synchronization

Once **ReplicationState** is 2, connect to each database or subset of databases using the secondary endpoint <fog-name>.secondary.database.windows.net and perform any query against the databases to ensure connectivity, proper security configuration, and data replication.

Initiate the move

1. Connect to the target server using the secondary endpoint <fog-name>.secondary.database.windows.net.

2. Use [Switch-AzSqlDatabaseFailoverGroup](#) to switch the secondary managed instance to be the primary with full synchronization. This operation will either succeed, or it will roll back.
3. Verify that the command has completed successfully by using nslookup <failover-group-name>.secondary.database.windows.net to ascertain that the DNS CNAME entry points to the target region IP address. If the switch command fails, the CNAME won't be updated.

Remove the source elastic pools

Once the move completes, remove the resources in the source region to avoid unnecessary charges.

1. Delete the failover group using [Remove-AzSqlDatabaseFailoverGroup](#).
2. Delete each source elastic pool on the source server using [Remove-AzSqlElasticPool](#).
3. Delete the source server using [Remove-AzSqlServer](#).
4. Remove the key vault, audit storage containers, event hub, Azure AD instance, and other dependent resources to stop being billed for them.

Move a managed instance

Verify prerequisites

1. For each source managed instance, create a target instance of SQL Managed Instance of the same size in the target region.
2. Configure the network for a managed instance. For more information, see [network configuration](#).
3. Configure the target master database with the correct logins. If you're not the subscription or SQL Managed Instance administrator, work with the administrator to assign the permissions that you need.
4. If your databases are encrypted with transparent data encryption and use your own encryption key in Azure Key Vault, ensure that the Azure Key Vault with identical encryption keys exists in both source and target regions. For more information, see [Transparent data encryption with customer-managed keys in Azure Key Vault](#).
5. If audit is enabled for the managed instance, ensure that:
 - The storage container or event hub with the existing logs is moved to the target region.
 - Audit is configured on the target instance. For more information, see [Auditing with SQL Managed Instance](#).
6. If your instance has a long-term retention policy (LTR), the existing LTR backups will remain associated with the current instance. Because the target instance is different, you'll be able to access the older LTR backups in the source region using the source instance, even if the instance is deleted.

Note

This will be insufficient for moving between the sovereign cloud and a public region. Such a migration will require moving the LTR backups to the target instance, which is not currently supported.

Prepare resources

Create a failover group between each source managed instance and the corresponding target instance of SQL Managed Instance.

Replication of all databases on each instance will be initiated automatically. For more information, see [Auto-failover groups](#).

Monitor the preparation process

You can periodically call [Get-AzSqlDatabaseFailoverGroup](#) to monitor replication of your databases from the source to the target. The output object of `Get-AzSqlDatabaseFailoverGroup` includes a property for the **ReplicationState**:

- **ReplicationState = 2** (CATCH_UP) indicates the database is synchronized and can be safely failed over.
- **ReplicationState = 0** (SEEDING) indicates that the database isn't yet seeded, and an attempt to fail over will fail.

Test synchronization

Once **ReplicationState** is 2, connect to each database, or subset of databases using the secondary endpoint `<fog-name>.secondary.database.windows.net` and perform any query against the databases to ensure connectivity, proper security configuration, and data replication.

Initiate the move

1. Connect to the target managed instance by using the secondary endpoint `<fog-name>.secondary.database.windows.net`.
2. Use [Switch-AzSqlDatabaseFailoverGroup](#) to switch the secondary managed instance to be the primary with full synchronization. This operation will succeed, or it will roll back.
3. Verify that the command has completed successfully by using `nslookup <fog-name>.secondary.database.windows.net` to ascertain that the DNS CNAME entry points to the target region IP address. If the switch command fails, the CNAME won't be updated.

Remove the source managed instances

Once the move finishes, remove the resources in the source region to avoid unnecessary charges.

1. Delete the failover group using [Remove-AzSqlDatabaseFailoverGroup](#). This will drop the failover group configuration and terminate geo-replication links between the two instances.
2. Delete the source managed instance using [Remove-AzSqlInstance](#).
3. Remove any additional resources in the resource group, such as the virtual cluster, virtual network, and security group.

Copy, back up and move your Azure Stream Analytics jobs between regions

- Article
- 12/30/2022
- 8 contributors

Feedback

In this article

1. [Before you begin](#)
2. [Visual Studio Code](#)
3. [Next steps](#)

If you want to move, copy or back up your Azure Stream Analytics jobs in Azure, the Azure Stream Analytics extension for Visual Studio Code allows you to export an existing job in Azure cloud to your local computer. All the configurations of your Stream Analytics job will be saved locally and you can resubmit it to another cloud region.

Note

- Copying a job to another region does not copy the last output time. Therefore, you cannot use **When last stopped** option when starting the copied job.

Before you begin

- If you don't have an Azure subscription, create a [free account](#).
- Sign in to the [Azure portal](#).
- Install [Azure Stream Analytics extension for Visual Studio Code](#).

Visual Studio Code

1. Select the **Azure** icon on the Visual Studio Code Activity Bar and then expand **Stream Analytics** node. Your jobs should appear under your subscriptions.
2. To export a job to a local project, locate the job you wish to export in the **Stream Analytics Explorer** in Visual Studio Code. Then select a folder for your project.

The project is exported to the folder you select and added to your current workspace.

3. To publish the job to another region or backup using another name, select **Select from your subscriptions to publish** in the query editor (*.asaql) and follow the instructions.

Move Azure Stream Analytics cluster using Azure PowerShell

- Article
- 03/31/2023
- 4 contributors

Feedback

In this article

1. [Move Azure Stream Analytics cluster to another region](#)
2. [Next steps](#)

Learn how to use Azure Az PowerShell module to move your Azure Stream Analytics cluster to another region. You can move a Stream Analytics cluster by exporting the cluster's ARM template using Azure portal and from there deploy another cluster with the same ARM template to your alternate region.

Move Azure Stream Analytics cluster to another region

You must have the Azure Az PowerShell module installed on your machine to complete this procedure. Install the [latest version of PowerShell](#) available for your operating system.

1. Open Azure Portal.
2. Select the resource group that contains the Stream Analytics cluster you want to move.
3. Select the Azure Stream Analytics resource you want to move and then click **Export template**.
4. Decompress the file and save the template to your local drive.
5. Sign in to Azure PowerShell using your Azure credentials.

PowerShellCopy

[Connect-AzAccount](#)

6. Run the following command, using the value for the subscription of the cluster you want to move.

PowerShellCopy

`Set-AZContext: Set-AzContext -Subscription "<subscription id>"`

7. Deploy the Stream Analytics template from your local drive.

PowerShellCopy

[New-AzResourceGroupDeployment](#)

```
-Name <Example>
-ResourceGroupName <name of your resource group>
-TemplateFile <path-to-template>
```

For more information on how to deploy a template using Azure PowerShell, see [Deploy a template](#).

Move Azure VMs across regions

- Article
- 03/27/2023
- 3 contributors

Feedback

In this article

1. [Sign in to Azure](#)
2. [Prerequisites](#)
3. [Prepare VMs](#)

4. [Select resources](#)

Show 9 more

This tutorial shows you how to move Azure VMs and related network/storage resources to a different Azure region using [Azure Resource Mover](#).

Azure Resource Mover helps you move Azure resources between Azure regions. You might move your resources to another region for many reasons. For example, to take advantage of a new Azure region, to deploy features or services available in specific regions only, to meet internal policy and governance requirements, or in response to capacity planning requirements.

In this tutorial, you learn how to:

- Move Azure VMs to another region with Azure Resource Mover.
- Move resources associated with VMs to another region.

Note

Tutorials show the quickest path for trying out a scenario, and use default options where possible.

Sign in to Azure

If you don't have an Azure subscription, create a [free account](#) before you begin and sign in to the [Azure portal](#).

Prerequisites

Before you begin, verify the following:

Requirement Description

Resource [Review](#) the supported regions and other common questions.

Mover support

Subscription permissions Check that you have *Owner* access on the subscription containing the resources that you want to move

Why do I need Owner access? The first time you add a resource for a specific source and destination pair in an Azure subscription, Resource Mover creates a [system-assigned managed identity](#), formerly known as Managed Service Identity (MSI) that's trusted by the subscription. To create the identity, and to assign it the required role (Contributor or User Access administrator in the source subscription), the account you use to add resources needs *Owner* permissions on the subscription. [Learn more](#) about Azure roles.

Requirement	Description
VM support	<ul style="list-style-type: none"> - Check that the VMs you want to move are supported. - Verify supported Windows VMs. - Verify supported Linux VMs and kernel versions. - Check supported compute, storage, and networking settings.
Destination subscription	The subscription in the destination region needs enough quota to create the resources you're moving in the target region. If it doesn't have a quota, request additional limits .
Destination region charges	Verify pricing and charges associated with the target region to which you're moving VMs. Use the pricing calculator to help you.

Prepare VMs

To prepare VMs for the move, follow these steps:

1. After checking that VMs meet the requirements, ensure that the VMs you want to move are turned on. All VMs disks that you want to be available in the destination region must be attached and initialized in the VM.
2. Ensure that VMs have the latest trusted root certificates and an updated certificate revocation list (CRL). To do this:
 - On Windows VMs, install the latest Windows updates.
 - On Linux VMs, follow distributor guidance so that machines have the latest certificates and CRL.
3. Allow outbound connectivity from VMs:
 - If you're using a URL-based firewall proxy to control outbound connectivity, allow access to these [URLs](#).
 - If you're using network security group (NSG) rules to control outbound connectivity, create these [service tag rules](#).

Select resources

Note that, all supported resource types in resource groups within the selected source region are displayed. The resources that have already been added for moving across regions aren't shown. You move resources to a target region in the same subscription as the source region. If you want to change the subscription, you can do that after the resources are moved.

To select the resources you want to move, follow these steps:

1. In the Azure portal, search for *resource mover*. Under **Services**, select **Azure Resource Mover**.

2. In **Overview** pane, select **Get Started**.
3. In **Move resources > Source + destination** tab, do the following:
 - a. Select the source subscription and region.
 - b. Under **Destination**, select the region to which you want to move the VMs.
 - c. Select **Next**.
4. In **Move resources > Resources to move** tab, do the following:
 - a. Select the **Select resources** option.
 - b. In **Select resources**, select the VM. You can only add the [resources supported for the move](#).
 - c. Select **Done**.
5. In **Review**, check the source and the destination settings.
6. Select **Proceed** to begin adding the resources.
7. After the add process finishes successfully, on the **Notifications** pane, select **Added resources for move**.
8. After you select the notification, review the resources on the **Across regions** page.

Note

- Added resources are in a *Prepare pending* state.
- The resource group for the VMs is added automatically.
- If you want to remove a resource from a move collection, the method for doing that depends on where you are in the move process. [Learn more](#).

Resolve dependencies

To resolve dependencies before the move, follow these steps:

1. Dependencies are automatically validated in the background when you add the resources. If you still see the **Validate dependencies** option, select it to trigger the validation manually.
2. If dependencies are found, select **Add dependencies** to add them.
3. On **Add dependencies**, retain the default **Show all dependencies** option.
 - **Show all dependencies** iterates through all the direct and indirect dependencies for a resource. For example, for a VM, it shows the NIC, virtual network, network security groups (NSGs), and so on.
 - **Show first-level dependencies only** shows only direct dependencies. For example, for a VM it shows the NIC but not the virtual network.
4. Select the dependent resources you want to add and select **Add dependencies**. You can monitor the progress in the notifications.
5. Dependencies are automatically validated in the background once you add them. If you see a **Validate dependencies** button, select it to trigger the manual validation.

Move the source resource group

Before you can prepare and move the VMs, the VM resource group must be present in the target region.

Prepare to move the source resource group

During the Prepare process, Resource Mover generates Azure Resource Manager (ARM) templates using the resource group settings. Resources inside the resource group aren't affected.

To prepare to move a source resource group, follow these steps:

1. On the **Across regions** pane, select the source resource group > **Prepare**.
2. On **Prepare resources** pane, select **Prepare** to start the process.

Note

After preparing the resource group, it's in the *Initiate move pending* state.

Move the source resource group

To start the move, follows these steps:

1. On the **Across regions** pane, select the resource group > **Initiate Move**.
2. On the **Move Resources** pane, select **Initiate move**. The resource group moves into an *Initiate move in progress* state.
3. After initiating the move, the target resource group is created, based on the generated ARM template. The source resource group moves into a *Commit move pending* state.

To commit and finish the move process:

1. On the **Across regions** pane, select the resource group > **Commit move**.
2. On the **Move Resources** pane select **Commit**.

Note

After committing the move, the source resource group is in a *Delete source pending* state.

Prepare resources to move

Now that the source resource group is moved, you can prepare to move other resources that are in the *Prepare pending* state.

To move resources that are in the *Prepare pending* state, follow these steps:

1. On the **Across regions** pane, verify that the resources are now in a *Prepare pending* state, with no issues. If they're not, validate again and resolve any outstanding issues.
2. If you want to edit target settings before beginning the move, select the link in the **Destination configuration** column for the resource, and edit the settings. If you edit the target VM settings, the target VM size shouldn't be smaller than the source VM size.

Now that the source resource group is moved, you can prepare to move the other resources.

3. Select the resources you want to prepare.

4. Select **Prepare**.

Note

- During the prepare process, the Azure Site Recovery Mobility agent is installed on the VMs to replicate them.
- VM data is replicated periodically to the target region. This doesn't affect the source VM.
- Resource Move generates ARM templates for the other source resources.
- After preparing resources, they're in an *Initiate move pending* state.

Initiate the move

With resources prepared, you can now initiate the move. To start the move, follow these steps:

1. On the **Across regions** pane, select resources with state *Initiate move pending*.
2. Select **Initiate move** to start the process.
3. On the **Move resources** tab, select **Initiate move**.
4. Track the move progress in the notifications bar.

Note

- For VMs, replica VMs are created in the target region. The source VM is shut down, and some downtime occurs (usually minutes).
- Resource Mover recreates other resources using the ARM templates that were prepared. There's usually no downtime.
- After moving resources, they're in a *Commit move pending* state.

Commit or discard the move

After the initial move, you can decide if you want to commit the move or discard it.

- **Discard:** You might discard a move if you're testing, and you don't want to actually move the source resource. Discarding the move returns the resource to a state of *Initiate move pending*.
- **Commit:** Commit completes the move to the target region. After committing, a source resource will be in a state of *Delete source pending*, and you can decide if you want to delete it.

Discard the move

You can discard the move as follows:

1. On **Across regions** pane, select resources with state *Commit move pending*, and select **Discard move**.
2. On **Discard move** pane, select **Discard**.
3. Track move progress in the notifications bar.

Note

After discarding resources, VMs are in an *Initiate move pending* state.

Commit the move

If you want to complete the move process, commit the move. To commit the move, follow these steps:

1. On **Across regions** pane, select resources with state *Commit move pending*, and select **Commit move**.
2. On **Commit resources** pane, select **Commit**.
3. Track the commit progress in the notifications bar.

Note

- After committing the move, VMs stop replicating. The source VM isn't impacted by the commit.
- Commit doesn't impact source networking resources.
- After committing the move, resources are in a *Delete source pending* state.

Configure settings after the move

You can configure the following settings after the move process:

- The Mobility service isn't uninstalled automatically from VMs. Uninstall it manually, or leave it if you plan to move the server again.
- Modify Azure role-based access control (Azure RBAC) rules after the move.

Delete source resources after commit

After the move, you can optionally delete resources in the source region. To delete source resources after commit:

Note

A few resources, for example key vaults and SQL Server servers, can't be deleted from the portal, and must be deleted from the resource property page.

1. On **Across Regions** pane, select the name of the source resource that you want to delete.
2. Select **Delete source**.

Delete additional resources created for move

After the move, you can manually delete the move collection and Site Recovery resources that were created.

Before you delete the additional resources created for the move, note that:

- The move collection is hidden by default. To see it you must turn on hidden resources.
- The cache storage has a lock, before deleting the cache storage you must first delete the lock.

To delete the additional resources created for the move, follow these steps:

1. Locate the resources in resource group `RegionMoveRG-<sourcerregion>-<target-region>`.
2. Check that all the VM and other source resources in the source region have been moved or deleted. This ensures that there are no pending resources using them.
3. Delete the resources:
 - The move collection name is `movecollection-<sourcerregion>-<target-region>`.
 - The cache storage account name is `resmovecache<guid>`
 - The vault name is `ResourceMove-<sourcerregion>-<target-region>-GUID`.

Move logic app resources to other Azure resource groups, regions, or subscriptions

- Article
- 12/07/2022
- 6 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Move resources between subscriptions](#)
3. [Move resources between resource groups](#)
4. [Move resources between regions](#)
5. [Next steps](#)

To migrate your logic app or related resources to another Azure resource group, region, or subscription, you have various ways to complete these tasks, such as the Azure portal, Azure PowerShell, Azure CLI, and REST API. Before you move resources, review these considerations:

- You can move only [specific logic app resource types](#) between Azure resource groups or subscriptions.
- Check the [limits](#) on the number of logic app resources that you can have in your Azure subscription and in each Azure region. These limits affect whether you can move specific resource types when the region stays the same across subscriptions or resource groups. For example, you can have only one Free tier integration account for each Azure region in each Azure subscription.
- When you move resources, Azure creates new resource IDs. So, make sure that you use the new IDs instead and update any scripts or tools that are associated with the moved resources.
- After you migrate logic apps between subscriptions, resource groups, or regions, you must recreate or reauthorize any connections that require Open Authentication (OAuth).
- You can move an [integration service environment \(ISE\)](#) only to another resource group that exists in the same Azure region or Azure subscription. You can't move an ISE to a resource group that exists in a different Azure region or Azure subscription. Also, after such a move, you must update all references to the ISE in your logic app workflows, integration accounts, connections, and so on.

Prerequisites

- The same Azure subscription that was used to create the logic app or integration account that you want to move
- Resource owner permissions to move and set up the resources that you want. Learn more about [Azure role-based access control \(Azure RBAC\)](#).

Move resources between subscriptions

To move a resource, such as a logic app or integration account, to another Azure subscription, you can use the Azure portal, Azure PowerShell, Azure CLI, or REST API. These steps cover the Azure portal, which you can use when the resource's region stays the same. For other steps and general preparation, see [Move resources to a new resource group or subscription](#).

1. In the [Azure portal](#), find and select the logic app resource that you want to move.
2. On the resource navigation menu, select **Overview**. Next to the **Subscription** label, select **move**.

You can also go to the resource's **Properties** page, and under **Subscription Name**, select **Change subscription**.

3. On the **Move resources** page, select the logic app resource and any related resources that you want to move.
4. From the **Subscription** list, select the destination subscription.
5. From the **Resource group** list, select the destination resource group. Or, to create a different resource group, select **Create a new group**.
6. To confirm your understanding that any scripts or tools that are associated with the moved resources won't work until you update them with the new resource IDs, select the confirmation box, and then select **OK**.

Move resources between resource groups

To move a resource, such as a logic app, integration account, or [integration service environment \(ISE\)](#), to another Azure resource group, you can use the Azure portal, Azure PowerShell, Azure CLI, or REST API. These steps cover the Azure portal, which you can use when the resource's region stays the same. For other steps and general preparation, see [Move resources to a new resource group or subscription](#).

Before actually moving resources between groups, you can test whether you can successfully move your resource to another group. For more information, see [Validate your move](#).

1. In the [Azure portal](#), find and select the logic app resource that you want to move.
2. On the resource's **Overview** page, next to **Resource group**, select the **change** link.
3. On the **Move resources** page, select the logic app resource and any related resources that you want to move.
4. From the **Resource group** list, select the destination resource group. Or, to create a different resource group, select **Create a new group**.
5. To confirm your understanding that any scripts or tools that are associated with the moved resources won't work until you update them with the new resource IDs, select the confirmation box, and then select **OK**.

Move resources between regions

When you want to move a logic app to a different region, your options depend on the way that you created your logic app. Based on the option that you choose, you must recreate or reauthorize the connections in your logic app.

- In the Azure portal, recreate the logic app in the new region and reconfigure the workflow settings. To save time, you can copy the underlying workflow definition and connections from the source app to the destination app. To view the "code" behind a logic app, on the Logic App Designer toolbar, select **Code view**.
- By using Visual Studio and the Azure Logic Apps Tools for Visual Studio, you can [open and download your logic app](#) from the Azure portal as an [Azure Resource Manager template](#). This template is mostly ready for deployment and includes the resource definitions for your logic app, including the workflow itself, and connections. The template also declares parameters for the values to use at deployment. That way, you can more easily change where and how you deploy the logic app, based on your needs. To specify the location and other necessary information for deployment, you can use a separate parameters file.
- If you created and deployed your logic app by using continuous integration (CI) and continuous delivery (CD) tools, such as Azure Pipelines in Azure DevOps, you can deploy your app to another region by using those tools.

For more information about deployment templates for logic apps, see these topics:

- [Overview: Automate deployment for Azure Logic Apps by using Azure Resource Manager templates](#)
- [Find, open, and download your logic app from the Azure portal into Visual Studio](#)
- [Create Azure Resource Manager templates for Azure Logic Apps](#)
- [Deploy Azure Resource Manager templates for Azure Logic Apps](#)

Related resources

Some Azure resources, such as on-premises data gateway resources in Azure, can exist in a region that differs from the logic apps that use those resources. However, other Azure resources, such as linked integration accounts, must exist in the same region as your logic apps. Based on your scenario, make sure that your logic apps can access the resources that your apps expect to exist in the same region.

For example, to link a logic app to an integration account, both resources must exist in the same region. In scenarios such as disaster recovery, you usually want integration accounts that have the same configuration and artifacts. In other scenarios, you might need integration accounts with different configurations and artifacts.

Custom connectors in Azure Logic Apps are visible to the connectors' authors and users who have the same Azure subscription and the same Azure Active Directory tenant. These connectors are available in the same region where logic apps are deployed. For more information, see [Share custom connectors in your organization](#).

The template that you get from Visual Studio includes only the resource definitions for your logic app and its connections. So, if your logic app uses other resources, for example, an integration account and B2B artifacts, such as partners, agreements, and schemas, you must export that integration account's template by using the Azure portal. This template includes the resource definitions for both the integration account and artifacts. However, the template isn't fully parameterized. So, you must manually parameterize the values that you want to use for deployment.

Export templates for integration accounts

1. In the [Azure portal](#), find and open your integration account.
2. On your integration account's menu, under **Settings**, select **Export template**.
3. On the toolbar, select **Download**, and save the template.
4. Open and edit the template to parameterize the necessary values for deployment.

Move a Maintenance Control configuration to another region

- Article
- 11/15/2022
- 7 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare and move](#)
3. [Verify the move](#)
4. [Clean up source resources](#)
5. [Next steps](#)

Applies to: ✓ Linux VMs ✓ Windows VMs ✓ Flexible scale sets ✓ Uniform scale sets

Follow this article to move a Maintenance Control configuration to a different Azure region. You might want to move a configuration for a number of reasons. For example, to take advantage of a new region, to deploy features or services available in a specific region, to meet internal policy and governance requirements, or in response to capacity planning.

[Maintenance control](#), with customized maintenance configurations, allows you to control how platform updates are applied to VMs, and to Azure Dedicated Hosts. There are a couple of scenarios for moving maintenance control across regions:

- To move your maintenance control configuration, but not the resources associated with the configuration, follow the instructions in this article.
- To move the resources associated with a maintenance configuration, but not the configuration itself, follow [these instructions](#).
- To move both the maintenance configuration and the resources associated with it, first follow the instructions in this article. Then, follow [these instructions](#).

Prerequisites

Before you begin moving a maintenance control configuration:

- Maintenance configurations are associated with Azure VMs or Azure Dedicated Hosts. Make sure that VM/host resources exist in the new region before you begin.
- Identify:
 - Existing maintenance control configurations.

- The resource groups in which existing configurations currently reside.
- The resource groups to which the configurations will be added after moving to the new region.
- The resources associated with the maintenance configuration you want to move.
- Check that the resources in the new region are the same as those associated with the current maintenance configurations. The configurations can have the same names in the new region as they did in the old, but this isn't required.

Prepare and move

1. Retrieve all of the maintenance configurations in each subscription. Run the CLI [az maintenance configuration list](#) command to do this, replacing \$subId with your subscription ID.

Copy

```
az maintenance configuration list --subscription $subId --query
"[*].{Name:name, Location:location, ResGroup:resourceGroup}" --output
table
```

2. Review the returned table list of configuration records within the subscription. Here's an example. Your list will contain values for your specific environment.

Name	Location	Resource group
Skip Maintenance	eastus2	configuration-resource-group
IgniteDemoConfig	eastus2	configuration-resource-group
defaultMaintenanceConfiguration-eastus	eastus	test-configuration

3. Save your list for reference. As you move the configurations, it helps you to verify that everything's been moved.
4. As a reference, map each configuration/resource group to the new resource group in the new region.
5. Create new maintenance configurations in the new region using [PowerShell](#), or [CLI](#).
6. Associate the configurations with the resources in the new region, using [PowerShell](#), or [CLI](#).

Verify the move

After moving the configurations, compare configurations and resources in the new region with the table list you prepared.

Clean up source resources

After the move, consider deleting the moved maintenance configurations in the source region, [PowerShell](#), or [CLI](#).

Move managed identity for Azure resources across regions

- Article
- 05/25/2023
- 2 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare and move](#)
3. [Verify](#)
4. [Clean up](#)
5. [Next steps](#)

There are situations in which you'd want to move your existing user-assigned managed identities from one region to another. For example, you may need to move a solution that uses user-assigned managed identities to another region. You may also want to move an existing identity to another region as part of disaster recovery planning, and testing.

Moving User-assigned managed identities across Azure regions isn't supported. You can however, recreate a user-assigned managed identity in the target region.

Prerequisites

- Permissions to list permissions granted to existing user-assigned managed identity.
- Permissions to grant a new user-assigned managed identity the required permissions.
- Permissions to assign a new user-assigned identity to the Azure resources.

- Permissions to edit Group membership, if your user-assigned managed identity is a member of one or more groups.

Prepare and move

1. Copy user-assigned managed identity assigned permissions. You can list [Azure role assignments](#) but that may not be enough depending on how permissions were granted to the user-assigned managed identity. You should confirm that your solution doesn't depend on permissions granted using a service specific option.
2. Create a [new user-assigned managed identity](#) at the target region.
3. Grant the managed identity the same permissions as the original identity that it's replacing, including Group membership. You can review [Assign Azure roles to a managed identity](#) and [Group membership](#).
4. Specify the new identity in the properties of the resource instance that uses the newly created user assigned managed identity.

Verify

After reconfiguring your service to use your new managed identities in the target region, you need to confirm that all operations have been restored.

Clean up

Once that you confirm your service is back online, you can proceed to delete any resources in the source region that you no longer use.

Move between regions

- Article
- 06/02/2023
- 6 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Prepare](#)
3. [Request a region move](#)
4. [During the region move](#)

Show 2 more

Important

At this time, we only offer the tenant migration service to customers with Power BI Premium capacities. We currently don't support tenant migration for Fabric.

The location selected during sign-up determines your default data region. However, this region might not be optimal if most of your users are located in a different geographic location. You might want to move to another region to reduce latency or to ensure data governance. You can't move your organization's tenant between regions by yourself. Self-service migration of Power BI resources stored in Azure isn't supported. If you need to change your default data location from the current region to another region, you have to contact Microsoft support to manage the migration for you.

Caution

This article describes how to request a move between regions and keep Power BI or data. Be sure you're aware of what can't be moved and steps you have to do before and after the region move. Moving between regions is considered a tenant migration. You can request a different process to move your tenant to another region if data loss and reconfiguration is acceptable. To determine your current data region, follow the steps in [Find the default region for your organization](#).

Prerequisites

- The person who requests the data region move must be assigned the global administrator role. You can learn more about the different admin roles and what they can do in [Understanding Power BI administration roles](#). We can't help identify your global administrator for you. Look for global administrator role holders in Microsoft 365 or Azure Active Directory or ask your help desk.
- We must receive written approval confirming your awareness and agreement of the effect of the tenant migration on your organization.
- Provide a point of contact for after business hours during the migration.

Prepare

The migration process moves all tenant data to the new region. The GUID assigned to datasets, reports, dashboards, and other content don't change. However, there are some limitations you should be aware of and some preparation steps you need to take.

Awareness

- **The end-to-end migration process may take up to six months.** We prioritize service reliability and deployment schedules can change, so we

may need to reschedule during migration at any time. We can't guarantee successful migration due to inconsistent data or bugs.

- During the migration process, it's possible to encounter unforeseen issues that may result in multiple failures. Allow for multiple attempts to ensure a successful migration.
- Migration requires about six hours of down time. During migration, users can't access Power BI and an error message similar to the one shown in the following screenshot is displayed. The actual down time depends on the volume of data to be migrated.

- Capacities and Premium workspaces can't be migrated.
- Power BI Premium Per User (PPU) capacity will be deleted before migration starts. After the migration, PPU capacity will be recreated at first PPU user sign-in. For more information about PPU licenses, see [Power BI Premium Per User](#).
- After migration, Excel workbooks that use the Analyze in Excel feature might fail to refresh. You might need to update the connection string or redownload the ODC connection for that dataset. Follow the steps in [Start in Power BI with Analyze in Excel](#) if necessary.
- Push datasets might not be migrated. If they can't be migrated, you need to delete the datasets.
- You have to reconfigure data gateways after migration. To learn more about this step, read [Migrate, restore, or take over an on-premises data gateway](#).
- Dataset and workspace storage modes shouldn't be changed one day before the migration. Changing the storage mode before the migration can leave the datasets unusable after the migration. For more information, read [Dataset modes in the Power BI service](#) and [Manage data storage in Power BI workspaces](#).
- Some usage data collected before migration is unavailable after migration. Usage data in the following sources will be lost:
 - [Power BI Activity Log](#)
 - View count in [Lineage view](#)
 - [Data protection metrics report](#)
 - [Usage metrics\(preview\)](#)

Preparation steps

Our support team works with you to verify that the following steps are done to prepare for the migration:

- We can't migrate capacities and Premium workspaces, so you have to delete all capacities before migration. After the region move, these resources can be recreated. If you move resources from a Premium workspace to a shared workspace, datasets larger than 1 GB can't be viewed until they're moved back to Premium capacity.
 - Gateways should be deleted in the target region to avoid conflicts during migration.
- To keep user activity logs, follow the steps in [Track user activities in Power BI](#). You can get log data from either the Power BI activity log or the Unified audit log.
- All Fabric items have to be deleted individually. Deleting the capacity or workspaces deletes the Fabric's items immediately.
- All Fabric Trial capacities have to be deleted before the migration. To delete Fabric trial capacities, the users who own the trial capacity need to cancel the trial from the Me Profile at least one and a half days before the migration.

Request a region move

To find out the best way to contact support, read [Power BI support options](#). Most admins can use the **Help + support** experience in the [Power Platform Admin Center](#) to submit a service request. Use the following steps to get started:

1. Go to [Power Platform Admin Center Help + support](#) and sign in using admin credentials.
2. Select **New support request**, then select the following options to request a region move:
 - Product: Power BI
 - Tell us what you need help with: Move to a different region
 - Problem type: Administration
 - Problem Subtype: Tenant Management
 - Are you contacting us to move your tenant to another region: Yes

Select **See solutions** to move to the next screen.

3. Select **Next** to continue to **Select your support plan**. Choose your support plan. Add a description and include the information in the following table:

Information needed	How to find the information
Tenant object ID	How to find your Azure Active Directory tenant ID
Current region	Find the default region for your organization
Proposed region	International availability of Microsoft Power Platform
Proposed date and time for migration	Give us three options in UTC time. The proposed dates should be at least two weeks later than when you submit the request.
Contact available after during off-business hours	Name, phone number, and email address

4. Under **Is the problem you're reporting related to a recent service change?**, choose N/A. Select a severity level, then select **Next**.
5. Add your contact information, then **Submit**.

Our support team will be in touch. The support team makes sure you're authorized to make this request, confirms your awareness of the issues listed earlier, and obtains written approval to confirm you want to move your tenant between regions.

Be sure to provide contact details for someone who can act as the point of contact for Support. The contact has to be available after business hours.

Support reviews the submitted information, including your tenant object ID, current data region, and target data region. After details are confirmed, we coordinate the proposed migration time frame with you.

During the region move

- Don't do any manual or scheduled refreshes until after migration is complete.
- Support copies your data to the new region. Power BI isn't available to users during the move.

After the region move

When migration is complete, you can access Power BI in about 20-30 minutes. Support contacts you to make sure everything is working.

Do the following steps to recreate the configuration of the original region:

1. Recreate capacities and move workspaces back to Premium. Read more about this step in [Configure and manage capacities in Power BI Premium](#).
2. If push datasets were deleted, recreate them. For more information, see [Real-time streaming in Power BI](#) to learn how to push data into a dataset.

3. Reconfigure your data gateways. Follow the steps in [Migrate, restore, or take over an on-premises data gateway](#).
4. Excel workbooks that use the Analyze in Excel feature might fail to refresh. You might need to update the connection string or redownload the ODC connection for that dataset. Follow the steps in [Start in Power BI with Analyze in Excel](#) if necessary.
5. Links to Power BI that are embedded in content might fail to connect when migration is complete. For example, an embedded link in SharePoint might result in a user error. To resolve this problem, you have to regenerate the embedded link in Power BI, and then update the locations where they're used. To fix this issue, follow the procedure in [Embed a report web part in SharePoint Online](#).

To verify that the default region for data storage has been moved, follow the steps in [Find the default region for your organization](#).

Frequently asked questions

Can I migrate back to the original region? If yes, what's the process and will I lose data?

No, you can't revert to using the old region.

Is my data deleted immediately from the old region? If not, how long is it kept and do I have access to it?

Data is retained in the old region for 30 days and is then deleted. Customers don't have access to data in the old region after migration.

What happens to my Microsoft 365 groups, SharePoint sites, etc.? Are they also migrated?

We only migrate Power BI-specific resources. Your Microsoft 365 groups and SharePoint sites aren't touched.

Can I request that some of my data be migrated to a different region?

No, migration of data to different regions isn't a supported scenario.

Does migration change any of my data or settings for Azure Active Directory?

No, migration doesn't affect anything outside of Power BI.

Can I use Power BI REST APIs for read-only operations during migration?

No, using Power BI during tenant migration activity isn't recommended.

Why do I need to provide three proposed migration dates?

We need to ensure that migration happens outside of the production deployment window. This time-frame is subject to change on a weekly basis. We can only confirm the actual migration date five days before the migration.

Can I request migration during weekdays (if my company allows) or on any public holiday recognized by my organization?

Yes, you can request migration during weekdays or public holidays.

How do I verify my data is now stored in the requested region?

Follow the steps in [Find where data is stored](#). You should see the new region next to **Your data is stored in**.

Can I migrate or merge my Power BI tenant into a different tenant (for example, because of a company merger)?

No, migration from one tenant to another isn't possible.

After migration, is it normal to still see some refreshes happening from the old tenant location?

Refresh in the old region should stop after migration.

My allowlist contains Power BI IP ranges that are used to access some data sources. Do I need to update the IP ranges to match the new location?

Yes. Because it's a new location, the IP ranges are also changing, and the ranges need to be updated. [Download the Azure IP Ranges JSON file](#) to identify the needed IP ranges.

Is there a cost to have my tenant moved to a different region?

No, there's no cost charged for region migration. Customers that have any paid licenses can migrate. A global administrator must request the operation.

Move a Recovery Services vault and Azure Site Recovery configuration to another Azure region

- Article
- 11/03/2022
- 6 contributors

Feedback

In this article

1. [Prerequisites](#)
2. [Identify the resources that were used by Azure Site Recovery](#)
3. [Disable the existing disaster recovery configuration](#)
4. [Delete the resources](#)

Show 2 more

There are various scenarios in which you might want to move your existing Azure resources from one region to another. Examples are for manageability, governance reasons, or because of company mergers and acquisitions. One of the related resources you might want to move when you move your Azure VMs is the disaster recovery configuration.

There's no first-class way to move an existing disaster recovery configuration from one region to another. This is because you configured your target region based on your source VM region. When you decide to change the source region, the previously existing configurations of the target region can't be reused and must be reset. This article defines the step-by-step process to reconfigure the disaster recovery setup and move it to a different region.

In this document, you will:

- Verify prerequisites for the move.
- Identify the resources that were used by Azure Site Recovery.
- Disable replication.
- Delete the resources.
- Set up Site Recovery based on the new source region for the VMs.

Important

Currently, there's no first-class way to move a Recovery Services vault and the disaster recovery configuration as is to a different region. This article guides you through the process of disabling replication and setting it up in the new region.

Prerequisites

- Make sure that you remove and delete the disaster recovery configuration before you try to move the Azure VMs to a different region.

Note

If your new target region for the Azure VM is the same as the disaster recovery target region, you can use your existing replication configuration and move it. Follow the steps in [**Move Azure IaaS VMs to another Azure region**](#).

- Ensure that you're making an informed decision and that stakeholders are informed. Your VM won't be protected against disasters until the move of the VM is complete.

Identify the resources that were used by Azure Site Recovery

We recommend that you do this step before you proceed to the next one. It's easier to identify the relevant resources while the VMs are being replicated.

For each Azure VM that's being replicated, go to **Protected Items > Replicated Items > Properties** and identify the following resources:

- Target resource group
- Cache storage account
- Target storage account (in case of an unmanaged disk-based Azure VM)
- Target network

Disable the existing disaster recovery configuration

1. Go to the Recovery Services vault.
2. In **Protected Items > Replicated Items**, right-click the machine and select **Disable replication**.
3. Repeat this step for all the VMs that you want to move.

Note

The mobility service won't be uninstalled from the protected servers. You must uninstall it manually. If you plan to protect the server again, you can skip uninstalling the mobility service.

Delete the resources

1. Go to the Recovery Services vault.
2. Select **Delete**.
3. Delete all the other resources you [previously identified](#).

Move Azure VMs to the new target region

Follow the steps in these articles based on your requirement to move Azure VMs to the target region:

- [Move Azure VMs to another region](#)
- [Move Azure VMs into Availability Zones](#)

Set up Site Recovery based on the new source region for the VMs

Configure disaster recovery for the Azure VMs that were moved to the new region by following the steps in [Set up disaster recovery for Azure VMs](#).

Move a SQL Server VM to another region within Azure with Azure Site Recovery

- Article
- 03/30/2023
- 10 contributors

Feedback

In this article

1. [Verify prerequisites](#)
2. [Prepare to move](#)
3. [Configure Azure Site Recovery vault](#)
4. [Test move process](#)

Show 3 more

Applies to: SQL Server on Azure VM

This article teaches you how to use Azure Site Recovery to migrate your SQL Server virtual machine (VM) from one region to another within Azure.

Moving a SQL Server VM to a different region requires doing the following:

1. [Preparing](#): Confirm that both your source SQL Server VM and target region are adequately prepared for the move.
2. [Configuring](#): Moving your SQL Server VM requires that it is a replicated object within the Azure Site Recovery vault. You need to add your SQL Server VM to the Azure Site Recovery vault.
3. [Testing](#): Migrating the SQL Server VM requires failing it over from the source region to the replicated target region. To ensure that the move process will succeed, you need to first test that your SQL Server VM can successfully fail over to the target region. This will help expose any issues and avoid them when performing the actual move.
4. [Moving](#): Once your test failover passed, and you know that you are safe to migrate your SQL Server VM, you can perform the move of the VM to the target region.
5. [Cleaning up](#): To avoid billing charges, remove the SQL Server VM from the vault, and any unnecessary resources that are left over in the resource group.

Verify prerequisites

- Confirm that moving from your source region to your target region [is supported](#).
- Review the [scenario architecture and components](#) as well as the [support limitations and requirements](#).
- Verify account permissions. If you created your free Azure account, you're the administrator of your subscription. If you're not the subscription administrator, work with the administrator to assign the permissions that you need. To enable replication for a VM and copy data using Azure Site Recovery, you must have:
 - Permissions to create a VM. The *Virtual Machine Contributor* built-in role has these permissions, which include:
 - Permissions to create a VM in the selected resource group.
 - Permissions to create a VM in the selected virtual network.
 - Permissions to write to the selected storage account.
 - Permissions to manage Azure Site Recovery operations.The *Site Recovery Contributor* role has all the permissions that are required to manage Site Recovery operations in a Recovery Services vault.
- Moving the [SQL virtual machines resource](#) is not supported. You need to [reinstall the SQL IaaS Agent extension](#) on the target region where you have planned your move. If you are moving your resources between subscriptions or tenants, make sure you've registered your [subscription with the resource](#)

[provider](#) before attempting to register your migrated SQL Server VM with the SQL IaaS Agent extension.

Prepare to move

Prepare both the source SQL Server VM and the target region for the move.

Prepare the source SQL Server VM

- Ensure that all the latest root certificates are on the SQL Server VM that you want to move. If the latest root certificates are not there, security constraints will prevent data copy to the target region.
- For Windows VMs, install all of the latest Windows updates on the VM, so that all the trusted root certificates are on the machine. In a disconnected environment, follow the standard Windows Update and certificate update process for your organization.
- For Linux VMs, follow the guidance provided by your Linux distributor to get the latest trusted root certificates and certificate revocation list on the VM.
- Make sure you're not using an authentication proxy to control network connectivity for the VMs that you want to move.
- If the VM that you're trying to move doesn't have access to the internet, or it's using a firewall proxy to control outbound access, check the requirements.
- Identify the source networking layout and all the resources that you're currently using. This includes but isn't limited to load balancers, network security groups (NSGs), and public IPs.

Prepare the target region

- Verify that your Azure subscription allows you to create VMs in the target region that's used for disaster recovery. Contact support to enable the required quota.
- Make sure that your subscription has enough resources to support VMs with size that match your source VMs. If you're using Site Recovery to copy data to the target, Site Recovery chooses the same size, or the closest possible size for the target VM.
- Make sure that you create a target resource for every component that's identified in the source networking layout. This step is important to ensure that your VMs have all the functionality and features in the target region that you had in the source region.
 - Azure Site Recovery automatically discovers and creates a virtual network when you enable replication for the source VM. You can also pre-create a network and assign it to the VM in the user flow for enabling replication. You need to manually create any other resources in the target region.

- To create the most commonly used network resources that are relevant for you based on the source VM configuration, see the following documentation:
 - [Network security groups](#)
 - [Load balancer](#)
 - [Public IP address](#)
 - For any additional networking components, see the [networking documentation](#).
- Manually create a non-production network in the target region if you want to test the configuration before you perform the final move to the target region. We recommend this step because it ensures minimal interference with the production network.

Configure Azure Site Recovery vault

The following steps show you how to use Azure Site Recovery to copy data to the target region. Create the Recovery Services vault in any region other than the source region.

1. Sign in to the [Azure portal](#).
2. Choose to **Create a resource** from the upper-left hand corner of the navigation pane.
3. Select **IT & Management tools** and then select **Backup and Site Recovery**.
4. On the **Basics** tab, under **Project details**, either create a new resource group in the target region, or select an existing resource group in the target region.
5. Under **Instance Details**, specify a name for your vault, and then select your target **Region** from the drop-down.
6. Select **Review + Create** to create your Recovery Services vault.
7. Select **All services** from the upper-left hand corner of the navigation pane and in the search box type recovery services.
8. (Optionally) Select the star next to **Recovery Services vaults** to add it to your quick navigation bar.
9. Select **Recovery services vaults** and then select the Recovery Services vault you created.
10. On the **Overview** pane, select **Replicate**.

11. Select **Source** and then select **Azure** as the source. Select the appropriate values for the other drop-down fields, such as the location for your source VMs. Only resources groups located in the **Source location** region will be visible in the **Source resource group** field.

12. Select **Virtual machines** and then choose the virtual machines you want to migrate. Select **OK** to save your VM selection.
13. Select **Settings**, and then choose your **Target location** from the dropdown. This should be the resource group you prepared earlier.
14. Once you have customized replication, select **Create target resources** to create the resources in the new location.
15. Once resource creation is complete, select **Enable replication** to start replication of your SQL Server VM from the source to the target region.
16. You can check the status of replication by navigating to your recovery vault, selecting **Replicated items** and viewing the **Status** of your SQL Server VM. A status of **Protected** indicates that replication has completed.

Test move process

The following steps show you how to use Azure Site Recovery to test the move process.

1. Navigate to your **Recovery Services vault** in the [Azure portal](#) and select **Replicated items**.
2. Select the SQL Server VM you would like to move, verify that the **Replication Health** shows as **Healthy** and then select **Test Failover**.
3. On the **Test Failover** page, select the **Latest app-consistent** recovery point to use for the failover, as that is the only type of snapshot that can guarantee SQL Server data consistency.
4. Select the virtual network under **Azure virtual network** and then select **OK** to test failover.

Important

We recommend that you use a separate Azure VM network for the failover test. Don't use the production network that was set up when you enabled replication and that you want to move your VMs into eventually.

5. To monitor progress, navigate to your vault, select **Site Recovery jobs** under **Monitoring**, and then select the **Test failover** job that's in progress.

6. Once the test completes, navigate to **Virtual machines** in the portal and review the newly created virtual machine. Make sure the SQL Server VM is running, is sized appropriately, and is connected to the appropriate network.
7. Delete the VM that was created as part of the test, as the **Failover** option will be grayed out until the failover test resources are cleaned up. Navigate back to the vault, select **Replicated items**, select the SQL Server VM, and then select **Cleanup test failover**. Record and save any observations associated with the test in the **Notes** section and select the checkbox next to **Testing is complete. Delete test failover virtual machines**. Select **OK** to clean up resources after the test.

Move the SQL Server VM

The following steps show you how to move the SQL Server VM from your source region to your target region.

1. Navigate to the **Recovery Services** vault, select **Replicated items**, select the VM, and then select **Failover**.
2. Select the **latest app-consistent** recover point under **Recovery Point**.
3. Select the check box next to **Shut down the machine before beginning failover**. Site Recovery will attempt to shut down the source VM before triggering the failover. Failover will continue even if shut down fails.
4. Select **OK** to start the failover.
5. You can monitor the failover process from the same **Site Recovery jobs** page you viewed when monitoring the failover test in the previous section.
6. After the job completes, check that the SQL Server VM appears in the target region as expected.
7. Navigate back to the vault, select **Replicated Items**, select the SQL Server VM, and select **Commit** to finish the move process to the target region. Wait until the commit job finishes.
8. Register your SQL Server VM with the SQL IaaS Agent extension to enable **SQL virtual machine** manageability in the Azure portal and

features associated with the extension. For more information, see [Register SQL Server VM with the SQL IaaS Agent extension](#).

Warning

SQL Server data consistency is only guaranteed with app-consistent snapshots. The **latest processed** snapshot can't be used for SQL Server failover as a crash recovery snapshot can't guarantee SQL Server data consistency.

Clean up source resources

To avoid billing charges, remove the SQL Server VM from the vault, and delete any unnecessary associated resources.

1. Navigate back to the **Site Recovery** vault, select **Replicated items**, and select the SQL Server VM.
2. Select **Disable Replication**. Select a reason for disabling protection, and then select **OK** to disable replication.

Important

It is important to perform this step to avoid being charged for Azure Site Recovery replication.

3. If you have no plans to reuse any of the resources in the source region, delete all relevant network resources, and corresponding storage accounts.

Use portal to create private link for managing Azure resources

- Article
- 08/04/2022
- 3 contributors

Feedback

In this article

1. [Understand architecture](#)
2. [Workflow](#)
3. [Required permissions](#)
4. [Create resource management private link](#)

Show 3 more

This article explains how you can use [Azure Private Link](#) to restrict access for managing resources in your subscriptions. It shows using the Azure portal for setting up management of resources through private access.

Private links enable you to access Azure services over a private endpoint in your virtual network. When you combine private links with Azure Resource Manager's operations, you block users who aren't at the specific endpoint from managing resources. If a malicious user gets credentials to an account in your subscription, that user can't manage the resources without being at the specific endpoint.

Private link provides the following security benefits:

- **Private Access** - users can manage resources from a private network via a private endpoint.

Note

Azure Kubernetes Service (AKS) currently doesn't support the ARM private endpoint implementation.

Azure Bastion doesn't support private links. It is recommended to use a private DNS zone for your resource management private link private endpoint configuration, but due to the overlap with the management.azure.com name, your Bastion instance will stop working. For more information, view [Azure Bastion FAQ](#).

Understand architecture

For this release, you can only apply private link management access at the level of the root [management group](#). This limitation means private link access is applied across your tenant.

There are two resource types you'll use when implementing management through a private link.

- Resource management private link
(Microsoft.Authorization/resourceManagementPrivateLinks)
- Private link association (Microsoft.Authorization/privateLinkAssociations)

The following image shows how to construct a solution that restricts access for managing resources.

The private link association extends the root management group. The private link association and the private endpoints reference the resource management private link.

Important

Multi-tenant accounts aren't currently supported for managing resources through a private link. You can't connect private link associations on different tenants to a single resource management private link.

If your account accesses more than one tenant, define a private link for only one of them.

Workflow

To set up a private link for resources, use the following steps. The steps are described in greater detail later in this article.

1. Create the resource management private link.
2. Create a private link association. The private link association extends the root management group. It also references the resource ID for the resource management private link.
3. Add a private endpoint that references the resource management private link.

After completing those steps, you can manage Azure resources that are within the hierarchy of the scope. You use a private endpoint that is connected to the subnet.

You can monitor access to the private link. For more information, see [Logging and monitoring](#).

Required permissions

To set up the private link for resource management, you need the following access:

- Owner on the subscription. This access is needed to create resource management private link resource.
- Owner or Contributor at the root management group. This access is needed to create the private link association resource.
- The Global Administrator for the Azure Active Directory doesn't automatically have permission to assign roles at the root management group. To enable creating resource management private links, the Global Administrator must have permission to read root management group and [elevate access](#) to have User Access Administrator permission on all subscriptions and management groups in the tenant. After you get the User Access Administrator permission,

the Global Administrator must grant Owner or Contributor permission at the root management group to the user creating the private link association.

Create resource management private link

When you create a resource management private link, the private link association is automatically created for you.

1. In the [portal](#), search for **Resource management private links** and select it from the available options.
2. If your subscription doesn't already have resource management private links, you'll see a blank page. Select **Create resource management private link**.
3. Provide values for the new resource management private link. The root management group for the directory you selected is used for the new resource. Select **Review + create**.
4. After validation passes, select **Create**.

Create private endpoint

Now, create a private endpoint that references the resource management private link.

1. Navigate to the **Private Link Center**. Select **Create private endpoint**.
2. In the **Basics** tab, provide values for your private endpoint.

3. In the **Resource** tab, select **Connect to an Azure resource in my directory**. For resource type, select **Microsoft.Authorization/resourceManagementPrivateLinks**. For target subresource, select **ResourceManagement**.
4. In the **Configuration** tab, select your virtual network. We recommend integrating with a private DNS zone. Select **Review + create**.
5. After validation passes, select **Create**.

Verify private DNS zone

To make sure your environment is properly configured, check the local IP address for the DNS zone.

1. In the resource group where you deployed the private endpoint, select the private DNS zone resource named **privatelink.azure.com**.
2. Verify that the record set named **management** has a valid local IP address.

Next steps

To learn more about private links, see [Azure Private Link](#).

Use APIs to create a private link for managing Azure resources

- Article
- 06/25/2022
- 2 contributors

Feedback

In this article

1. [Understand architecture](#)
2. [Workflow](#)
3. [Required permissions](#)

4. [Create resource management private link](#)

Show 3 more

This article explains how you can use [Azure Private Link](#) to restrict access for managing resources in your subscriptions.

Private links enable you to access Azure services over a private endpoint in your virtual network. When you combine private links with Azure Resource Manager's operations, you block users who aren't at the specific endpoint from managing resources. If a malicious user gets credentials to an account in your subscription, that user can't manage the resources without being at the specific endpoint.

Private link provides the following security benefits:

- **Private Access** - users can manage resources from a private network via a private endpoint.

Note

Azure Kubernetes Service (AKS) currently doesn't support the ARM private endpoint implementation.

Azure Bastion doesn't support private links. It is recommended to use a private DNS zone for your resource management private link private endpoint configuration, but due to the overlap with the management.azure.com name, your Bastion instance will stop working. For more information, view [Azure Bastion FAQ](#).

Understand architecture

For this release, you can only apply private link management access at the level of the root [management group](#). This limitation means private link access is applied across your tenant.

There are two resource types you'll use when implementing management through a private link.

- Resource management private link
(Microsoft.Authorization/resourceManagementPrivateLinks)
- Private link association (Microsoft.Authorization/privateLinkAssociations)

The following image shows how to construct a solution that restricts access for managing resources.

The private link association extends the root management group. The private link association and the private endpoints reference the resource management private link.

Important

Multi-tenant accounts aren't currently supported for managing resources through a private link. You can't connect private link associations on different tenants to a single resource management private link.

If your account accesses more than one tenant, define a private link for only one of them.

Workflow

To set up a private link for resources, use the following steps. The steps are described in greater detail later in this article.

1. Create the resource management private link.
2. Create a private link association. The private link association extends the root management group. It also references the resource ID for the resource management private link.
3. Add a private endpoint that references the resource management private link.

After completing those steps, you can manage Azure resources that are within the hierarchy of the scope. You use a private endpoint that is connected to the subnet.

You can monitor access to the private link. For more information, see [Logging and monitoring](#).

Required permissions

To set up the private link for resource management, you need the following access:

- Owner on the subscription. This access is needed to create resource management private link resource.
- Owner or Contributor at the root management group. This access is needed to create the private link association resource.
- The Global Administrator for the Azure Active Directory doesn't automatically have permission to assign roles at the root management group. To enable creating resource management private links, the Global Administrator must have permission to read root management group and [elevate access](#) to have User Access Administrator permission on all subscriptions and management groups in the tenant. After you get the User Access Administrator permission,

the Global Administrator must grant Owner or Contributor permission at the root management group to the user creating the private link association.

Create resource management private link

To create resource management private link, send the following request:

- [Azure CLI](#)
- [PowerShell](#)
- [REST](#)

Example

Azure CLICopy

```
# Login first with az login if not using Cloud Shell
az resourcemanagement private-link create --location WestUS --resource-group
PrivateLinkTestRG --name NewRMPL --public-network-access enabled
```

Note the ID that is returned for the new resource management private link. You'll use it for creating the private link association.

Create private link association

The resource name of a private link association resource must be a GUID, and it isn't yet supported to disable the publicNetworkAccess field.

To create the private link association, use:

- [Azure CLI](#)
- [PowerShell](#)
- [REST](#)

Example

Azure CLICopy

```
# Login first with az login if not using Cloud Shell
az private-link association create --management-group-id fc096d27-0434-4460-a3ea-
110df0422a2d --name 1d7942d1-288b-48de-8d0f-2d2aa8e03ad4 --privatelink
"/subscriptions/00000000-0000-0000-0000-
0000000000/resourceGroups/PrivateLinkTestRG/providers/Microsoft.Authorization/re
sourceManagementPrivateLinks/newRMPL"
```

Add private endpoint

This article assumes you already have a virtual network. In the subnet that will be used for the private endpoint, you must turn off private endpoint network policies. If you haven't turned off private endpoint network policies, see [Disable network policies for private endpoints](#).

To create a private endpoint, see Private Endpoint documentation for creating via [Portal](#), [PowerShell](#), [CLI](#), [Bicep](#), or [template](#).

In the request body, set the `privateServiceLinkId` to the ID from your resource management private link. The `groupIds` must contain `ResourceManagement`. The location of the private endpoint must be the same as the location of the subnet.

JSONCopy

```
{
  "location": "westus2",
  "properties": {
    "privateLinkServiceConnections": [
      {
        "name": "{connection-name}",
        "properties": {
          "privateLinkServiceId": "/subscriptions/{subID}/resourceGroups/{rgName}/providers/Microsoft.Authorization/resourceManagementPrivateLinks/{name}",
          "groupIds": [
            "ResourceManagement"
          ]
        }
      }
    ],
    "subnet": {
      "id": "/subscriptions/{subID}/resourceGroups/{rgName}/providers/Microsoft.Network/virtualNetworks/{vnet-name}/subnets/{subnet-name}"
    }
  }
}
```

The next step varies depending whether you're using automatic or manual approval. For more information about approval, see [Access to a private link resource using approval workflow](#).

The response includes approval state.

JSONCopy

```
"privateLinkServiceConnectionState": {
  "actionsRequired": "None",
  "description": "",
  "status": "Approved"
```

},

If your request is automatically approved, you can continue to the next section. If your request requires manual approval, wait for the network admin to approve your private endpoint connection.

Next steps

To learn more about private links, see [Azure Private Link](#).

Manage resource management private links

- Article
- 03/09/2023
- 2 contributors

Feedback

In this article

1. [Resource management private links](#)
2. [Private link association](#)
3. [Next steps](#)

This article explains how you to work with existing resource management private links. It shows API operations for getting and deleting existing resources.

If you need to create a resource management private link, see [Use portal to create private link for managing Azure resources](#) or [Use APIs to create private link for managing Azure resources](#).

Resource management private links

To **get a specific** resource management private link, send the following request:

- [Azure CLI](#)
- [PowerShell](#)
- [REST](#)

Example

Azure CLICopy

```
# Login first with az login if not using Cloud Shell
az resourcemanagement private-link show --resource-group PrivateLinkTestRG --name
NewRMPL
```

To **get all** resource management private links in a subscription, use:

- [Azure CLI](#)
- [PowerShell](#)
- [REST](#)

Azure CLICopy

```
# Login first with az login if not using Cloud Shell
az resourcemanagement private-link list
```

To **delete a specific** resource management private link, use:

- [Azure CLI](#)
- [PowerShell](#)
- [REST](#)

Example

Azure CLICopy

```
# Login first with az login if not using Cloud Shell
az resourcemanagement private-link delete --resource-group PrivateLinkTestRG --
name NewRMPL
```

Private link association

To **get a specific** private link association for a management group, use:

- [Azure CLI](#)
- [PowerShell](#)
- [REST](#)

Example

Azure CLICopy

```
# Login first with az login if not using Cloud Shell
az private-link association show --management-group-id fc096d27-0434-4460-a3ea-
110df0422a2d --name 1d7942d1-288b-48de-8d0f-2d2aa8e03ad4
```

To **delete** a private link association, use:

- [Azure CLI](#)
- [PowerShell](#)
- [REST](#)

Example

Azure CLICopy

```
# Login first with az login if not using Cloud Shell
az private-link association delete --management-group-id 24f15700-370c-45bc-86a7-
aee1b0c4eb8a --name 1d7942d1-288b-48de-8d0f-2d2aa8e03ad4
```

Next steps

- To learn more about private links, see [Azure Private Link](#).
- To manage your private endpoints, see [Manage Private Endpoints](#).
- To create a resource management private links, see [Use portal to create private link for managing Azure resources](#) or [Use REST API to create private link for managing Azure resources](#).

Azure resource providers and types

- Article
- 04/09/2023
- 4 contributors

Feedback

In this article

1. [Register resource provider](#)
2. [Azure portal](#)
3. [Azure PowerShell](#)
4. [Azure CLI](#)
5. [Next steps](#)

An Azure resource provider is a collection of REST operations that provide functionality for an Azure service. For example, the Key Vault service consists of a resource provider named **Microsoft.KeyVault**. The resource provider defines [REST operations](#) for working with vaults, secrets, keys, and certificates.

The resource provider defines the Azure resources that are available for you to deploy to your account. The name of a resource type is in the format: {resource-

{provider}/{resource-type}. The resource type for a key vault is **Microsoft.KeyVault/vaults**.

In this article, you learn how to:

- View all resource providers in Azure
- Check registration status of a resource provider
- Register a resource provider
- View resource types for a resource provider
- View valid locations for a resource type
- View valid API versions for a resource type

You can do these steps through the Azure portal, Azure PowerShell, or Azure CLI.

For a list that maps resource providers to Azure services, see [Resource providers for Azure services](#).

Register resource provider

Before using a resource provider, your Azure subscription must be registered for the resource provider. Registration configures your subscription to work with the resource provider.

Important

Only register a resource provider when you're ready to use it. The registration step enables you to maintain least privileges within your subscription. A malicious user can't use resource providers that aren't registered.

Some resource providers are registered by default. For a list of resource providers registered by default, see [Resource providers for Azure services](#).

Other resource providers are registered automatically when you take certain actions. When you create a resource through the portal, the resource provider is typically registered for you. When you deploy an Azure Resource Manager template or Bicep file, resource providers defined in the template are registered automatically. However, if a resource in the template creates supporting resources that aren't in the template, such as monitoring or security resources, you need to manually register those resource providers.

For other scenarios, you may need to manually register a resource provider.

Important

Your application code **shouldn't block the creation of resources** for a resource provider that is in the **registering** state. When you register the resource provider, the operation is done individually for each supported region. To create resources in a region, the registration only needs to be completed in that region. By not blocking a resource provider in the registering state, your application can continue much sooner than waiting for all regions to complete.

You must have permission to do the `/register/action` operation for the resource provider. The permission is included in the Contributor and Owner roles.

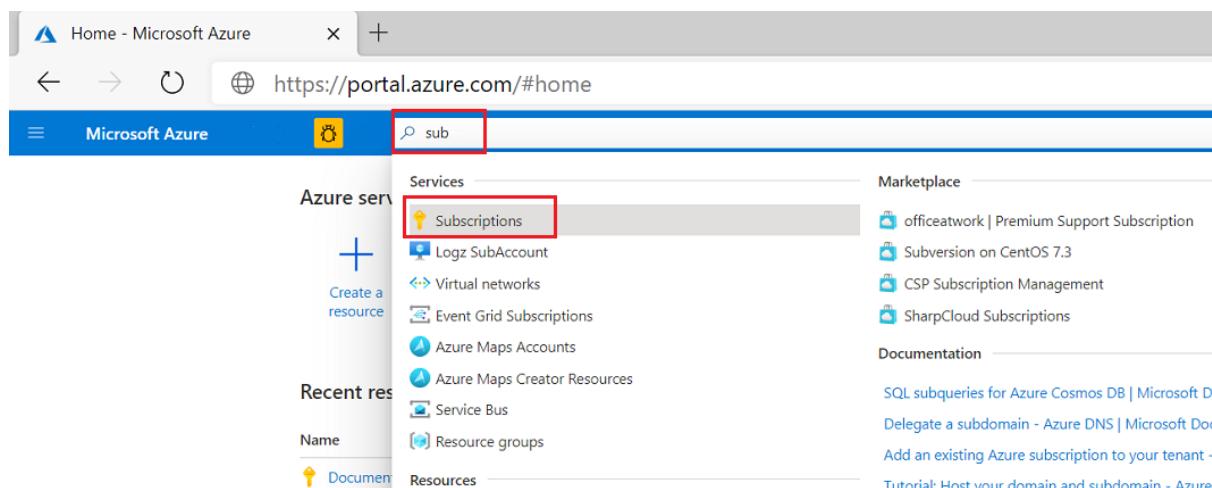
You can't unregister a resource provider when you still have resource types from that resource provider in your subscription.

Azure portal

Register resource provider

To see all resource providers, and the registration status for your subscription:

1. Sign in to the [Azure portal](#).
2. On the Azure portal menu, search for **Subscriptions**. Select it from the available options.



3. Select the subscription you want to view.

The screenshot shows the Microsoft Azure Subscriptions page. At the top, there's a header bar with the Microsoft Azure logo, a search bar, and a 'Home >' link. Below the header, the title 'Subscriptions' is displayed, along with a 'Microsoft' logo and a 'Add' button. A note says 'View list of subscriptions for which you have role-based access control (RBAC) permissions to manage Azure resources.' It also mentions 'Showing subscriptions in Microsoft directory. Don't see a subscription? [Switch directories](#)' and 'My role [\(i\)](#)'. There's a dropdown menu showing '8 selected'. Below this, there's an 'Apply' button. The main table lists two subscriptions:

Subscription name ↑↓	Subscription ID ↑↓
Documentation Testing 1	
Third Internal Consumption	

A red box highlights the 'Documentation Testing 1' row.

4. On the left menu, under **Settings**, select **Resource providers**.

Microsoft Azure ? Search resources, services, and docs (G+/)

Home > Subscriptions >

Documentation Testing 1

Subscription

Search (Ctrl+ /) Cancel subscription Rename Change directory Feedback

AUDIT RECOMMENDATIONS

Billing

Invoices

Settings

Programmatic deployment

Billing properties

Resource groups

Resources

Preview features

Usage + quotas

Policies

Management certificates

My permissions

Resource providers

Deployments

Resource locks

Support + troubleshooting

Essentials

Subscription ID :

Directory : Microsoft (microsoft.onmicrosoft.com)

Status : Active

Parent management group :

Costs by resource View details >

Resource Provider	Cost
demoaccountx7aiywacfkrro	0.00 USD
stage3a4176e058d34bb88cc	0.00 USD
tfstorers5ksicjjpn33m	0.00 USD
Others	0.00 USD

5. Find the resource provider you want to register, and select **Register**. To maintain least privileges in your subscription, only register those resource providers that you're ready to use.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, the breadcrumb navigation shows 'Home > Subscriptions > Documentation Testing 1'. The main title is 'Documentation Testing 1 | Resource providers'. On the left, a sidebar has sections like 'Subscription', 'Search (Ctrl+ /)', 'AU�ISOR RECOMMENDATIONS', 'Billing', 'Invoices', 'Settings', 'Programmatic deployment', 'Billing properties', 'Resource groups', 'Resources', 'Preview features', 'Usage + quotas', 'Policies', 'Management certificates', 'My permissions', 'Resource providers' (which is selected and highlighted in grey), 'Deployments', and 'Resource locks'. At the top right, there are buttons for 'Register', 'Unregister', and 'Refresh', with 'Register' being the one highlighted with a red box. Below these are 'Filter by name...' and a list of providers. The list includes: Microsoft.AzureStackHCI (NotRegistered), Microsoft.BareMetalInfrastructure (NotRegistered), Microsoft.BatchAI (NotRegistered), Microsoft.Billing (Registered), Microsoft.Bing (NotRegistered), Microsoft.BingMaps (NotRegistered), Microsoft.BlockchainTokens (NotRegistered), Microsoft.Blueprint (NotRegistered) - this row is also highlighted with a red box, Microsoft.Capacity (NotRegistered), Microsoft.Chaos (NotRegistered), Microsoft.ClassicInfrastructureMigrate (NotRegistered), and Microsoft.ClassicSubscription (Registered). The 'Status' column uses icons to indicate the registration status: a crossed-out circle for NotRegistered and a green checkmark for Registered.

Provider	Status
Microsoft.AzureStackHCI	NotRegistered
Microsoft.BareMetalInfrastructure	NotRegistered
Microsoft.BatchAI	NotRegistered
Microsoft.Billing	Registered
Microsoft.Bing	NotRegistered
Microsoft.BingMaps	NotRegistered
Microsoft.BlockchainTokens	NotRegistered
Microsoft.Blueprint	NotRegistered
Microsoft.Capacity	NotRegistered
Microsoft.Chaos	NotRegistered
Microsoft.ClassicInfrastructureMigrate	NotRegistered
Microsoft.ClassicSubscription	Registered

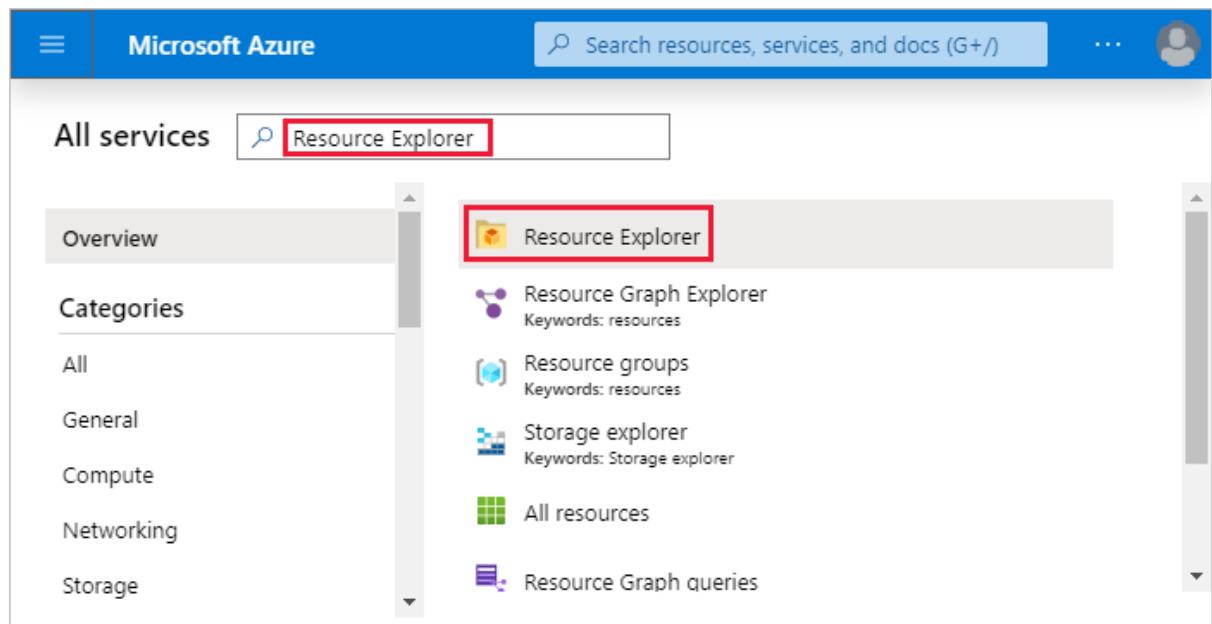
Important

As [noted earlier](#), don't block the creation of resources for a resource provider that is in the **registering** state. By not blocking a resource provider in the registering state, your application can continue much sooner than waiting for all regions to complete.

View resource provider

To see information for a particular resource provider:

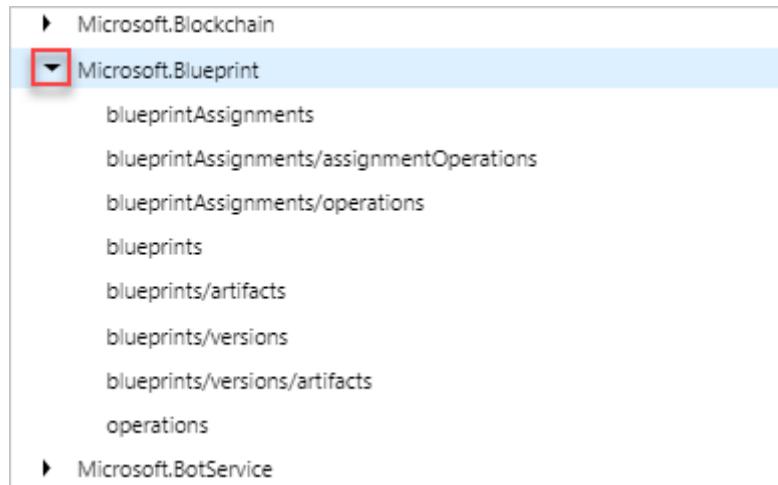
1. Sign in to the [Azure portal](#).
2. On the Azure portal menu, select **All services**.
3. In the **All services** box, enter **resource explorer**, and then select **Resource Explorer**.



4. Expand **Providers** by selecting the right arrow.

This screenshot shows the 'Resource Explorer' interface. At the top, it says 'Resource Explorer' and 'Microsoft'. Below that is a search bar with a magnifying glass icon and the placeholder 'Search...'. Underneath the search bar is a list of resource providers. The first item, 'Providers', is preceded by a red square with a white downward arrow, indicating it is expanded. The expanded list shows various provider names, each preceded by a red triangle icon. The providers listed are: 84codes.CloudAMQP, AppDynamics.APM, Aspera.Transfers, Auth0.Cloud, Citrix.Cloud, CloudSimple.PrivateCloudaaS, Conexlink.MyCloudIT, Crypteron.DataSecurity, and Dynatrace.DynatraceSaaS.

5. Expand a resource provider and resource type that you want to view.



6. Resource Manager is supported in all regions, but the resources you deploy might not be supported in all regions. Also, there may be limitations on your subscription that prevent you from using some regions that support the resource. The resource explorer displays valid locations for the resource type.

A screenshot of the Azure Resource Explorer interface. The left pane shows the 'Resource Explorer' sidebar with a search bar and a list of resource providers. 'Microsoft.Batch' is selected and highlighted with a red box. The right pane shows the details for 'Microsoft.Batch' with a response time of 439ms. The URL is '/providers/Microsoft.Batch?api-version=2014-04-01-preview'. The response body is a JSON object with numbered lines:

```
1  {
2      "namespace": "Microsoft.Batch",
3      "resourceTypes": [
4          {
5              "resourceType": "batchAccounts",
6              "locations": [
7                  "West Europe",
8                  "East US",
9                  "East US 2",
10                 "West US",
11                 "North Central US",
12                 "Brazil South",
13                 "North Europe",
14                 "Central US",

```

The 'locations' array is highlighted with a red box.

7. The API version corresponds to a version of REST API operations that are released by the resource provider. As a resource provider enables new features, it releases a new version of the REST API. The resource explorer displays valid API versions for the resource type.

The screenshot shows the Microsoft Resource Explorer interface. On the left, a tree view lists various Azure resource providers. The 'Microsoft.Batch' provider is selected and expanded, showing its resource types: batchAccounts, locations, locations/accountOperationResults, locations/checkNameAvailability, locations/quotas, and operations. On the right, the detailed configuration for the Microsoft.Batch provider is displayed in JSON format. A red box highlights the 'apiVersions' field, which contains a list of supported API versions from 2014-05-01-privatepreview to 2018-12-01.

```

{
  "namespace": "Microsoft.Batch",
  "resourceTypes": [
    {
      "resourceType": "batchAccounts",
      "locations": [ ... ],
      "apiVersions": [
        "2018-12-01",
        "2017-09-01",
        "2017-05-01",
        "2017-01-01",
        "2015-12-01",
        "2015-09-01",
        "2015-07-01",
        "2014-05-01-privatepreview"
      ],
      "apiProfiles": [
        ...
      ]
    }
  ]
}

```

Azure PowerShell

To see all resource providers in Azure, and the registration status for your subscription, use:

Azure PowerShellCopy

Open Cloudshell

```
Get-AzResourceProvider -ListAvailable | Select-Object ProviderNamespace, RegistrationState
```

The command returns:

OutputCopy

ProviderNamespace	RegistrationState
Microsoft.ClassicCompute	Registered
Microsoft.ClassicNetwork	Registered
Microsoft.ClassicStorage	Registered
Microsoft.CognitiveServices	Registered
...	

To see all registered resource providers for your subscription, use:

Azure PowerShellCopy

Open Cloudshell

```
Get-AzResourceProvider -ListAvailable | Where-Object RegistrationState -eq "Registered" | Select-Object ProviderNamespace, RegistrationState | Sort-Object ProviderNamespace
```

To maintain least privileges in your subscription, only register those resource providers that you're ready to use. To register a resource provider, use:

Azure PowerShellCopy

Open Cloudshell

```
Register-AzResourceProvider -ProviderNamespace Microsoft.Batch
```

The command returns:

OutputCopy

```
ProviderNamespace : Microsoft.Batch
RegistrationState : Registering
ResourceTypes     : {batchAccounts, operations, locations, locations/quotas}
Locations        : {West Europe, East US, East US 2, West US...}
```

Important

As **noted earlier**, **don't block the creation of resources** for a resource provider that is in the **registering** state. By not blocking a resource provider in the registering state, your application can continue much sooner than waiting for all regions to complete.

To see information for a particular resource provider, use:

Azure PowerShellCopy

Open Cloudshell

```
Get-AzResourceProvider -ProviderNamespace Microsoft.Batch
```

The command returns:

OutputCopy

```
{ProviderNamespace : Microsoft.Batch
RegistrationState : Registered
ResourceTypes     : {batchAccounts}
Locations        : {West Europe, East US, East US 2, West US...}}
```

...

To see the resource types for a resource provider, use:

Azure PowerShellCopy

Open Cloudshell

```
(Get-AzResourceProvider -ProviderNamespace Microsoft.Batch).ResourceTypes.ResourceTypeName
```

The command returns:

OutputCopy

```
batchAccounts  
operations  
locations  
locations/quotas
```

The API version corresponds to a version of REST API operations that are released by the resource provider. As a resource provider enables new features, it releases a new version of the REST API.

To get the available API versions for a resource type, use:

Azure PowerShellCopy

Open Cloudshell

```
((Get-AzResourceProvider -ProviderNamespace Microsoft.Batch).ResourceTypes |  
Where-Object ResourceTypeName -eq batchAccounts).ApiVersions
```

The command returns:

OutputCopy

```
2017-05-01  
2017-01-01  
2015-12-01  
2015-09-01  
2015-07-01
```

Resource Manager is supported in all regions, but the resources you deploy might not be supported in all regions. Also, there may be limitations on your subscription that prevent you from using some regions that support the resource.

To get the supported locations for a resource type, use.

Azure PowerShellCopy

Open Cloudshell

```
((Get-AzResourceProvider -ProviderNamespace Microsoft.Batch).ResourceTypes |  
Where-Object ResourceTypeName -eq batchAccounts).Locations
```

The command returns:

OutputCopy

```
West Europe
```

```
East US
East US 2
West US
...
```

Azure CLI

To see all resource providers in Azure, and the registration status for your subscription, use:

Azure CLICopy

Open Cloudshell

```
az provider list --query "[].{Provider:namespace, Status:registrationState}" --out table
```

The command returns:

OutputCopy

Provider	Status
Microsoft.ClassicCompute	Registered
Microsoft.ClassicNetwork	Registered
Microsoft.ClassicStorage	Registered
Microsoft.CognitiveServices	Registered
...	

To see all registered resource providers for your subscription, use:

Azure CLICopy

Open Cloudshell

```
az provider list --query
"sort_by([?registrationState=='Registered']).{Provider:namespace,
Status:registrationState}, &Provider)" --out table
```

To maintain least privileges in your subscription, only register those resource providers that you're ready to use. To register a resource provider, use:

Azure CLICopy

Open Cloudshell

```
az provider register --namespace Microsoft.Batch
```

The command returns a message that registration is on-going.

To see information for a particular resource provider, use:

Azure CLICopy

Open Cloudshell

```
az provider show --namespace Microsoft.Batch
```

The command returns:

OutputCopy

```
{  
  "id": "/subscriptions/#####-####/providers/Microsoft.Batch",  
  "namespace": "Microsoft.Batch",  
  "registrationsState": "Registering",  
  "resourceTypes": [  
    ...  
  ]  
}
```

Important

As [noted earlier](#), don't block the creation of resources for a resource provider that is in the **registering** state. By not blocking a resource provider in the registering state, your application can continue much sooner than waiting for all regions to complete.

To see the resource types for a resource provider, use:

Azure CLICopy

Open Cloudshell

```
az provider show --namespace Microsoft.Batch --query  
"resourceTypes[*].resourceType" --out table
```

The command returns:

OutputCopy

```
Result  
-----  
batchAccounts  
operations  
locations  
locations/quotas
```

The API version corresponds to a version of REST API operations that are released by the resource provider. As a resource provider enables new features, it releases a new version of the REST API.

To get the available API versions for a resource type, use:

Azure CLICopy

Open Cloudshell

```
az provider show --namespace Microsoft.Batch --query  
"resourceTypes[?resourceType=='batchAccounts'].apiVersions | [0]" --out table
```

The command returns:

OutputCopy

Result

```
-----  
2017-05-01  
2017-01-01  
2015-12-01  
2015-09-01  
2015-07-01
```

Resource Manager is supported in all regions, but the resources you deploy might not be supported in all regions. Also, there may be limitations on your subscription that prevent you from using some regions that support the resource.

To get the supported locations for a resource type, use.

Azure CLICopy

Open Cloudshell

```
az provider show --namespace Microsoft.Batch --query  
"resourceTypes[?resourceType=='batchAccounts'].locations | [0]" --out table
```

The command returns:

OutputCopy

Result

```
-----  
West Europe  
East US  
East US 2  
West US  
...
```

Azure Resource Manager resource group and resource deletion

- Article
- 04/10/2023
- 10 contributors

Feedback

In this article

1. [How order of deletion is determined](#)
2. [After deletion](#)
3. [Delete resource group](#)
4. [Delete resource](#)

Show 3 more

This article shows how to delete resource groups and resources. It describes how Azure Resource Manager orders the deletion of resources when you delete a resource group.

Note

This article was partially created with the help of artificial intelligence. Before publishing, an author reviewed and revised the content as needed. See [Our principles for using AI-generated content in Microsoft Learn](#).

How order of deletion is determined

When you delete a resource group, Resource Manager determines the order to delete resources. It uses the following order:

1. All the child (nested) resources are deleted.
2. Resources that manage other resources are deleted next. A resource can have the `managedBy` property set to indicate that a different resource manages it. When this property is set, the resource that manages the other resource is deleted before the other resources.
3. The remaining resources are deleted after the previous two categories.

After the order is determined, Resource Manager issues a DELETE operation for each resource. It waits for any dependencies to finish before proceeding.

For synchronous operations, the expected successful response codes are:

- 200
- 204
- 404

For asynchronous operations, the expected successful response is 202. Resource Manager tracks the location header or the azure-async operation header to determine the status of the asynchronous delete operation.

Deletion errors

When a delete operation returns an error, Resource Manager retries the DELETE call. Retries happen for the 5xx, 429 and 408 status codes. By default, the time period for retry is 15 minutes.

After deletion

Resource Manager issues a GET call on each resource that it tried to delete. The response of this GET call is expected to be 404. When Resource Manager gets a 404, it considers the deletion to have completed successfully. Resource Manager removes the resource from its cache.

However, if the GET call on the resource returns a 200 or 201, Resource Manager recreates the resource.

If the GET operation returns an error, Resource Manager retries the GET for the following error code:

- Less than 100
- 408
- 429
- Greater than 500

For other error codes, Resource Manager fails the deletion of the resource.

Important

Resource Group deletion is irreversible.

Delete resource group

Use one of the following methods to delete the resource group.

- [PowerShell](#)
- [Azure CLI](#)
- [Portal](#)
- [Python](#)

Azure PowerShellCopy

Open Cloudshell

```
Remove-AzResourceGroup -Name ExampleResourceGroup
```

Delete resource

Use one of the following methods to delete a resource.

- [PowerShell](#)
- [Azure CLI](#)
- [Portal](#)
- [Python](#)

Azure PowerShellCopy

Open Cloudshell

```
Remove-AzResource
  -ResourceGroupName ExampleResourceGroup
  -ResourceName ExampleVM
  -ResourceType Microsoft.Compute/virtualMachines
```

Required access and deletion failures

To delete a resource group, you need access to the delete action for the **Microsoft.Resources/subscriptions/resourceGroups** resource.

Important

The only permission required to delete a resource group is permission to the delete action for deleting resource groups. You do **not** need permission to delete individual resources within that resource group. Additionally, delete actions that are specified in **notActions** for a roleAssignment are superseded by the resource group delete action. This is consistent with the scope hierarchy in the Azure role-based access control model.

For a list of operations, see [Azure resource provider operations](#). For a list of built-in roles, see [Azure built-in roles](#).

If you have the required access, but the delete request fails, it may be because there's a [lock on the resources or resource group](#). Even if you didn't manually lock a resource group, it may have been [automatically locked by a related service](#). Or, the deletion can fail if the resources are connected to resources in other resource groups that aren't being deleted. For example, you can't delete a virtual network with subnets that are still in use by a virtual machine.

Accidental deletion

If you accidentally delete a resource group or resource, in some situations it might be possible to recover it.

Some resource types support *soft delete*. You might have to configure soft delete before you can use it. For more information about enabling soft delete, see the documentation for [Azure Key Vault](#), [Azure Backup](#), and [Azure Storage](#).

You can also [open an Azure support case](#). Provide as much detail as you can about the deleted resources, including their resource IDs, types, and resource names, and request that the support engineer check if the resources can be restored.

Note

Recovery of deleted resources is not possible under all circumstances. A support engineer will investigate your scenario and advise you whether it's possible.

Lock your resources to protect your infrastructure

- Article
- 04/07/2023
- 23 contributors

Feedback

In this article

1. [Lock inheritance](#)
2. [Understand scope of locks](#)
3. [Considerations before applying your locks](#)
4. [Who can create or delete locks](#)

Show 3 more

As an administrator, you can lock an Azure subscription, resource group, or resource to protect them from accidental user deletions and modifications. The lock overrides any user permissions.

Note

This article was partially created with the help of artificial intelligence. Before publishing, an author reviewed and revised the content as needed. See [Our principles for using AI-generated content in Microsoft Learn](#).

You can set locks that prevent either deletions or modifications. In the portal, these locks are called **Delete** and **Read-only**. In the command line, these locks are called **CanNotDelete** and **ReadOnly**.

- **CanNotDelete** means authorized users can read and modify a resource, but they can't delete it.
- **ReadOnly** means authorized users can read a resource, but they can't delete or update it. Applying this lock is similar to restricting all authorized users to the permissions that the **Reader** role provides.

Unlike role-based access control (RBAC), you use management locks to apply a restriction across all users and roles. To learn about setting permissions for users and roles, see [Azure RBAC](#).

Lock inheritance

When you apply a lock at a parent scope, all resources within that scope inherit the same lock. Even resources you add later inherit the same parent lock. The most restrictive lock in the inheritance takes precedence.

[Extension resources](#) inherit locks from the resource they're applied to. For example, Microsoft.Insights/diagnosticSettings is an extension resource type. If you apply a diagnostic setting to a storage blob, and lock the storage account, you're unable to delete the diagnostic setting. This inheritance makes sense because the full resource ID of the diagnostic setting is:

JSONCopy

```
/subscriptions/{sub-id}/resourceGroups/{rg-name}/providers/Microsoft.Storage/storageAccounts/{storage-name}/blobServices/default/providers/microsoft.insights/diagnosticSettings/{setting-name}"
```

Which matches the scope of the resource ID of the resource that is locked:

JSONCopy

```
/subscriptions/{sub-id}/resourceGroups/{rg-name}/providers/Microsoft.Storage/storageAccounts/{storage-name}
```

If you have a **Delete** lock on a resource and attempt to delete its resource group, the feature blocks the whole delete operation. Even if the resource group or other resources in the resource group are unlocked, the deletion doesn't happen. You never have a partial deletion.

When you [cancel an Azure subscription](#):

- A resource lock doesn't block the subscription cancellation.
- Azure preserves your resources by deactivating them instead of immediately deleting them.
- Azure only deletes your resources permanently after a waiting period.

Understand scope of locks

Note

Locks only apply to control plane Azure operations and not to data plane operations.

Azure control plane operations go to <https://management.azure.com>. Azure data plane operations go to your service instance, such as <https://myaccount.blob.core.windows.net/>. See [Azure control plane and data plane](#). To discover which operations use the control plane URL, see the [Azure REST API](#).

The distinction means locks protect a resource from changes, but they don't restrict how a resource performs its functions. A `ReadOnly` lock, for example, on an SQL Database logical server, protects it from deletions or modifications. It allows you to create, update, or delete data in the server database. Data plane operations allow data transactions. These requests don't go to <https://management.azure.com>.

Considerations before applying your locks

Applying locks can lead to unexpected results. Some operations, which don't seem to modify a resource, require blocked actions. Locks prevent the POST method from sending data to the Azure Resource Manager (ARM) API. Some common examples of blocked operations are:

- A read-only lock on a **storage account** prevents users from listing the account keys. A POST request handles the Azure Storage [List Keys](#) operation to protect access to the account keys. The account keys provide complete access to data in the storage account. When a read-only lock is configured for a storage account, users who don't have the account keys need to use Azure AD credentials to access blob or queue data. A read-only lock also prevents the assignment of Azure RBAC roles that are scoped to the storage account or to a data container (blob container or queue).
- A read-only lock on a **storage account** protects RBAC assignments scoped for a storage account or a data container (blob container or queue).
- A read-only lock on a **storage account** prevents the creation of a blob container.

- A read-only lock or cannot-delete lock on a **storage account** doesn't prevent its data from deletion or modification. It also doesn't protect the data in a blob, queue, table, or file.
- The Storage Account API exposes [data plane](#) and [control plane](#) operations. If a request uses **data plane** operations, the lock on the storage account doesn't protect blob, queue, table, or file data within that storage account. If the request uses **control plane** operations, however, the lock protects those resources.

For example, if a request uses [File Shares - Delete](#), which is a control plane operation, the deletion fails. If the request uses [Delete Share](#), which is a data plane operation, the deletion succeeds. We recommend that you use a control plane operation.

- A read-only lock or cannot-delete lock on a **network security group (NSG)** prevents the creation of a traffic flow log for the NSG.
- A read-only lock on an **App Service** resource prevents Visual Studio Server Explorer from displaying files for the resource because that interaction requires write access.
- A read-only lock on a **resource group** that contains an [App Service plan](#) prevents you from [scaling up or out of the plan](#).
- A read-only lock on a **resource group** that contains a **virtual machine** prevents all users from starting or restarting a virtual machine. These operations require a POST method request.
- A read-only lock on a **resource group** that contains an **automation account** prevents all runbooks from starting. These operations require a POST method request.
- A cannot-delete lock on a **resource** or **resource group** prevents the deletion of Azure RBAC assignments.
- A cannot-delete lock on a **resource group** prevents Azure Resource Manager from [automatically deleting deployments](#) in the history. If you reach 800 deployments in the history, your deployments fail.
- A cannot-delete lock on the **resource group** created by **Azure Backup Service** causes backups to fail. The service supports a maximum of 18 restore points. When locked, the backup service can't clean up restore points. For more information, see [Frequently asked questions-Back up Azure VMs](#).
- A cannot-delete lock on a **resource group** that contains **Azure Machine Learning** workspaces prevents autoscaling of [Azure Machine Learning compute clusters](#) from working correctly. With the lock, autoscaling can't remove unused nodes. Your solution consumes more resources than are required for the workload.

- A read-only lock on a **Log Analytics workspace** prevents **User and Entity Behavior Analytics (UEBA)** from being enabled.
- A cannot-delete lock on a **Log Analytics workspace** doesn't prevent [data purge operations](#), remove the [data purge](#) role from the user instead.
- A read-only lock on a **subscription** prevents **Azure Advisor** from working correctly. Advisor is unable to store the results of its queries.
- A read-only lock on an **Application Gateway** prevents you from getting the backend health of the application gateway. That [operation uses a POST method](#), which a read-only lock blocks.
- A read-only lock on an Azure Kubernetes Service (AKS) cluster limits how you can access cluster resources through the portal. A read-only lock prevents you from using the AKS cluster's Kubernetes resources section in the Azure portal to choose a cluster resource. These operations require a POST method request for authentication.
- A cannot-delete lock on a **Virtual Machine** that is protected by **Site Recovery** prevents certain resource links related to Site Recovery from being removed properly when you remove the protection or disable replication. If you plan to protect the VM again later, you need to remove the lock prior to disabling protection. If you don't remove the lock, you need to follow certain steps to clean up the stale links before you can protect the VM. For more information, see [Troubleshoot Azure VM replication](#).

Who can create or delete locks

To create or delete management locks, you need access to `Microsoft.Authorization/*` or `Microsoft.Authorization/locks/*` actions. Only the **Owner** and the **User Access Administrator** built-in roles can create and delete management locks. You can create a custom role with the required permissions.

Managed applications and locks

Some Azure services, such as Azure Databricks, use [managed applications](#) to implement the service. In that case, the service creates two resource groups. One is an unlocked resource group that contains a service overview. The other is a locked resource group that contains the service infrastructure.

If you try to delete the infrastructure resource group, you get an error stating that the resource group is locked. If you try to delete the lock for the infrastructure resource group, you get an error stating that the lock can't be deleted because a system application owns it.

Instead, delete the service, which also deletes the infrastructure resource group.

For managed applications, choose the service you deployed.

The screenshot shows the Azure portal's 'Resource groups' blade for the 'example-group' resource group. On the left, there's a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Events, Settings, Quickstart, Resource costs, Deployments, and Policies. The 'Overview' link is selected. On the right, there's a main content area with sections for 'Subscription (change)', 'Tags (change)', and a table listing resources. The table has a single row for 'example-databricks'. A red box highlights this row. At the top right of the main area, there are buttons for 'Add', 'Edit columns', and 'Delete resource group'.

Notice the service includes a link for a **Managed Resource Group**. That resource group holds the infrastructure and is locked. You can only delete it indirectly.

The screenshot shows the Azure portal's 'Azure Databricks Service' blade for the 'example-databricks' service. On the left, there's a sidebar with links like Overview, Activity log, and Access control (IAM). The 'Overview' link is selected. On the right, there's a table with service details. One of the rows, 'Managed Resource Group', contains the value 'databricks-rg-example-databricks-45fkl5zso2zh6'. This value is highlighted with a red box.

To delete everything for the service, including the locked infrastructure resource group, choose **Delete** for the service.

The screenshot shows the Azure portal's 'Azure Databricks Service' blade for the 'example-databricks' service. On the left, there's a sidebar with links like Overview, Activity log, and Access control (IAM). The 'Overview' link is selected. On the right, there's a table with service details. At the bottom right of the table, there's a large 'Delete' button, which is highlighted with a red box.

Configure locks

Portal

In the left navigation panel, the subscription lock feature's name is **Resource locks**, while the resource group lock feature's name is **Locks**.

1. In the Settings blade for the resource, resource group, or subscription that you wish to lock, select **Locks**.

The screenshot shows the Microsoft Azure portal interface. At the top is a blue header bar with the text "Microsoft Azure". Below it is a breadcrumb navigation bar with "Home > ExampleGroup Resource group". A search bar is located below the breadcrumb. On the left, there is a vertical navigation menu with options: "Settings" (selected), "Deployments", "Security", "Policies", "Properties", and "Locks" (highlighted with a red box). The main content area is titled "ExampleGroup" and "Resource group".

2. To add a lock, select **Add**. If you want to create a lock at a parent level, select the parent. The currently selected resource inherits the lock from the parent. For example, you could lock the resource group to apply a lock to all its resources.

The screenshot shows the "Locks" blade for the "ExampleGroup" resource group. At the top, there are four buttons: "+ Add" (highlighted with a red box), "Resource group", "Subscription", and "Refresh". Below the buttons is a table with columns: "Lock name", "Lock type", "Scope", and "Notes". A message "This resource has no locks." is displayed in the table body.

3. Give the lock a name and lock level. Optionally, you can add notes that describe the lock.

Lock name: DatabaseServerLock

Lock type: Delete

Notes: Prevent deleting the database server.

OK **Cancel**

4. To delete the lock, select the **Delete** button.

Lock name	Lock type	Scope	Notes	Actions
DatabaseServerLock	Delete	databaseserverexample0503	Prevent deleting the database server.	Edit Delete

Template

When using an ARM template or Bicep file to deploy a lock, it's good to understand how the deployment scope and the lock scope work together. To apply a lock at the deployment scope, such as locking a resource group or a subscription, leave the scope property unset. When locking a resource, within the deployment scope, set the scope property on the lock.

The following template applies a lock to the resource group it's deployed to. Notice there isn't a scope property on the lock resource because the lock scope matches the deployment scope. Deploy this template at the resource group level.

- [JSON](#)
- [Bicep](#)

JSONCopy

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
  },
  "resources": [
    {
      "type": "Microsoft.Authorization/locks",
      "name": "[concat('lock-', uniqueId())]",
      "apiVersion": "2018-11-01",
      "location": "[resourceGroup().location]",
      "properties": {
        "lockType": "Delete"
      }
    }
  ]
}
```

```

    "type": "Microsoft.Authorization/locks",
    "apiVersion": "2016-09-01",
    "name": "rgLock",
    "properties": {
        "level": "CanNotDelete",
        "notes": "Resource group should not be deleted."
    }
}
]
}

```

To create a resource group and lock it, deploy the following template at the subscription level.

- [JSON](#)
- [Bicep](#)

JSONCopy

```
{
  "$schema": "https://schema.management.azure.com/schemas/2018-05-01/subscriptionDeploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "rgName": {
      "type": "string"
    },
    "rgLocation": {
      "type": "string"
    }
  },
  "variables": {},
  "resources": [
    {
      "type": "Microsoft.Resources/resourceGroups",
      "apiVersion": "2021-04-01",
      "name": "[parameters('rgName')]",
      "location": "[parameters('rgLocation')]",
      "properties": {}
    },
    {
      "type": "Microsoft.Resources/deployments",
      "apiVersion": "2021-04-01",
      "name": "lockDeployment",
      "resourceGroup": "[parameters('rgName')]",
      "dependsOn": [
        "[ resourceId('Microsoft.Resources/resourceGroups/', parameters('rgName'))]"
      ],
      "properties": {
        "mode": "Incremental",
        "template": {
          "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
          "contentVersion": "1.0.0.0",
          "parameters": {},
          "variables": {}
        }
      }
    }
  ]
}
```

```

    "resources": [
        {
            "type": "Microsoft.Authorization/locks",
            "apiVersion": "2016-09-01",
            "name": "rgLock",
            "properties": {
                "level": "CanNotDelete",
                "notes": "Resource group and its resources should not be deleted."
            }
        }
    ],
    "outputs": {}
}
],
"outputs": {}
}

```

When applying a lock to a **resource** within the resource group, add the scope property. Set the scope to the name of the resource to lock.

The following example shows a template that creates an app service plan, a website, and a lock on the website. The lock's scope is set to the website.

- [JSON](#)
- [Bicep](#)

JSONCopy

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "hostingPlanName": {
            "type": "string"
        },
        "location": {
            "type": "string",
            "defaultValue": "[resourceGroup().location]"
        }
    },
    "variables": {
        "siteName": "[concat('ExampleSite', uniqueString(resourceGroup().id))]"
    },
    "resources": [
        {
            "type": "Microsoft.Web/serverfarms",
            "apiVersion": "2020-12-01",
            "name": "[parameters('hostingPlanName')]",
            "location": "[parameters('location')]",
            "sku": {
                "tier": "Free",
                "name": "f1",
                "capacity": 0
            },

```

```

    "properties": {
      "targetWorkerCount": 1
    }
  },
  {
    "type": "Microsoft.Web/sites",
    "apiVersion": "2020-12-01",
    "name": "[variables('siteName')]",
    "location": "[parameters('location')]",
    "dependsOn": [
      "[resourceId('Microsoft.Web/serverfarms', parameters('hostingPlanName'))]"
    ],
    "properties": {
      "serverFarmId": "[parameters('hostingPlanName')]"
    }
  },
  {
    "type": "Microsoft.Authorization/locks",
    "apiVersion": "2016-09-01",
    "name": "siteLock",
    "scope": "[concat('Microsoft.Web/sites/', variables('siteName'))]",
    "dependsOn": [
      "[resourceId('Microsoft.Web/sites', variables('siteName'))]"
    ],
    "properties": {
      "level": "CanNotDelete",
      "notes": "Site should not be deleted."
    }
  }
]
}

```

Azure PowerShell

You lock deployed resources with Azure PowerShell by using the [New-AzResourceLock](#) command.

To lock a resource, provide the name of the resource, its resource type, and its resource group name.

Azure PowerShellCopy

Open Cloudshell

```
New-AzResourceLock -LockLevel CanNotDelete -LockName LockSite -ResourceName examplesite -ResourceType Microsoft.Web/sites -ResourceGroupName exempleresourcegroup
```

To lock a resource group, provide the name of the resource group.

Azure PowerShellCopy

Open Cloudshell

```
New-AzResourceLock -LockName LockGroup -LockLevel CanNotDelete -ResourceGroupName exempleresourcegroup
```

To get information about a lock, use [Get-AzResourceLock](#). To get all the locks in your subscription, use:

Azure PowerShellCopy

Open Cloudshell

```
Get-AzResourceLock
```

To get all locks for a resource, use:

Azure PowerShellCopy

Open Cloudshell

```
Get-AzResourceLock -ResourceName examplesite -ResourceType Microsoft.Web/sites -ResourceGroupName exempleresourcegroup
```

To get all locks for a resource group, use:

Azure PowerShellCopy

Open Cloudshell

```
Get-AzResourceLock -ResourceGroupName exempleresourcegroup
```

To delete a lock for a resource, use:

Azure PowerShellCopy

Open Cloudshell

```
$lockId = (Get-AzResourceLock -ResourceGroupName exempleresourcegroup -ResourceName examplesite -ResourceType Microsoft.Web/sites).LockId  
Remove-AzResourceLock -LockId $lockId
```

To delete a lock for a resource group, use:

Azure PowerShellCopy

Open Cloudshell

```
$lockId = (Get-AzResourceLock -ResourceGroupName exempleresourcegroup).LockId  
Remove-AzResourceLock -LockId $lockId
```

Azure CLI

You lock deployed resources with Azure CLI by using the [az lock create](#) command.

To lock a resource, provide the name of the resource, its resource type, and its resource group name.

Azure CLICopy

```
az lock create --name LockSite --lock-type CanNotDelete --resource-group  
exampleresourcegroup --resource-name examplesite --resource-type  
Microsoft.Web/sites
```

To lock a resource group, provide the name of the resource group.

Azure CLICopy

```
az lock create --name LockGroup --lock-type CanNotDelete --resource-group  
exampleresourcegroup
```

To get information about a lock, use [az lock list](#). To get all the locks in your subscription, use:

Azure CLICopy

```
az lock list
```

To get all locks for a resource, use:

Azure CLICopy

```
az lock list --resource-group exampleresourcegroup --resource-name examplesite --  
namespace Microsoft.Web --resource-type sites --parent ""
```

To get all locks for a resource group, use:

Azure CLICopy

```
az lock list --resource-group exampleresourcegroup
```

To delete a lock for a resource, use:

Azure CLICopy

```
lockid=$(az lock show --name LockSite --resource-group exampleresourcegroup --  
resource-type Microsoft.Web/sites --resource-name examplesite --output tsv --query  
id)  
az lock delete --ids $lockid
```

To delete a lock for a resource group, use:

Azure CLICopy

```
lockid=$(az lock show --name LockGroup --resource-group exampleresourcegroup --  
output tsv --query id)  
az lock delete --ids $lockid
```

Python

You lock deployed resources with Python by using the [ManagementLockClient.management_locks.create or update at resource group level](#) command.

To lock a resource, provide the name of the resource, its resource type, and its resource group name.

PythonCopy

```
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ManagementLockClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

lock_client = ManagementLockClient(credential, subscription_id)

lock_result = lock_client.management_locks.create_or_update_at_resource_level(
    "exampleGroup",
    "Microsoft.Web",
    "",
    "sites",
    "examplesite",
    "lockSite",
    {
        "level": "CanNotDelete"
    }
)
```

To lock a resource group, provide the name of the resource group.

PythonCopy

```
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ManagementLockClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

lock_client = ManagementLockClient(credential, subscription_id)

lock_result =
lock_client.management_locks.create_or_update_at_resource_group_level(
    "exampleGroup",
    "lockGroup",
    {
        "level": "CanNotDelete"
    }
)
```

To get information about all locks in your subscription, use [ManagementLockClient.management_locks.get](#). To get all the locks in your subscription, use:

PythonCopy

```
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ManagementLockClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

lock_client = ManagementLockClient(credential, subscription_id)

lock_result = lock_client.management_locks.list_at_subscription_level()

for lock in lock_result:
    print(f"Lock name: {lock.name}")
    print(f"Lock level: {lock.level}")
    print(f"Lock notes: {lock.notes}")
```

To get a lock for a resource, use:

PythonCopy

```
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ManagementLockClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

lock_client = ManagementLockClient(credential, subscription_id)

lock_result = lock_client.management_locks.get_at_resource_level(
    "exampleGroup",
    "Microsoft.Web",
    "",
    "sites",
    "examplesite",
    "lockSite"
)

print(f"Lock ID: {lock_result.id}")
print(f"Lock Name: {lock_result.name}")
print(f"Lock Level: {lock_result.level}")
```

To get a lock for a resource group, use:

PythonCopy

```
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ManagementLockClient
```

```

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

lock_client = ManagementLockClient(credential, subscription_id)

lock_result = lock_client.management_locks.get_at_resource_group_level(
    "exampleGroup",
    "lockGroup"
)

print(f"Lock ID: {lock_result.id}")
print(f"Lock Level: {lock_result.level}")

```

To delete a lock for a resource, use:

PythonCopy

```

import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ManagementLockClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

lock_client = ManagementLockClient(credential, subscription_id)

lock_client.management_locks.delete_at_resource_level(
    "exampleGroup",
    "Microsoft.Web",
    "",
    "sites",
    "examplesite",
    "lockSite"
)

```

To delete a lock for a resource group, use:

PythonCopy

```

import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ManagementLockClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

lock_client = ManagementLockClient(credential, subscription_id)

lock_client.management_locks.delete_at_resource_group_level("exampleGroup",
    "lockGroup")

```

REST API

You can lock deployed resources with the [REST API for management locks](#). The REST API lets you create and delete locks and retrieve information about existing locks.

To create a lock, run:

HTTPCopy

```
PUT  
https://management.azure.com/{scope}/providers/Microsoft.Authorization/locks/{lock  
-name}?api-version={api-version}
```

The scope could be a subscription, resource group, or resource. The lock name can be whatever you want to call it. For the API version, use **2016-09-01**.

In the request, include a JSON object that specifies the lock properties.

JSONCopy

```
{  
  "properties": {  
    "level": "CanNotDelete",  
    "notes": "Optional text notes."  
  }  
}
```

Authenticate requests across tenants

- Article
- 04/09/2023
- 2 contributors

Feedback

In this article

1. [Header values for authentication](#)
2. [Processing the request](#)
3. [Next steps](#)

When creating a multi-tenant application, you may need to handle authentication requests for resources that are in different tenants. A common scenario is when a virtual machine in one tenant must join a virtual network in another tenant. Azure Resource Manager provides a header value for storing auxiliary tokens to authenticate the requests to different tenants.

Header values for authentication

The request has the following authentication header values:

Header name	Description	Example value
Authorization	Primary token	Bearer <primary-token>
x-ms-authorization-auxiliary	Auxiliary tokens	Bearer <auxiliary-token1>, EncryptedBearer <auxiliary-token2>, Bearer <auxiliary-token3>

The auxiliary header can hold up to three auxiliary tokens.

In the code of your multi-tenant app, get the authentication token for other tenants and store them in the auxiliary headers. The user or application must have been invited as a guest to the other tenants.

Processing the request

When your app sends a request to Resource Manager, the request is run under the identity from the primary token. The primary token must be valid and unexpired. This token must be from a tenant that can manage the subscription.

When the request references a resource from different tenant, Resource Manager checks the auxiliary tokens to determine if the request can be processed. All auxiliary tokens in the header must be valid and unexpired. If any token is expired, Resource Manager returns a 401 response code. The response includes the client ID and tenant ID from the token that isn't valid. If the auxiliary header contains a valid token for the tenant, the cross tenant request is processed.

Throttling Resource Manager requests

- Article
- 03/09/2023
- 5 contributors

Feedback

In this article

1. [Subscription and tenant limits](#)
2. [Resource provider limits](#)
3. [Error code](#)
4. [Remaining requests](#)

Show 2 more

This article describes how Azure Resource Manager throttles requests. It shows you how to track the number of requests that remain before reaching the limit, and how to respond when you've reached the limit.

Throttling happens at two levels. Azure Resource Manager throttles requests for the subscription and tenant. If the request is under the throttling limits for the subscription and tenant, Resource Manager routes the request to the resource provider. The resource provider applies throttling limits that are tailored to its operations.

The following image shows how throttling is applied as a request goes from the user to Azure Resource Manager and the resource provider. The image shows that requests are initially throttled per principal ID and per Azure Resource Manager instance in the region of the user sending the request. The requests are throttled per hour. When the request is forwarded to the resource provider, requests are throttled per region of the resource rather than per Azure Resource Manager instance in region of the user. The resource provider requests are also throttled per principal user ID and per hour.

Subscription and tenant limits

Every subscription-level and tenant-level operation is subject to throttling limits. Subscription requests are ones that involve passing your subscription ID, such as retrieving the resource groups in your subscription. Tenant requests don't include your subscription ID, such as retrieving valid Azure locations.

The default throttling limits per hour are shown in the following table.

Scope	Operations	Limit
Subscription	reads	12000
Subscription	deletes	15000
Subscription	writes	1200
Tenant	reads	12000
Tenant	writes	1200

These limits are scoped to the security principal (user or application) making the requests and the subscription ID or tenant ID. If your requests come from more than

one security principal, your limit across the subscription or tenant is greater than 12,000 and 1,200 per hour.

These limits apply to each Azure Resource Manager instance. There are multiple instances in every Azure region, and Azure Resource Manager is deployed to all Azure regions. So, in practice, the limits are higher than these limits. The requests from a user are usually handled by different instances of Azure Resource Manager.

The remaining requests are returned in the [response header values](#).

Resource provider limits

Resource providers apply their own throttling limits. Within each subscription, the resource provider throttles per region of the resource in the request. Because Resource Manager throttles by instance of Resource Manager, and there are several instances of Resource Manager in each region, the resource provider might receive more requests than the default limits in the previous section.

This section discusses the throttling limits of some widely used resource providers.

Storage throttling

The following limits apply only when you perform management operations by using Azure Resource Manager with Azure Storage. The limits apply per region of the resource in the request.

Resource	Limit
Storage account management operations (read)	800 per 5 minutes
Storage account management operations (write)	10 per second / 1200 per hour
Storage account management operations (list)	100 per 5 minutes

Network throttling

The Microsoft.Network resource provider applies the following throttle limits:

Operation	Limit
write / delete (PUT)	1000 per 5 minutes
read (GET)	10000 per 5 minutes

Note

Azure DNS and **Azure Private DNS** have a throttle limit of 500 read (GET) operations per 5 minutes.

Compute throttling

For information about throttling limits for compute operations, see [Troubleshooting API throttling errors - Compute](#).

For checking virtual machine instances within a Virtual Machine Scale Set, use the [Virtual Machine Scale Sets operations](#). For example, use the [Virtual Machine Scale Set VMs - List](#) with parameters to check the power state of virtual machine instances. This API reduces the number of requests.

Azure Resource Graph throttling

[Azure Resource Graph](#) limits the number of requests to its operations. The steps in this article to determine the remaining requests and how to respond when the limit is reached also apply to Resource Graph. However, Resource Graph sets its own limit and reset rate. For more information, see [Resource Graph throttling headers](#).

Other resource providers

For information about throttling in other resource providers, see:

- [Azure Key Vault throttling guidance](#)
- [AKS troubleshooting](#)
- [Managed identities](#)

Error code

When you reach the limit, you receive the HTTP status code **429 Too many requests**. The response includes a **Retry-After** value, which specifies the number of seconds your application should wait (or sleep) before sending the next request. If you send a request before the retry value has elapsed, your request isn't processed and a new retry value is returned.

After waiting for specified time, you can also close and reopen your connection to Azure. By resetting the connection, you may connect to a different instance of Azure Resource Manager.

If you're using an Azure SDK, the SDK may have an auto retry configuration. For more information, see [Retry guidance for Azure services](#).

Some resource providers return 429 to report a temporary problem. The problem could be an overload condition that isn't directly caused by your request. Or, it could represent a temporary error about the state of the target resource or dependent resource. For example, the network resource provider returns 429 with the **RetryableErrorDueToAnotherOperation** error code when the target resource is locked by another operation. To determine if the error comes from throttling or a temporary condition, view the error details in the response.

Remaining requests

You can determine the number of remaining requests by examining response headers. Read requests return a value in the header for the number of remaining read requests. Write requests include a value for the number of remaining write requests. The following table describes the response headers you can examine for those values:

Response header	Description
x-ms-ratelimit-remaining-subscription-deletes	Subscription scoped deletes remaining. This value is returned on delete operations.
x-ms-ratelimit-remaining-subscription-reads	Subscription scoped reads remaining. This value is returned on read operations.
x-ms-ratelimit-remaining-subscription-writes	Subscription scoped writes remaining. This value is returned on write operations.
x-ms-ratelimit-remaining-tenant-reads	Tenant scoped reads remaining
x-ms-ratelimit-remaining-tenant-writes	Tenant scoped writes remaining
x-ms-ratelimit-remaining-subscription-resource-requests	Subscription scoped resource type requests remaining. This header value is only returned if a service has overridden the default limit. Resource Manager adds this value instead of the subscription reads or writes.
x-ms-ratelimit-remaining-subscription-resource-entities-read	Subscription scoped resource type collection requests remaining. This header value is only returned if a service has overridden the default limit. This value provides the number of remaining collection requests (list resources).
x-ms-ratelimit-remaining-tenant-resource-requests	Tenant scoped resource type requests remaining. This header is only added for requests at tenant level, and only if a

Response header	Description
	service has overridden the default limit. Resource Manager adds this value instead of the tenant reads or writes.
x-ms-ratelimit-remaining-tenant-resource-entities-read	Tenant scoped resource type collection requests remaining.
	This header is only added for requests at tenant level, and only if a service has overridden the default limit.

The resource provider can also return response headers with information about remaining requests. For information about response headers returned by the Compute resource provider, see [Call rate informational response headers](#).

Retrieving the header values

Retrieving these header values in your code or script is no different than retrieving any header value.

For example, in **C#**, you retrieve the header value from an **HttpWebResponse** object named **response** with the following code:

C#Copy

```
response.Headers.GetValues("x-ms-ratelimit-remaining-subscription-
reads").GetValue(0)
```

In **PowerShell**, you retrieve the header value from an Invoke-WebRequest operation.

PowerShellCopy

```
$r = Invoke-WebRequest -Uri
https://management.azure.com/subscriptions/{guid}/resourcegroups?api-version=2016-
09-01 -Method GET -Headers $authHeaders
$r.Headers["x-ms-ratelimit-remaining-subscription-reads"]
```

For a complete PowerShell example, see [Check Resource Manager Limits for a Subscription](#).

If you want to see the remaining requests for debugging, you can provide the **-Debug** parameter on your **PowerShell** cmdlet.

PowerShellCopy

```
Get-AzResourceGroup -Debug
```

Which returns many values, including the following response value:

OutputCopy

```
DEBUG: ===== HTTP RESPONSE =====
```

Status Code:

OK

Headers:

```
Pragma : no-cache
x-ms-ratelimit-remaining-subscription-reads: 11999
```

To get write limits, use a write operation:

PowerShellCopy

```
New-AzResourceGroup -Name myresourcegroup -Location westus -Debug
```

Which returns many values, including the following values:

OutputCopy

```
DEBUG: ===== HTTP RESPONSE =====
```

Status Code:

Created

Headers:

```
Pragma : no-cache
x-ms-ratelimit-remaining-subscription-writes: 1199
```

In **Azure CLI**, you retrieve the header value by using the more verbose option.

Azure CLICopy

```
az group list --verbose --debug
```

Which returns many values, including the following values:

OutputCopy

```
msrest.http_logger : Response status: 200
msrest.http_logger : Response headers:
msrest.http_logger :     'Cache-Control': 'no-cache'
msrest.http_logger :     'Pragma': 'no-cache'
msrest.http_logger :     'Content-Type': 'application/json; charset=utf-8'
msrest.http_logger :     'Content-Encoding': 'gzip'
msrest.http_logger :     'Expires': '-1'
msrest.http_logger :     'Vary': 'Accept-Encoding'
msrest.http_logger :     'x-ms-ratelimit-remaining-subscription-reads': '11998'
```

To get write limits, use a write operation:

Azure CLICopy

```
az group create -n myresourcegroup --location westus --verbose --debug
```

Which returns many values, including the following values:

OutputCopy

```
msrest.http_logger : Response status: 201
msrest.http_logger : Response headers:
msrest.http_logger :   'Cache-Control': 'no-cache'
msrest.http_logger :   'Pragma': 'no-cache'
msrest.http_logger :   'Content-Length': '163'
msrest.http_logger :   'Content-Type': 'application/json; charset=utf-8'
msrest.http_logger :   'Expires': '-1'
msrest.http_logger :   'x-ms-ratelimit-remaining-subscription-writes': '1199'
```

Track asynchronous Azure operations

- Article
- 02/01/2022
- 1 contributor

Feedback

In this article

1. [Status codes for asynchronous operations](#)
2. [URL to monitor status](#)
3. [Azure-AsyncOperation request and response](#)
4. [provisioningState values](#)

Show 2 more

Some Azure REST operations run asynchronously because the operation can't be completed quickly. This article describes how to track the status of asynchronous operations through values returned in the response.

Status codes for asynchronous operations

An asynchronous operation initially returns an HTTP status code of either:

- 201 (Created)
- 202 (Accepted)

However, that status code doesn't necessarily mean the operation is asynchronous. An asynchronous operation also returns a value for provisioningState that indicates the operation hasn't finished. The value can vary by operation but won't include **Succeeded**, **Failed**, or **Canceled**. Those three values indicate the operation has finished. If no value is returned for provisioningState, the operation has finished and succeeded.

When the operation successfully completes, it returns either:

- 200 (OK)
- 204 (No Content)

Refer to the [REST API documentation](#) to see the responses for the operation you're executing.

After getting the 201 or 202 response code, you're ready to monitor the status of the operation.

URL to monitor status

There are two different ways to monitor the status the asynchronous operation. You determine the correct approach by examining the header values that are returned from your original request. First, look for:

- Azure-AsyncOperation - URL for checking the ongoing status of the operation. If your operation returns this value, use it to track the status of the operation.
- Retry-After - The number of seconds to wait before checking the status of the asynchronous operation.

If Azure-AsyncOperation isn't one of the header values, then look for:

- Location - URL for determining when an operation has completed. Only use this value only when Azure-AsyncOperation isn't returned.
- Retry-After - The number of seconds to wait before checking the status of the asynchronous operation.

Azure-AsyncOperation request and response

If you have a URL from the Azure-AsyncOperation header value, send a GET request to that URL. Use the value from Retry-After to schedule how often to check the status. You'll get a response object that indicates the status of the operation. A different response is returned when checking the status of the operation with the Location URL. For more information about the response from a location URL, see [Create storage account \(202 with Location and Retry-After\)](#).

The response properties can vary but always include the status of the asynchronous operation.

JSONCopy

```
{  
  "status": "{status-value}"
```

```
}
```

The following example shows other values that might be returned from the operation:

JSONCopy

```
{  
    "id": "{resource path from GET operation}",  
    "name": "{operation-id}",  
    "status" : "Succeeded | Failed | Canceled | {resource provider values}",  
    "startTime": "2017-01-06T20:56:36.002812+00:00",  
    "endTime": "2017-01-06T20:56:56.002812+00:00",  
    "percentComplete": {double between 0 and 100 },  
    "properties": {  
        /* Specific resource provider values for successful operations */  
    },  
    "error" : {  
        "code": "{error code}",  
        "message": "{error description}"  
    }  
}
```

The error object is returned when the status is Failed or Canceled. All other values are optional. The response you receive may look different than the example.

provisioningState values

Operations that create, update, or delete (PUT, PATCH, DELETE) a resource typically return a provisioningState value. When an operation has completed, one of following three values is returned:

- Succeeded
- Failed
- Canceled

All other values indicate the operation is still running. The resource provider can return a customized value that indicates its state. For example, you may receive **Accepted** when the request is received and running.

Example requests and responses

Start virtual machine (202 with Azure-AsyncOperation)

This example shows how to determine the status of [start operation for virtual machines](#). The initial request is in the following format:

HTTPCopy

```
POST  
https://management.azure.com/subscriptions/{subscription-  
id}/resourceGroups/{resource-  
group}/providers/Microsoft.Compute/virtualMachines/{vm-name}/start?api-  
version=2019-12-01
```

It returns status code 202. Among the header values, you see:

HTTPCopy

```
Azure-AsyncOperation : https://management.azure.com/subscriptions/{subscription-  
id}/providers/Microsoft.Compute/locations/{region}/operations/{operation-id}?api-  
version=2019-12-01
```

To check the status of the asynchronous operation, sending another request to that URL.

HTTPCopy

```
GET  
https://management.azure.com/subscriptions/{subscription-  
id}/providers/Microsoft.Compute/locations/{region}/operations/{operation-id}?api-  
version=2019-12-01
```

The response body contains the status of the operation:

JSONCopy

```
{  
  "startTime": "2017-01-06T18:58:24.7596323+00:00",  
  "status": "InProgress",  
  "name": "9a062a88-e463-4697-bef2-fe039df73a02"  
}
```

Deploy resources (201 with Azure-AsyncOperation)

This example shows how to determine the status of [deployments operation for deploying resources](#) to Azure. The initial request is in the following format:

HTTPCopy

```
PUT  
https://management.azure.com/subscriptions/{subscription-  
id}/resourcegroups/{resource-  
group}/providers/microsoft.resources/deployments/{deployment-name}?api-  
version=2020-06-01
```

It returns status code 201. The body of the response includes:

JSONCopy

```
"provisioningState": "Accepted",
```

Among the header values, you see:

HTTPCopy

```
Azure-AsyncOperation: https://management.azure.com/subscriptions/{subscription-id}/resourcegroups/{resource-group}/providers/Microsoft.Resources/deployments/{deployment-name}/operationStatuses/{operation-id}?api-version=2020-06-01
```

To check the status of the asynchronous operation, sending another request to that URL.

HTTPCopy

```
GET  
https://management.azure.com/subscriptions/{subscription-id}/resourcegroups/{resource-group}/providers/Microsoft.Resources/deployments/{deployment-name}/operationStatuses/{operation-id}?api-version=2020-06-01
```

The response body contains the status of the operation:

JSONCopy

```
{  
    "status": "Running"  
}
```

When the deployment is finished, the response contains:

JSONCopy

```
{  
    "status": "Succeeded"  
}
```

Create storage account (202 with Location and Retry-After)

This example shows how to determine the status of the [create operation for storage accounts](#). The initial request is in the following format:

HTTPCopy

```
PUT  
https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/{resource-group}/providers/Microsoft.Storage/storageAccounts/{storage-name}?api-version=2019-06-01
```

And the request body contains properties for the storage account:

JSONCopy

```
{  
    "location": "South Central US",  
    "properties": {},  
    "sku": {  
        "name": "Standard_LRS"  
    },  
    "kind": "Storage"  
}
```

It returns status code 202. Among the header values, you see the following two values:

HTTPCopy

```
Location: https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Storage/operations/{operation-id}?monitor=true&api-version=2019-06-01  
Retry-After: 17
```

After waiting for number of seconds specified in Retry-After, check the status of the asynchronous operation by sending another request to that URL.

HTTPCopy

```
GET  
https://management.azure.com/subscriptions/{subscription-id}/providers/Microsoft.Storage/operations/{operation-id}?monitor=true&api-version=2019-06-01
```

If the request is still running, you receive a status code 202. If the request has completed, you receive a status code 200. The body of the response contains the properties of the storage account that was created.

Azure Resource Manager metrics in Azure Monitor

- Article
- 04/26/2023
- 3 contributors

Feedback

In this article

1. [Accessing Azure Resource Manager metrics](#)
2. [Metric definition](#)
3. [Examples](#)

4. [Next steps](#)

When you create and manage resources in Azure, your requests are orchestrated through Azure's [control plane](#), Azure Resource Manager. This article describes how to monitor the volume and latency of control plane requests made to Azure.

With these metrics, you can observe traffic and latency for control plane requests throughout your subscriptions. For example, you can now figure out when your requests have been throttled or failed by filtering for specific status codes by [examining throttled requests](#) and [examining server errors](#).

The metrics are available for up to three months (93 days) and only track synchronous requests. For a scenario like a VM creation, the metrics do not represent the performance or reliability of the long running asynchronous operation.

Accessing Azure Resource Manager metrics

You can access control plane metrics via the Azure Monitor REST APIs, SDKs, and the Azure portal (by selecting the "Azure Resource Manager" metric). For an overview on Azure Monitor, see [Azure Monitor Metrics](#).

There is no opt-in or sign-up process to access control plane metrics.

For guidance on how to retrieve a bearer token and make requests to Azure, see [Azure REST API reference](#).

Metric definition

The definition for Azure Resource Manager metrics in Azure Monitor is only accessible through the 2017-12-01-preview API version. To retrieve the definition, you can run the following snippet, with your subscription ID replacing "00000000-0000-0000-0000-000000000000":

BashCopy

```
curl --location --request GET  
'https://management.azure.com/subscriptions/00000000-0000-0000-0000-  
000000000000/providers/microsoft.insights/metricDefinitions?api-version=2017-12-  
01-preview&metricnamespace=microsoft.resources/subscriptions' \  
--header 'Authorization: bearer {{bearerToken}}'
```

This snippet returns the definition for the metrics schema. Notably, this schema includes the dimensions you can filter on with the Monitor API:

Dimension	Description
Name	
ResourceUri	The full Resource ID for a particular resource.
RequestRegion	The Azure Resource Manager region where your control plane requests land, like "EastUS2". This region is not the resource's location.
StatusCode	Response type from Azure Resource Manager for your control plane request. Possible values are (but not limited to): - 0 - 200 - 201 - 400 - 404 - 429 - 500 - 502
StatusCodeGen	The class for the status code returned from Azure Resource Manager. Possible values are: - 2xx - 4xx - 5xx
Namespace	The namespace for the Resource Provider, in all caps, like "MICROSOFT.COMPUTE"
ResourceType	Any resource type in Azure that you have created or sent a request to, in all caps, like "VIRTUALMACHINES"
Method	The HTTP method used in the request made to Azure Resource Manager. Possible values are: - GET - HEAD - PUT - POST - PATCH - DELETE

Examples

Now, let's look at some scenarios that can help you exploring Azure Resource Manager metrics.

Query traffic and latency control plane metrics via Azure portal

First, navigate to the Azure Monitor blade within the [portal](#):

After selecting **Explore Metrics**, select a single subscription and then select the **Azure Resource Manager** metric:

Then, after selecting **Apply**, you can visualize your Traffic or Latency control plane metrics with custom filtering and splitting:

Query traffic and latency control plane metrics via REST API

After you are authenticated with Azure, you can make a request to retrieve control plane metrics for your subscription. In the script, replace "00000000-0000-0000-0000-000000000000" with your subscription ID. The script will retrieve the average request latency (in seconds) and the total request count for the two day timespan, broken down by one day intervals:

BashCopy

```
curl --location --request GET  
"https://management.azure.com/subscriptions/00000000-0000-0000-0000-  
000000000000/providers/microsoft.insights/metrics?api-version=2021-05-  
01&interval=P1D&metricnames=Latency&metricnamespace=microsoft.resources/subscripti  
ons&region=global&aggregation=average,count&timespan=2021-11-01T00:00:00Z/2021-11-  
03T00:00:00Z" \  
--header "Authorization: bearer {{bearerToken}}"
```

In the case of Azure Resource Manager metrics, you can retrieve the traffic count by using the Latency metric and including the 'count' aggregation. You'll see a JSON response for the request:

JsonCopy

```
{  
    "cost": 5758,  
    "timespan": "2021-11-01T00:00:00Z/2021-11-03T00:00:00Z",  
    "interval": "P1D",  
    "value": [  
        {  
            "id": "subscriptions/00000000-0000-0000-0000-  
000000000000/providers/Microsoft.Insights/metrics/Latency",  
            "type": "Microsoft.Insights/metrics",  
            "name": {
```

```

        "value": "Latency",
        "localizedValue": "Latency"
    },
    "displayDescription": "Latency data for all requests to Azure Resource
Manager",
    "unit": "Seconds",
    "timeseries": [
        {
            "metadatavalues": [],
            "data": [
                {
                    "timeStamp": "2021-11-01T00:00:00Z",
                    "count": 1406.0,
                    "average": 0.19345163584637273
                },
                {
                    "timeStamp": "2021-11-02T00:00:00Z",
                    "count": 1517.0,
                    "average": 0.28294792353328935
                }
            ]
        },
        {
            "errorCode": "Success"
        }
    ],
    "namespace": "microsoft.resources/subscriptions",
    "resourcerregion": "global"
}

```

If you want to retrieve only the traffic count, then you can utilize the Traffic metric with the 'count' aggregation:

BashCopy

```

curl --location --request GET
'https://management.azure.com/subscriptions/00000000-0000-0000-0000-
000000000000/providers/microsoft.insights/metrics?api-version=2021-05-
01&interval=P1D&metricnames=Traffic&metricnamespace=microsoft.resources/subscripti
ons&region=global&aggregation=count&timespan=2021-11-01T00:00:00Z/2021-11-
03T00:00:00Z' \
--header 'Authorization: bearer {{bearerToken}}'

```

The response for the request is:

JsonCopy

```

{
    "cost": 2879,
    "timespan": "2021-11-01T00:00:00Z/2021-11-03T00:00:00Z",
    "interval": "P1D",
    "value": [
        {
            "id": "subscriptions/00000000-0000-0000-0000-
000000000000/providers/Microsoft.Insights/metrics/Traffic",
            "type": "Microsoft.Insights/metrics",
            "name": {

```

```

        "value": "Traffic",
        "localizedValue": "Traffic"
    },
    "displayDescription": "Traffic data for all requests to Azure Resource
Manager",
    "unit": "Count",
    "timeseries": [
        {
            "metadatavalues": [],
            "data": [
                {
                    "timeStamp": "2021-11-01T00:00:00Z",
                    "count": 1406.0
                },
                {
                    "timeStamp": "2021-11-02T00:00:00Z",
                    "count": 1517.0
                }
            ]
        }
    ],
    "errorCode": "Success"
},
],
"namespace": "microsoft.resources/subscriptions",
"resourcerregion": "global"
}

```

For the metrics supporting dimensions, you need to specify the dimension value to see the corresponding metrics values. For example, if you want to focus on the **Latency** for successful requests to ARM, you need to filter the **StatusCodesClass** dimension with **2XX**.

If you want to look at the number of requests made in your subscription for Networking resources, like Virtual Networks and Load Balancers, you would need to filter the **Namespace** dimension for **MICROSOFT.NETWORK**.

Examining Throttled Requests

To view only your throttled requests, you need to filter for 429 status code responses only. For REST API calls, filtering is accomplished via the [\\$filter property](#) and the StatusCode dimension by appending: \$filter=StatusCode eq '429' as seen at the end of the request in the following snippet:

BashCopy

```

curl --location --request GET
'https://management.azure.com/subscriptions/00000000-0000-0000-0000-
000000000000/providers/microsoft.insights/metrics?api-version=2021-05-
01&interval=P1D&metricnames=Latency&metricnamespace=microsoft.resources/subscripti
ons&region=global&aggregation=count,average&timespan=2021-11-01T00:00:00Z/2021-11-
03T00:00:00Z&$filter=StatusCode%20eq%20%27429%27' \
--header 'Authorization: bearer {{bearerToken}}'

```

You can also filter directly in portal:

Examining Server Errors

Similar to looking at throttled requests, you view *all* requests that returned a server error response code by filtering 5xx responses only. For REST API calls, filtering is accomplished via the [\\$filter property](#) and the StatusCodeClass dimension by appending: \$filter=StatusCodeClass eq '5xx' as seen at the end of the request in the following snippet:

BashCopy

```
curl --location --request GET  
'https://management.azure.com/subscriptions/00000000-0000-0000-0000-  
000000000000/providers/microsoft.insights/metrics?api-version=2021-05-  
01&interval=P1D&metricnames=Latency&metricnamespace=microsoft.resources/subscription  
s&region=global&aggregation=count,average&timespan=2021-11-01T00:00:00Z/2021-11-  
03T00:00:00Z&$filter=StatusCodeClass%20eq%20%275xx%27' \  
--header 'Authorization: bearer {{bearerToken}}'
```

You can also accomplish generic server errors filtering within portal by setting the filter property to 'StatusCodeClass' and the value to '5xx', similar to what was done in the throttling example.

App Service overview

- Article
- 06/15/2023
- 26 contributors

Feedback

In this article

1. [Why use App Service?](#)
2. [App Service on Linux](#)
3. [Next steps](#)

Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, be it .NET, .NET Core, Java, Node.js, PHP, and Python. Applications run and scale with ease on both Windows and [Linux](#)-based environments.

App Service adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. Additionally, you can take advantage of its DevOps capabilities, such as continuous deployment from Azure

DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and TLS/SSL certificates.

With App Service, you pay for the Azure compute resources you use. The compute resources you use are determined by the *App Service plan* that you run your apps on. For more information, see [Azure App Service plans overview](#).

Why use App Service?

Azure App Service is a fully managed platform as a service (PaaS) offering for developers. Here are some key features of App Service:

- **Multiple languages and frameworks** - App Service has first-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can also run [PowerShell and other scripts or executables](#) as background services.
- **Managed production environment** - App Service automatically [patches and maintains the OS and language frameworks](#) for you. Spend time writing great apps and let Azure worry about the platform.
- **Containerization and Docker** - Dockerize your app and host a custom Windows or Linux container in App Service. Run multi-container apps with Docker Compose. Migrate your Docker skills directly to App Service.
- **DevOps optimization** - Set up [continuous integration and deployment](#) with Azure DevOps, GitHub, BitBucket, Docker Hub, or Azure Container Registry. Promote updates through [test and staging environments](#). Manage your apps in App Service by using [Azure PowerShell](#) or the [cross-platform command-line interface \(CLI\)](#).
- **Global scale with high availability** - Scale [up](#) or [out](#) manually or automatically. Host your apps anywhere in Microsoft's global datacenter infrastructure, and the App Service [SLA](#) promises high availability.
- **Connections to SaaS platforms and on-premises data** - Choose from [many hundreds of connectors](#) for enterprise systems (such as SAP), SaaS services (such as Salesforce), and internet services (such as Facebook). Access on-premises data using [Hybrid Connections](#) and [Azure Virtual Networks](#).
- **Security and compliance** - App Service is [ISO, SOC, and PCI compliant](#). Create [IP address restrictions](#) and [managed service identities](#). [Prevent subdomain takeovers](#).
- **Authentication** - [Authenticate users](#) using the built-in authentication component. Authenticate users with [Azure Active Directory](#), [Google](#), [Facebook](#), [Twitter](#), or [Microsoft account](#).
- **Application templates** - Choose from an extensive list of application templates in the [Azure Marketplace](#), such as WordPress, Joomla, and Drupal.
- **Visual Studio and Visual Studio Code integration** - Dedicated tools in Visual Studio and Visual Studio Code streamline the work of creating, deploying, and debugging.

- **API and mobile features** - App Service provides turn-key CORS support for RESTful API scenarios, and simplifies mobile app scenarios by enabling authentication, offline data sync, push notifications, and more.
- **Serverless code** - Run a code snippet or script on-demand without having to explicitly provision or manage infrastructure, and pay only for the compute time your code actually uses (see [Azure Functions](#)).

Besides App Service, Azure offers other services that can be used for hosting websites and web applications. For most scenarios, App Service is the best choice. For microservice architecture, consider [Azure Spring Apps](#) or [Service Fabric](#). If you need more control over the VMs on which your code runs, consider [Azure Virtual Machines](#). For more information about how to choose between these Azure services, see [Azure App Service, Virtual Machines, Service Fabric, and Cloud Services comparison](#).

App Service on Linux

App Service can also host web apps natively on Linux for supported application stacks. It can also run custom Linux containers (also known as Web App for Containers).

Built-in languages and frameworks

App Service on Linux supports a number of language specific built-in images. Just deploy your code. Supported languages include: Node.js, Java (8, 11, and 17), Tomcat, PHP, Python, .NET Core, and Ruby. Run `az webapp list-runtimes --os linux` to view the latest languages and supported versions. If the runtime your application requires is not supported in the built-in images, you can deploy it with a custom container.

Outdated runtimes are periodically removed from the Web Apps Create and Configuration blades in the Portal. These runtimes are hidden from the Portal when they are deprecated by the maintaining organization or found to have significant vulnerabilities. These options are hidden to guide customers to the latest runtimes where they will be the most successful.

When an outdated runtime is hidden from the Portal, any of your existing sites using that version will continue to run. If a runtime is fully removed from the App Service platform, your Azure subscription owner(s) will receive an email notice before the removal.

If you need to create another web app with an outdated runtime version that is no longer shown on the Portal see the language configuration guides for instructions on how to get the runtime version of your site. You can use the Azure CLI to create

another site with the same runtime. Alternatively, you can use the **Export Template** button on the web app blade in the Portal to export an ARM template of the site. You can reuse this template to deploy a new site with the same runtime and configuration.

Limitations

- App Service on Linux is not supported on [Shared](#) pricing tier.
- The Azure portal shows only features that currently work for Linux apps. As features are enabled, they're activated on the portal.
- When deployed to built-in images, your code and content are allocated a storage volume for web content, backed by Azure Storage. The disk latency of this volume is higher and more variable than the latency of the container filesystem. Apps that require heavy read-only access to content files may benefit from the custom container option, which places files in the container filesystem instead of on the content volume.

App Service Environment overview

- Article
- 06/27/2023
- 12 contributors

Feedback

In this article

1. [Usage scenarios](#)
2. [Dedicated environment](#)
3. [Virtual network support](#)
4. [Feature differences](#)

Show 4 more

An App Service Environment is an Azure App Service feature that provides a fully isolated and dedicated environment for running App Service apps securely at high scale.

Note

This article covers the features, benefits, and use cases of App Service Environment v3, which is used with App Service Isolated v2 plans.

An App Service Environment can host your:

- Windows web apps

- Linux web apps
- Docker containers (Windows and Linux)
- Functions
- Logic apps (Standard)

App Service Environments are appropriate for application workloads that require:

- High scale.
- Isolation and secure network access.
- High memory utilization.
- High requests per second (RPS). You can create multiple App Service Environments in a single Azure region or across multiple Azure regions. This flexibility makes an App Service Environment ideal for horizontally scaling stateless applications with a high RPS requirement.

An App Service Environment can host applications from only one customer, and they do so on one of their virtual networks. Customers have fine-grained control over inbound and outbound application network traffic. Applications can establish high-speed secure connections over VPNs to on-premises corporate resources.

Usage scenarios

App Service Environments have many use cases, including:

- Internal line-of-business applications.
- Applications that need more than 30 App Service plan instances.
- Single-tenant systems to satisfy internal compliance or security requirements.
- Network-isolated application hosting.
- Multi-tier applications.

There are many networking features that enable apps in a multi-tenant App Service to reach network-isolated resources or become network-isolated themselves. These features are enabled at the application level. With an App Service Environment, no added configuration is required for the apps to be on a virtual network. The apps are deployed into a network-isolated environment that's already on a virtual network. If you really need a complete isolation story, you can also deploy your App Service Environment onto dedicated hardware.

Dedicated environment

An App Service Environment is a single-tenant deployment of Azure App Service that runs on your virtual network.

Applications are hosted in App Service plans, which are created in an App Service Environment. An App Service plan is essentially a provisioning profile for an application host. As you scale out your App Service plan, you create more application hosts with all the apps in that App Service plan on each host. A single App Service Environment v3 can have up to 200 total App Service plan instances across all the App Service plans combined. A single App Service Isolated v2 (Iv2) plan can have up to 100 instances by itself.

When you're deploying onto dedicated hardware (hosts), you're limited in scaling across all App Service plans to the number of cores in this type of environment. An App Service Environment that's deployed on dedicated hosts has 132 vCores available. I1v2 uses two vCores, I2v2 uses four vCores, and I3v2 uses eight vCores per instance.

Virtual network support

The App Service Environment feature is a deployment of Azure App Service into a single subnet on a virtual network. When you deploy an app into an App Service Environment, the app is exposed on the inbound address that's assigned to the App Service Environment. If your App Service Environment is deployed with an internal virtual IP (VIP) address, the inbound address for all the apps will be an address in the App Service Environment subnet. If your App Service Environment is deployed with an external VIP address, the inbound address will be an internet-addressable address, and your apps will be in a public Domain Name System.

The number of addresses that are used by an App Service Environment v3 in its subnet will vary, depending on the number of instances and the amount of traffic. Some infrastructure roles are automatically scaled, depending on the number of App Service plans and the load. The recommended size for your App Service Environment v3 subnet is a /24 Classless Inter-Domain Routing (CIDR) block with 256 addresses in it, because that size can host an App Service Environment v3 that's scaled out to its limit.

The apps in an App Service Environment don't need any features enabled to access resources on the same virtual network that the App Service Environment is in. If the App Service Environment virtual network is connected to another network, the apps in the App Service Environment can access resources in those extended networks. Traffic can be blocked by user configuration on the network.

The multi-tenant version of Azure App Service contains numerous features to enable your apps to connect to your various networks. With those networking features, your apps can act as though they're deployed on a virtual network. The apps in an App

Service Environment v3 don't need any added configuration to be on the virtual network.

A benefit of using an App Service Environment instead of a multi-tenant service is that any network access controls for the App Service Environment-hosted apps are external to the application configuration. With the apps in the multi-tenant service, you must enable the features on an app-by-app basis and use role-based access control or a policy to prevent any configuration changes.

Feature differences

App Service Environment v3 differs from earlier versions in the following ways:

- There are no networking dependencies on the customer's virtual network. You can secure all inbound and outbound traffic and route outbound traffic as you want.
- You can deploy an App Service Environment v3 that's enabled for zone redundancy. You set zone redundancy only during creation and only in regions where all App Service Environment v3 dependencies are zone redundant. In this case, each App Service Plan on the App Service Environment will need to have a minimum of three instances so that they can be spread across zones. For more information, see [Migrate App Service Environment to availability zone support](#).
- You can deploy an App Service Environment v3 on a dedicated host group. Host group deployments aren't zone redundant.
- Scaling is much faster than with an App Service Environment v2. Although scaling still isn't immediate, as in the multi-tenant service, it's a lot faster.
- Front-end scaling adjustments are no longer required. App Service Environment v3 front ends automatically scale to meet your needs and are deployed on better hosts.
- Scaling no longer blocks other scale operations within the App Service Environment v3. Only one scale operation can be in effect for a combination of OS and size. For example, while your Windows small App Service plan is scaling, you could kick off a scale operation to run at the same time on a Windows medium or anything else other than Windows small.
- You can reach apps in an internal-VIP App Service Environment v3 across global peering. Such access wasn't possible in earlier versions.

A few features that were available in earlier versions of App Service Environment aren't available in App Service Environment v3. For example, you can no longer do the following:

- Perform a backup and restore operation on a storage account behind a firewall.
- Access the FTPS endpoint using a custom domain suffix.

App Service Environment networking

- Article
- 03/10/2023
- 7 contributors

Feedback

In this article

1. [Subnet requirements](#)
2. [Addresses](#)
3. [Ports and network restrictions](#)
4. [Network routing](#)

Show 3 more

App Service Environment is a single-tenant deployment of Azure App Service that hosts Windows and Linux containers, web apps, API apps, logic apps, and function apps. When you install an App Service Environment, you pick the Azure virtual network that you want it to be deployed in. All of the inbound and outbound application traffic is inside the virtual network you specify. You deploy into a single subnet in your virtual network, and nothing else can be deployed into that subnet.

Note

This article is about App Service Environment v3, which is used with isolated v2 App Service plans.

Subnet requirements

You must delegate the subnet to `Microsoft.Web/hostingEnvironments`, and the subnet must be empty.

The size of the subnet can affect the scaling limits of the App Service plan instances within the App Service Environment. It's a good idea to use a /24 address space (256 addresses) for your subnet, to ensure enough addresses to support production scale.

Note

Windows Containers uses an additional IP address per app for each App Service plan instance, and you need to size the subnet accordingly. If your App Service Environment has for example 2 Windows Container App Service plans each with 25 instances and each with 5 apps running, you will need 300 IP addresses and additional addresses to support horizontal (up/down) scale.

If you use a smaller subnet, be aware of the following limitations:

- Any particular subnet has five addresses reserved for management purposes. In addition to the management addresses, App Service Environment dynamically scales the supporting infrastructure, and uses between 4 and 27 addresses, depending on the configuration and load. You can use the remaining addresses for instances in the App Service plan. The minimal size of your subnet is a /27 address space (32 addresses).
- If you run out of addresses within your subnet, you can be restricted from scaling out your App Service plans in the App Service Environment. Another possibility is that you can experience increased latency during intensive traffic load, if Microsoft isn't able to scale the supporting infrastructure.

Addresses

App Service Environment has the following network information at creation:

Address type	Description
App Service Environment virtual network	The virtual network deployed into.
App Service Environment subnet	The subnet deployed into.
Domain suffix	The domain suffix that is used by the apps made.
Virtual IP (VIP)	The VIP type used. The two possible values are internal and external.
Inbound address	The inbound address is the address at which your apps are reached. If you have an internal VIP, it's an address in your App Service Environment subnet. If the address is external, it's a public-facing address.
Default outbound addresses	The apps use this address, by default, when making outbound calls to the internet.

You can find details in the **IP Addresses** portion of the portal, as shown in the following screenshot:

As you scale your App Service plans in your App Service Environment, you'll use more addresses out of your subnet. The number of addresses you use varies, based on the number of App Service plan instances you have, and how much traffic there is.

Apps in the App Service Environment don't have dedicated addresses in the subnet. The specific addresses used by an app in the subnet will change over time.

Ports and network restrictions

For your app to receive traffic, ensure that inbound network security group (NSG) rules allow the App Service Environment subnet to receive traffic from the required ports. In addition to any ports you'd like to receive traffic on, you should ensure that Azure Load Balancer is able to connect to the subnet on port 80. This port is used for health checks of the internal virtual machine. You can still control port 80 traffic from the virtual network to your subnet.

It's a good idea to configure the following inbound NSG rule:

Source / Destination Port(s)	Direction	Source	Destination	Purpose
* / 80,443	Inbound	VirtualNetwork	App Service Environment subnet range	Allow app traffic and internal health ping traffic

The minimal requirement for App Service Environment to be operational is:

Source / Destination Port(s)	Direction	Source	Destination	Purpose
* / 80	Inbound	AzureLoadBalancer	App Service Environment subnet range	Allow internal health ping traffic

If you use the minimum required rule, you might need one or more rules for your application traffic. If you're using any of the deployment or debugging options, you must also allow this traffic to the App Service Environment subnet. The source of these rules can be the virtual network, or one or more specific client IPs or IP ranges. The destination is always the App Service Environment subnet range. The internal health ping traffic on port 80 is isolated between the Load balancer and the internal servers. No outside traffic can reach the health ping endpoint.

The normal app access ports inbound are as follows:

Use	Ports
HTTP/HTTPS	80, 443
FTP/FTPS	21, 990, 10001-10020
Visual Studio remote debugging	4022, 4024, 4026

Use	Ports
Web Deploy service	8172

Note

For FTP access, even if you want to disallow standard FTP on port 21, you still need to allow traffic from the LoadBalancer to the App Service Environment subnet range on port 21, as this is used for internal health ping traffic for the ftp service specifically.

Network routing

You can set route tables without restriction. You can tunnel all of the outbound application traffic from your App Service Environment to an egress firewall device, such as Azure Firewall. In this scenario, the only thing you have to worry about is your application dependencies.

Application dependencies include endpoints that your app needs during runtime. Besides APIs and services the app is calling, dependencies could also be derived endpoints like certificate revocation list (CRL) check endpoints and identity/authentication endpoint, for example Azure Active Directory. If you're using [continuous deployment in App Service](#), you might also need to allow endpoints depending on type and language. Specifically for [Linux continuous deployment](#), you'll need to allow oryx-cdn.microsoft.io:443.

You can put your web application firewall devices, such as Azure Application Gateway, in front of inbound traffic. Doing so allows you to expose specific apps on that App Service Environment.

Your application will use one of the default outbound addresses for egress traffic to public endpoints. If you want to customize the outbound address of your applications on an App Service Environment, you can add a NAT gateway to your subnet.

Note

Outbound SMTP connectivity (port 25) is supported for App Service Environment v3. The supportability is determined by a setting on the subscription where the virtual network is deployed. For virtual networks/subnets created before 1. August 2022 you need to initiate a temporary configuration change to the virtual network/subnet for the setting to be synchronized from the subscription. An example could be to add a temporary subnet, associate/dissociate an NSG temporarily or configure a service endpoint temporarily. For more information and troubleshooting, see [Troubleshoot outbound SMTP connectivity problems in Azure](#).

Private endpoint

In order to enable Private Endpoints for apps hosted in your App Service Environment, you must first enable this feature at the App Service Environment level.

You can activate it through the Azure portal. In the App Service Environment configuration pane, turn **on** the setting `Allow new private endpoints`. Alternatively the following CLI can enable it:

Azure CLICopy

Open Cloudshell

```
az appservice ase update --name myasename --allow-new-private-endpoint-connections true
```

For more information about Private Endpoint and Web App, see [Azure Web App Private Endpoint](#)

DNS

The following sections describe the DNS considerations and configuration that apply inbound to and outbound from your App Service Environment. The examples use the domain suffix `appserviceenvironment.net` from Azure Public Cloud. If you're using other clouds like Azure Government, you'll need to use their respective domain suffix.

DNS configuration to your App Service Environment

If your App Service Environment is made with an external VIP, your apps are automatically put into public DNS. If your App Service Environment is made with an internal VIP, you might need to configure DNS for it. When you created your App Service Environment, if you selected having Azure DNS private zones configured automatically, then DNS is configured in your virtual network. If you chose to configure DNS manually, you need to either use your own DNS server or configure Azure DNS private zones. To find the inbound address, go to the App Service Environment portal, and select **IP Addresses**.

If you want to use your own DNS server, add the following records:

1. Create a zone for `<App Service Environment-name>.appserviceenvironment.net`.
2. Create an A record in that zone that points * to the inbound IP address used by your App Service Environment.
3. Create an A record in that zone that points @ to the inbound IP address used by your App Service Environment.

4. Create a zone in <App Service Environment-name>.appserviceenvironment.net named scm.
5. Create an A record in the scm zone that points * to the IP address used by the private endpoint of your App Service Environment.

To configure DNS in Azure DNS private zones:

1. Create an Azure DNS private zone named <App Service Environment-name>.appserviceenvironment.net.
2. Create an A record in that zone that points * to the inbound IP address.
3. Create an A record in that zone that points @ to the inbound IP address.
4. Create an A record in that zone that points *.scm to the inbound IP address.

In addition to the default domain provided when an app is created, you can also add a custom domain to your app. You can set a custom domain name without any validation on your apps. If you're using custom domains, you need to ensure they have DNS records configured. You can follow the preceding guidance to configure DNS zones and records for a custom domain name (replace the default domain name with the custom domain name). The custom domain name works for app requests, but doesn't work for the scm site. The scm site is only available at <appname>.scm.<asename>.appserviceenvironment.net.

DNS configuration for FTP access

For FTP access to Internal Load balancer (ILB) App Service Environment v3 specifically, you need to ensure DNS is configured. Configure an Azure DNS private zone or equivalent custom DNS with the following settings:

1. Create an Azure DNS private zone named ftp.appserviceenvironment.net.
2. Create an A record in that zone that points <App Service Environment-name> to the inbound IP address.

In addition to setting up DNS, you also need to enable it in the [App Service Environment configuration](#) and at the [app level](#).

DNS configuration from your App Service Environment

The apps in your App Service Environment will use the DNS that your virtual network is configured with. If you want some apps to use a different DNS server, you can manually set it on a per app basis, with the app settings WEBSITE_DNS_SERVER and WEBSITE_DNS_ALT_SERVER. WEBSITE_DNS_ALT_SERVER configures the secondary DNS server. The secondary DNS server is only used when there's no response from the primary DNS server.

Azure App Service diagnostics overview

- Article
- 03/04/2023
- 7 contributors

Feedback

In this article

1. [Open App Service diagnostics](#)
2. [Diagnostic Interface](#)
3. [Ask Genie search box](#)
4. [Risk Alerts](#)

Show 6 more

When you're running a web application, you want to be prepared for any issues that may arise, from 500 errors to your users telling you that your site is down. App Service diagnostics is an intelligent and interactive experience to help you troubleshoot your app with no configuration required. If you do run into issues with your app, App Service diagnostics points out what's wrong to guide you to the right information to more easily and quickly troubleshoot and resolve the issue.

Although this experience is most helpful when you're having issues with your app within the last 24 hours, all the diagnostic graphs are always available for you to analyze.

App Service diagnostics works for not only your app on Windows, but also apps on [Linux/containers](#), [App Service Environment](#), and [Azure Functions](#).

Open App Service diagnostics

To access App Service diagnostics, navigate to your App Service web app or App Service Environment in the [Azure portal](#). In the left navigation, click on **Diagnose and solve problems**.

For Azure Functions, navigate to your function app, and in the top navigation, click on **Platform features**, and select **Diagnose and solve problems** from the **Resource management** section.

In the App Service diagnostics homepage, you can perform a search for a symptom with your app, or choose a diagnostic category that best describes the issue with your app. Next, there is a new feature called Risk Alerts that provides an actionable report to improve your App. Finally, this page is where you can find **Diagnostic Tools**. See [Diagnostic tools](#).

Note

If your app is down or performing slow, you can [**collect a profiling trace**](#) to identify the root cause of the issue. Profiling is light weight and is designed for production scenarios.

Diagnostic Interface

The homepage for App Service diagnostics offers streamlined diagnostics access using four sections:

- **Ask Genie search box**
- **Risk Alerts**
- **Troubleshooting categories**
- **Popular troubleshooting tools**

Ask Genie search box

The Genie search box is a quick way to find a diagnostic. The same diagnostic can be found through Troubleshooting categories.

Risk Alerts

The App Service diagnostics homepage performs a series of configuration checks and offers recommendations based on your unique application's configuration.

Recommendations and checks performed can be reviewed by clicking "View more details" link.

Troubleshooting categories

Troubleshooting categories group diagnostics for ease of discovery. The following are available:

- **Availability and Performance**
- **Configuration and Management**
- **SSL and Domains**
- **Risk Assessments**
- **Navigator (Preview)**
- **Diagnostic Tools**

The tiles or the Troubleshoot link show the available diagnostics for the category. If you were interested in investigating Availability and performance the following diagnostics are offered:

- **Overview**
- **Web App Down**
- **Web App Slow**
- **High CPU Analysis**
- **Memory Analysis**
- **Web App Restarted**
- **Application Change (Preview)**
- **Application Crashes**
- **HTTP 4xx Errors**
- **SNAT Failed Connection Endpoints**
- **SWAP Effects on Availability**
- **TCP Connections**
- **Testing in Production**
- **WebJob Details**

Diagnostic report

After you choose to investigate the issue further by clicking on a topic, you can view more details about the topic often supplemented with graphs and markdowns. Diagnostic report can be a powerful tool for pinpointing the problem with your app. The following is the Web App Down from Availability and Performance:

Resiliency Score

To review tailored best practice recommendations, check out the Resiliency Score Report. This is available as a downloadable PDF Report. To get it, simply click on the "Get Resilience Score report" button available on the command bar of any of the Troubleshooting categories.

Investigate application code issues (only for Windows app)

Because many app issues are related to issues in your application code, App Service diagnostics integrates with [Application Insights](#) to highlight exceptions and dependency issues to correlate with the selected downtime. Application Insights has to be enabled separately.

To view Application Insights exceptions and dependencies, select the **web app down** or **web app slow** tile shortcuts.

Troubleshooting steps

If an issue is detected with a specific problem category within the last 24 hours, you can view the full diagnostic report, and App Service diagnostics may prompt you to view more troubleshooting advice and next steps for a more guided experience.

Diagnostic tools

Diagnostics Tools include more advanced diagnostic tools that help you investigate application code issues, slowness, connection strings, and more. and proactive tools that help you mitigate issues with CPU usage, requests, and memory.

Proactive CPU monitoring (only for Windows app)

Proactive CPU monitoring provides you an easy, proactive way to take an action when your app or child process for your app is consuming high CPU resources. You can set your own CPU threshold rules to temporarily mitigate a high CPU issue until

the real cause for the unexpected issue is found. For more information, see [Mitigate your CPU problems before they happen](#).

Auto-healing

Auto-healing is a mitigation action you can take when your app is having unexpected behavior. You can set your own rules based on request count, slow request, memory limit, and HTTP status code to trigger mitigation actions. Use the tool to temporarily mitigate an unexpected behavior until you find the root cause. The tool is currently available for Windows Web Apps, Linux Web Apps, and Linux Custom Containers. Supported conditions and mitigation vary depending on the type of the web app. For more information, see [Announcing the new auto healing experience in app service diagnostics](#) and [Announcing Auto Heal for Linux](#).

Proactive auto-healing (only for Windows app)

Like proactive CPU monitoring, proactive auto-healing is a turn-key solution to mitigating unexpected behavior of your app. Proactive auto-healing restarts your app when App Service determines that your app is in an unrecoverable state. For more information, see [Introducing Proactive Auto Heal](#).

Navigator and change analysis (only for Windows app)

In a large team with continuous integration and where your app has many dependencies, it can be difficult to pinpoint the specific change that causes an unhealthy behavior. Navigator helps get visibility on your app's topology by automatically rendering a dependency map of your app and all the resources in the same subscription. Navigator lets you view a consolidated list of changes made by your app and its dependencies and narrow down on a change causing unhealthy behavior. It can be accessed through the homepage tile **Navigator** and needs to be enabled before you use it the first time. For more information, see [Get visibility into your app's dependencies with Navigator](#).

Change analysis for app changes can be accessed through tile shortcuts, **Application Changes** and **Application Crashes in Availability and Performance** so you can use it concurrently with other metrics. Before using the feature, you must first enable it. For more information, see [Announcing the new change analysis experience in App Service Diagnostics](#).

Azure storage account fundamentals

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

The CTO of your company, Tailwind Traders,

has tasked your team with

migrating all of your files to the cloud.

Your team has chosen Azure Storage,

which is a service that you can use to store files,

messages, tables, and other types of information.

Clients such as websites,

mobile apps, desktop applications,

and many other types of custom solutions,

can read data from and write data to Azure Storage.

Azure storage is also used by Infrastructure as

a Service Virtual Machines and

Platform as a Service Client Services.

To begin using Azure Storage,

you first create an Azure Storage account

to store your data objects.

You can create an Azure Storage account

by using the Azure portal,

PowerShell or the Azure CLI.

You should note that Azure Storage is not the same as Azure Database Services.

Your storage account will contain all of your Azure Storage data objects such as blobs, files and disks.

For example, by using storage accounts to store her files in the cloud, Sally will be able to access these files through unique namespace, using HTTP or HTTPS.

These files will be highly available and securely stored within this Azure Storage account.

Please note that Azure VMs, you use Azure Disk Storage to store virtual disks.

However, you can't use Azure Disk Storage to store a disk outside of a virtual machine.

A Storage account provides a unique namespace for your Azure storage data, that's accessible from anywhere in the world over HTTP or HTTPS.

Data in this account is secure, highly available, durable, and massively scalable.

For more information, you can refer to the Microsoft Azure product documentation on how to create a storage account.

Disk storage fundamentals

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using

arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Disk storage provides disks for Azure Virtual Machines.

Applications and other services can

access and use these disks as needed,

similar to how they would in on-premises scenarios.

Disk storage allows data to be persistently stored and accessed from an attached virtual hard disk.

Disks come in many different sizes and performance levels, from solid-state drives, SSDs, to traditional spinning hard disk drives, HDDs, with varying performance tiers.

You can use standard SSD and HDD disks for less critical workloads.

Premium SSD disks for mission-critical production applications, and ultra disks for data-intensive workloads such as SAP HANA, top-tier Databases, and transaction-heavy workloads.

Azure has consistently delivered enterprise-grade durability for Infrastructure as a servers disks with an industry-leading zero percent annualized failure rate.

An Azure Virtual Machine can use separate disks to store different data.

Azure Blob Storage fundamentals

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using

arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Azure Blob Storage is

an object storage solution for the Cloud.

It can store massive amounts of data,

such as text or binary data.

Azure Blob Storage is unstructured,

meaning that there are no restrictions

on the kinds of data it can hold.

Blob Storage can manage

a thousands of simultaneous uploads,

massive amounts of video data,

constantly growing log files,

and can be reached from

anywhere with an Internet connection.

Blobs aren't limited to common file formats.

A blob could contain gigabytes of

binary data streamed from a scientific instrument,

an encrypted message for another application,

or data in a custom format for an app you're developing.

One advantage of Blob Storage

over Disk Storage is that it

does not require developers to

think about or managed disks.

Data is uploaded as blobs and

Azure takes care of the physical storage needs.

Blob Storage is ideal for

storing up to eight terabytes

of data for virtual machines,

storing data for analysis by

an on-premises or Azure hosted service,

storing data for backup and restore,

disaster recovery, and archiving.

Streaming video and audio,

storing files for distributed access,

serving images or documents directly to a browser.

This diagram illustrates how you might use
Azure accounts, containers, and blobs.

Azure files fundamentals

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Azure Files offers fully managed file shares
in the Cloud that are accessible via
the industry standard server message block and
network file system preview protocols.

Azure file shares can be mounted concurrently by Cloud
or on-premises deployments of Windows, Linux, and macOS.

Applications running in Azure Virtual Machines
or Cloud services can mount
a file storage share to access
file data just as a desktop application would mount,
a typical SMB share.

Any number of Azure Virtual Machines or roles can
mount and access the file storage share simultaneously.

Typical usage scenarios would
be to share files anywhere in the world,
diagnostic data or application data sharing.
Use Azure Files for the following situations;
many on-premises applications use file shares.
Azure Files makes it easier to migrate
those applications that share data to Azure.

If you mount the Azure 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 a minimal if any changes.

Store configuration files on
a file share and access them for 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.

Write data to a file share and
process or analyze the data later.

For example, you might want to
do this with diagnostic logs,
metrics and crashed dumps.

This illustration shows Azure Files being used to
share data between two geographical locations.

Azure Files ensures the data is encrypted at rest,
and the SMB protocol
ensures that the data is encrypted in transit.

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 by using a URL that points to the file.

You can also use shared access signature or SAS
tokens to allow access to
a private asset for a specific amount of time.

A service SAS URI will show
the resource URI and the SAS token.

Understanding blob access tiers

Save note

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Data stored in the cloud can grow at an exponential pace.

To manage costs for your expanding storage needs, it's helpful to organize your data based on attributes like frequency of access and planned retention period.

Data stored in the cloud can be different based on how it's generated, processed and accessed over its lifetime.

Some data is actively accessed and modified throughout its lifetime.

Some data is accessed frequently early in its lifetime, with access dropping drastically as the data ages.

Some data remains idle in the cloud and is rarely if ever accessed after it's stored.

To accommodate these different access needs, Azure provides several access tiers, which you can use to balance your storage costs with your access needs.

Azure storage offers different access tiers for your blob storage, helping you store object data in the most cost effective manner.

The available access tiers include hot access tier, optimized for storing data that is accessed frequently, for example, images for your website.

Cool access tier, optimized for data that is infrequently accessed and stored for at least 30 days, for example, invoices for your customers.

Archive access tier, appropriate for data that is rarely accessed and stored for at least 180 days with flexible latency requirements, for example, long term backups.

Additionally, some considerations apply to the different access tiers.

Only the hot and cool access tiers can be set at the account level.

The archive access tier isn't available at the account level.

Hot, cool and archive tiers can be set up at the blob level during upload or after upload.

Data in the cool access tier can tolerate slightly lower availability, but still requires high durability, retrieval latency, and

throughput characteristics similar to hot data.

For cool data, a slightly lower availability service level agreement and higher access costs compared to hot data are acceptable tradeoffs for lower storage costs.

Archive storage stores data offline and offers the lowest storage costs, but also the highest cost to rehydrate and access data.

This illustration demonstrates choosing between the hot and cool access tiers on a general purpose storage account.

Azure Virtual network fundamentals

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Tailwind Traders has an on-premises data center

that you plan to keep,

but you want to use Azure to offload

peak traffic by using virtual machines hosted in Azure.

You want to keep your existing IP addressing scheme,

and network appliances while

ensuring that any data transfer is secure.

Using Azure Virtual Network for

your virtual networking can help you reach your goals.

Azure virtual networks enable

Azure resources such as VMs,

web apps, and databases,

to communicate with each other,

with users on the Internet,

and with your on-premises client computers.

You can think of an Azure network as a set of resources that links other Azure resources.

Azure Virtual Networks provide the following key networking capabilities:

Isolation and segmentation, Internet communications, communicate between Azure resources, communicate with on-premises resources, route network traffic, filter network traffic, and connect virtual networks.

Azure virtual networks

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

Now let's explore the networking capabilities provided by Azure isolation and segmentation.

Virtual Network allows you to create multiple isolated virtual networks.

When you set up a virtual network, you define a private IP Address space by using either public or private IP Address ranges.

You can divide that IP Address space into sub nets or allocate part of the defined address space to each named sub net.

For name resolution.

You can use the name resolution service that's built into azure.

You also can configure the virtual network to use either an internal or an external DNS's server.

Internet communications.

VM in Azure can connect to the Internet by default.

You can enable incoming connections from the Internet by defining a public IP

address or a public load balancer.

For VM management,

you can connect via the Azure Lai Remote desktop protocol or secure shell.

Communicate between Azure resources.

You'll want to enable azure resources to communicate securely with each other.

You could do that in one of two ways.

Virtual networks.

Virtual networks can connect not only VMS but

other Azure resources such as the APP service environment for power ups,

azure kubernetes, service on Azure virtual machine scale sets service endpoints.

You can use service endpoints to connect to other azure resource types,

such as as your sequel databases on storage accounts.

This approach enables you to link multiple azure resources to virtual networks

to improve security and provide optimal writing between resources.

Azure virtual networks enable you to link resources together in your on premises

environment and within your azure subscription.

In effect, you can create a network that spans both your local and

cloud environments.

There are three mechanisms for you to achieve this can activity.

The point to cite virtual private networks approach is like a virtual private network

connection that a computer outside your organization makes back into your

corporate network, except that it's working in the opposite direction.

In this case, the client computer initiates an encrypted VPN connection to

azure to connect that computer to the Azure Virtual network.

A site to site virtual private network links your on premises VPN device or

gateway to the azure VPN gateway in a virtual network.

In effect, the devices in Azure can appear as being on the local network.

The connection is encrypted and works over the Internet.

Azure Express right is the best approach for environment,

where you need greater bandwidth and even higher levels of security.

Express Right provides dedicated private connectivity toe azure that doesn't travel

over the Internet.

You'll learn more about express right later, as your virtual networks enable

you to filter traffic between sub net by using the following approaches.

Network security groups.

A network security group is an azure resource that can contain multiple inbound and outbound security rules.

You can define these rules to allow or block traffic based on factors such as source and destination IP address, port and protocol.

Virtual Appliances, a network virtual appliance is a specialized VM that can be compared to a hardened network appliance.

A network virtual appliance carries out a particular network function, such as running a firewall or performing a wide area network or one optimization.

Play video starting at :3:40 and follow transcript3:40

You can link virtual networks together by using virtual network peering.

Peering enables resources in each virtual network to communicate with each other.

These virtual networks could be in separate regions, which allows you to create a global, interconnected network through Azure.

You DR is user defined routing.

You'd is a significant update.

To Azure is virtual networks, as this allows network admin to control the routing tables between subnets within a subnet as well as between VPCs, thereby allowing for greater control over network traffic flow.

Azure virtual network settings

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

You can create and configure

Azure Virtual Network instances from the Azure portal,

Azure PowerShell on your local computer,

or Azure Cloud Shell.

When you create an Azure Virtual Network,
you can figure a number of basic settings.
You'll have the option to configure advanced settings,
such as multiple subnets,
Distributed Denial of Service or
DDoS Protection, and Service Endpoints.
You'll configure the following settings for
a basic virtual network: Network name.
The network name must be unique in your subscription,
but it doesn't need to be globally unique.
Make the name a descriptive one
that's easy to remember and
identified from other virtual networks. Address space.
When you set up a virtual network,
you define the internal address space in classes,
intra-domain routing, or CIDR format.
An address space in Azure is similar
to an on-premises IP addressing scheme.
This address space will be
unique within your subscription,
just as a subnet is within your on-premises environment.
You can then assign these address spaces
to your virtual networks,
ensuring there is no overlap or conflicts.
Subscription only applies if you
have multiple subscriptions to choose from.
Like any other Azure resource,
a virtual network needs to exist in a resource group.
You can either select an existing resource group,
or create a new one.
You will have an opportunity to select
the location where you want the virtual network to exist.
Within each virtual network address range,
you can create one or more subnets
that partition the virtual networks address space.

Routing between subnets will then depend on the default traffic routes.

You also can define custom routes.

Alternatively, you can define one subnet that encompasses all that a virtual networks address ranges.

Please note that subnet names must begin with a letter or number and end with a letter, number, or underscore.

They may contain only letters, numbers, underscores, periods, or hyphens.

There are two types of services available: Basic or Standard DDoS Protection.

Standard DDoS Protection is a premium service.

For more information on Standard DDoS Protection, see Azure DDoS Protection Standard Overview on the Azure product documentation.

You can also enable service endpoints.

You can select from a list of Azure service endpoints which ones you want to enable.

Options include: Azure Cosmos DB, Azure Service Bus, Azure Key Vault, and so on.

After you've completed the configuration of these settings, you are ready to create your Azure Virtual Network.

Now you can define additional settings.

After you create a virtual network, you can then define further settings.

These include: Network security groups, have security rules that enable you to filter the type of network traffic that can flow in and out of virtual network subnets, and network interfaces.

You create the network security group separately.

Then you associate it with the virtual network.

Azure automatically creates a route table for each subnet within an Azure Virtual Network

and add system default routes to the table.

You can add custom route tables to
modify traffic between virtual networks.

You can also amend the service endpoints.

After you've created a virtual network,
you can change any further settings on
the virtual network pane in the Azure portal.

Alternatively, you can use

PowerShell commands or commands
in Cloud Shell to make changes.

You can then review and change
settings and further sub-panes.

You can add additional address spaces
to the initial definition.

Under connected devices, use
the virtual network to connect machines.

You can also add additional subnets,
and under peerings you can link
virtual networks in peering arrangements.

You can also monitor and troubleshoot virtual networks,
or you can create
an automation script to
generate the current virtual network.

Virtual networks are
powerful and highly configurable mechanisms
for connecting entities in Azure.

You can connect Azure resources to one
another or to resources you have on-premises.

You can isolate, filter,
and route your network traffic.

Azure allows you to increase
security where you feel you need it.

Azure VPN gateway fundamentals

[Save note](#)
[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at :: and follow transcript0:00

A virtual private network, or VPN, is a type of private, interconnected network.

VPNs use an encrypted tunnel within another network.

They're typically deployed to connect two or more trusted private networks to one another over an untrusted network.

Typically, the public internet traffic is encrypted while traveling over the untrusted network to prevent eavesdropping or other attacks.

For our tailwind, trade of scenario VPNs can enable branch offices to share sensitive information between locations.

For example, let's say that your office is on the East Coast region of North America need to access your company's private customer data.

Which is stored on servers that are physically located in a West Coast region.

A VPN that connects your East Coast offices to your West Coast servers allows your company to securely access your private customer data.

A VPN gateway is a type of virtual network gateway.

Azure VPN gateway instances are deployed in azure virtual network instances on enable the following connectivity.

Connect on premises data centers to virtual networks through a site to site connection.

Connect individual devices to virtual networks through a point to site connection.

Connect virtual networks to other virtual networks through a network to network connection.

All transfer data is encrypted in a private tunnel as it crosses the internet.

You can deploy only one VPN gateway in each virtual network, but you can use one gateway to connect to multiple locations, which includes other virtual networks or on premises data centers.

When you deploy a VPN gateway, you specify the VPN type,

either policy based or right based.

The main difference between these two types of VPNs is how traffic to be encrypted is specified.

In Azure both types of VPN gateways use a pre shared key as the only method of authentication.

Policy-based VPNs gateways specify statically the IP address of packets that should be encrypted through each tunnel.

This type of device evaluates every data packet against those sets of IP addresses to choose the tunnel where that packet is going to be sent through.

Key features of policy based VPN gateways in Azure include support for IKEV1 only IKE or internet key exchange is a protocol used to set up a secure, authenticated communications channel between two parties.

Use of static writing where combinations of address prefixes from both networks control high traffic, is encrypted and decrypted through the VPN tunnel.

The source and destination of the tunnel networks are declared in the policy and don't need to be declared in writing tables.

Policy-based VPN must be used in specific scenarios that require them, such as for compatibility with legacy on premises VPN devices.

You can use right based gateways if defining which IP addresses are behind each tunnel is too cumbersome with route-based gateways IP SEC tunnels are modeled as a network interface or virtual tunnel interface.

IP writing either static rights or dynamic writing protocols, decides which one of these tunnel interfaces to use when sending each packet.

Right-based VPNs are the preferred connection method for on premises devices.

They're more resilient to topology changes such as the creation of new sub nets.

Use a right-based VPN gateway if you need any of the following types of connectivity.

Connections between virtual networks point-to-site connections.

Multisite connections, co existence with an Azure Express right gateway.

Key features of right-based VPN gateways in Azure include IKEv2 support use of any to any wild card traffic selectors use of dynamic writing protocols for writing, forwarding tables direct traffic to different IP SEC tunnels.

In this case, the source and destination networks aren't statically defined as they are in policy-based VPNs, or even in right based VPNs with static writing.

Instead, data packets are encrypted based on network writing tables that are created

dynamically using writing protocols such as Border Gateway Protocol or BGP.

The capabilities of your VPN gateway are determined by the skew or size that you deploy.

This table shows the main capabilities of each available skew.

Note that a basic VPN gateway should only be used for Dev test workloads.

In addition, it's unsupported to migrate from basic to the VPN w 1 2 3

SKU skews at a later time without having to remove the gateway and redeploy.

Before you can deploy a VPN gateway, you'll need some azure and on premises resources.

You'll need one of these azure resources before you can deploy an operational VPN gateway.

Deploy a virtual network with enough address space for the additional subnet that you'll need for the VPN gateway.

The address space for

this virtual network must not overlap with the on premises network that you'll be connecting to you can deploy only one VPN gateway within a virtual network.

Deploy a subnet called Gateway SubNet for the VPN gateway.

Use at least a forward /27 address mask to make sure you have enough IP addresses in the subnet for future growth, you can't use this subnet for any other services.

Create a basic skew dynamic public IP address If you're using a non zone aware gateway, this address provides a public rideable IP address as the target for your on premises VPN device.

This IP address is dynamic, but it won't change unless you delete and recreate the VPN gateway.

Play video starting at :5:48 and follow transcript5:48

Create a local network gateway to define the on premises networks configuration, such as where the VPN gateway will connect and what it will connect to.

The configuration includes the on premises VPN devices public IPv4 address on the on premises rideable networks.

This information is used by the VPN gateway to write packets that are destined for on premises networks through the IP SEC tunnel.

Create the virtual network gateway to write traffic between the virtual network on the on premises data center or other virtual networks.

The virtual network gateway could be either a VPN or express right gateway, but this unit only deals with VPN virtual network gateways.

You'll learn more about express right in a separate unit later in this module.

Create a connection resource to create a logical connection between the VPN gateway on the local network gateway.

The connection is made to the on premises VPN devices IPV4 address as defined by the local network gateway.

The connection is made from the virtual network gateway on its associated public IP address.

This diagram shows the combination of resources on their relationships to help you better understand what's required to deploy a VPN gateway.

To connect your data center to a VPN gateway there are some required on premises resources a VPN device that supports policy-based or right based VPN gateways.

A public-facing internet writable IP address.

There are several ways to ensure you have a fault tolerant configuration by default VPN gateways are deployed as two instances in an active stand by configuration.

Even if you only see one VPN gateway resource in azure when planned maintenance or unplanned disruption affects the active instance.

The standby instance automatically assumes responsibility for connections without any user intervention connections are interrupted during this fail over.

But they're typically restored within a few seconds for planned maintenance or within 90 seconds for unplanned disruptions.

With the introduction off support for the BGP writing protocol, you could also deploy VPN gateways in an active configuration.

In this configuration, you assign a unique public IP address to each instance.

You then create separate tunnels from the on premises device to each IP address.

You can extend the high availability by deploying an additional VPN device on premises.

Another high availability option is to configure a VPN gateway as a secure, fail over path for express right connections.

Express right circuits have resiliency built in, but they aren't immune to physical problems that affect the cables delivering connectivity or outages that affect the complete express right location.

In high availability scenarios where there's risk associated with an outage open express right circuit, you could also provision of VPN gateway that uses the internet as an alternative method of connectivity.

In this way, you can ensure there's always a connection to the virtual networks.

In regions that support availability zones, VPN, gateways and express right gateways could be deployed in a zone-redundant configuration.

This configuration brings a resiliency, scalability and higher availability to virtual network gateways.

Deploying gateways in azure availability zones physically and logically separates gateways within a region while protecting your on premises network connectivity to azure from zone level failures.

These gateways require different gateway, skews and use standard public IP addresses instead of basic public IP addresses.

Azure ExpressRoute fundamentals

[Save note](#)

[Transcript](#)[Notes](#)[Downloads](#)[Discuss](#)

Interactive Transcript - Enable basic transcript mode by pressing the escape key

You may navigate through the transcript using tab. To save a note for a section of text press CTRL + S. To expand your selection you may use CTRL + arrow key. You may contract your selection using shift + CTRL + arrow key. For screen readers that are incompatible with using arrow keys for shortcuts, you can replace them with the H J K L keys. Some screen readers may require using CTRL in conjunction with the alt key

Play video starting at ::1 and follow transcript0:01

ExpressRoute lets you exchange your on premises networks into the Microsoft cloud over a private connection with the help of a connectivity provider.

With ExpressRoute you can establish connections to Microsoft Cloud services such as Microsoft Azure on Microsoft 365.

Connectivity can be from any to any pipe VPN network, a point to point ethernet network or a virtual cross connection through a collectivity provider at a co location facility.

ExpressRoute connections don't go over the public Internet, this allows express right connections to offer more reliability, faster speeds, consistent latencies on higher security than typical connections over the Internet.

For information on how to connect your networks to Microsoft using ExpressRoute see express right connectivity models.

As part of your work for tailwind traders, you should understand what as your

ExpressRoute is and how it integrates with on premises and as your networks.

Play video starting at ::58 and follow transcript0:58

Now we will cover the benefits that ExpressRoute provides compared to other site to site connectivity options.

As a result, you'll learn whether express right could provide your company with the best possible network performance throughout.

This unit will focus on two different layers of the open systems interconnection or OSI model.

Layer 2 or L2 is the dazzling glare, which provides no to note communication between two nodes on the same network.

Layer 3 or L3 is the network layer, which provides addressing and writing between nodes on a multi node network.

There are several benefits to using ExpressRoute as the connection service between azure and on premises networks.

Layer 3 connectivity between your on premises network and the Microsoft Cloud through a connectivity provider.

Connectivity can be from any to any IP VPN network, a point to point ethernet connection or through a virtual cross connection via an ethernet exchange.

Connectivity to Microsoft cloud services across all regions in the geopolitical region.

Global connectivity to Microsoft services across all regions with the ExpressRoute premium add on.

Dynamic writing between your network on Microsoft via BDP, built in redundancy in every peering location for higher reliability, connection up time SLA and QOS support for Skype for business.

ExpressRoute provides layer three address level connectivity between your on premises network on the Microsoft Cloud through connectivity partners.

These connections could be from a point to point or any to any network, they can also be virtual cross connections through an exchange.

Each connectivity provider uses redundant devices to ensure that connection's established with Microsoft are highly available.

You can configure multiple circuits to complement this feature.

All redundant connections are configured with a layer three connectivity to meet service level agreements.

Play video starting at :3:4 and follow transcript3:04

ExpressRoute enables connectivity to Microsoft Cloud Services that includes direct access to the following services in all regions.

Microsoft Office 365, Microsoft Dynamics 365,

Azure compute services such as Azure virtual machines,

Azure cloud services such as Azure Cosmos DB and Azure Storage.

Office 365 was created to be accessed securely and reliably via the Internet.

For this reason, we recommend the use of ExpressRoute for specific scenarios.

You can enable express right global reach to exchange data across your on premises sites by connecting your ExpressRoute circuits.

For example, assume that you have a private data center in

California connected to ExpressRoute in Silicon Valley.

You have another private data center in Texas connected to ExpressRoute in Dallas.

With ExpressRoute global reach,

you can connect your private data centers through to ExpressRoute circuits.

Your cross data center traffic will travel through the Microsoft Network

.ExpressRoute uses the Border Gateway Protocol or be GP writing protocol.

BGP is used to exchange routes between on premises networks on resources running in azure.

This protocol enables dynamic routing between your on premises network and services running in the Microsoft Cloud.

ExpressRoute supports three models that you can use to connect your on premises network to the Microsoft Cloud.

Cloud Exchange Co-location point to point Ethernet connection any to any connection.

Co-located providers can normally offer both Layer 2 and

Layer 3 connections between your infrastructure,

which might be located in the co location facility on the Microsoft Cloud.

For example, If your data center is co-located at a cloud exchange such as an eye ESP, you can request a virtual cross connection to the Microsoft Cloud.

Point-to-point connections provide layer 2 and

layer 3 connectivity between your on premises site and azure.

You can connect your offices or

data centers to azure by using the point to point links.

For example, if you have an on premises data center,

you can use a point to point ethernet link to connect to Microsoft.

With any-to-any connectivity, you can integrate your wide area network with

azure by providing connections to your offices and data centers.

Azure integrates with your WAN connection to provide a connection like you would have between your data center and any branch offices.

With any-to-any connections, all WAV providers offer layer three connectivity.

For example, if you already use multi protocol label switching to connect to your branch offices or other sites in your organization on ExpressRoute connection to Microsoft behaves like any other location on your private WAN.

With ExpressRoute, your data doesn't travel over the public Internet, so it's not exposed to the potential risks associated with Internet communications.

ExpressRoute is a private connection from your own premises infrastructure to your other infrastructure, even if you have an ExpressRoute connection.

DNA's Queries Certificate Revocation list Checking on as
your content delivery network requests are still sent over the public Internet