Microsoft Azure Fundamentals

THIRD EDITION

Exam Ref AZ-900

Jim Cheshire

Microsoft

Exam Ref AZ-900 Microsoft Azure Fundamentals

Third Edition

Jim Cheshire

Exam Ref AZ-900 Microsoft Azure Fundamentals, Third Edition

Published with the authorization of Microsoft Corporation by:

Pearson Education, Inc.

ISBN-13: 978-0-13-795514-5
ISBN-10: 0-13-795514-6

Library of Congress Control Number: On file

ScoutAutomatedPrintCode

EDITOR-IN-CHIEF
Brett Bartow

EXECUTIVE EDITOR
Loretta Yates

SPONSORING EDITOR
Charvi Arora

DEVELOPMENT EDITOR
Rick Kughen

MANAGING EDITOR
Sandra Schroeder

SENIOR PROJECT EDITOR
Tracey Croom

COPY EDITOR
Rick Kughen

INDEXER
Valerie Haynes Perry

PROOFREADER
Dan Foster

TECHNICAL EDITOR
Tim Warner

EDITORIAL ASSISTANT
Cindy Teeters

COVER DESIGNER
Twist Creative, Seattle

COMPOSITOR
Danielle Foster

Pearson's Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity of all learners. We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, socioeconomic status, ability, age, sexual orientation, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where:

- Everyone has an equitable and lifelong opportunity to succeed through learning.
- Our educational products and services are inclusive and represent the rich diversity of learners.
- Our educational content accurately reflects the histories and experiences of the learners we serve.
- Our educational content prompts deeper discussions with learners and motivates them to expand their own learning (and worldview).

While we work hard to present unbiased content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.

Please contact us with concerns about any potential bias at *https://www.pearson.com/report-bias.html*.

Contents at a glance

Contents

Acknowledgments

I'd like to express my deep gratitude to the following people, without whom this book would not have been possible.

Thank you to Loretta Yates for bringing me into this project. After two decades of working together on numerous projects, you still seem to find a way to bring freshness and excitement to each one. Thank you, Rick Kughen, for painstakingly editing every corner of this book to make it a better reading experience. Thanks to Tim for all the times you made me take a second look at my approach and for adding real value with your ideas. Thanks to Charvi Arora for taking care of all the details that keep everything on track. Finally, thank you to all the people at Microsoft Press who worked so hard to create this book from the digital manuscript.

About the author

Jim Cheshire is a technology enthusiast with more than 25 years of experience in various roles within IT. Jim has authored more than 15 books on technology, and he's held numerous training sessions on Microsoft Azure, both in private enterprises and through Safari's Live Training program. You can follow Jim and interact with him on LinkedIn at https://www.linkedin.com/in/jimcheshire. Jim is in his 24th year at Microsoft, and he's currently focused on the technical skilling strategy for Microsoft's developer offerings, Microsoft Azure, and Microsoft Windows.

Introduction

Both businesses and individuals are adopting cloud technologies at a breakneck pace, and Microsoft Azure is often the choice for cloud-based applications and services. The purpose of the AZ-900 exam is to test your understanding of the fundamentals of Azure. The exam includes high-level concepts that apply across all of Azure to important concepts specific to particular services of Azure. Like the exam, this book is geared toward giving you a broad understanding of Azure itself as well as many common services and components in Azure.

While we've made every effort possible to make the information in this book accurate, Azure is rapidly evolving. There's a chance that some of the screens in the Azure portal are slightly different now than when this book was written. It's also possible that other minor changes have taken place, such as minor name changes in features and so on.

In this edition of the book, we've meticulously reviewed the content in the first two editions and updated everything to reflect the current state of Azure. We've also reorganized the book and added new content to reflect the current state of the AZ-900 exam. Microsoft has recently added new concepts, services, and Azure features to the AZ-900 exam, and we've added those to this edition. We've also corrected a few things and made quite a few changes based on reader feedback from the first two editions.

This book covers every major topic area found on the exam, but it does not cover every exam question. Only the Microsoft exam team has access to the exam questions, and Microsoft regularly adds new questions to the exam, making it impossible to cover specific questions. You should consider this book a supplement to your relevant real-world experience and other study materials. In many cases, we've provided links in the "More Info" sections of the book, and these links are a great source for additional study.

Organization of this book

This book is organized by the "Skills measured" list published for the exam. The "Skills measured" list is available for each exam on the Microsoft Learn website: *http://microsoft.com/learn*. Each chapter in this book corresponds to a major topic area in the list, and the technical tasks in each topic area determine a chapter's organization. Because the AZ-900 exam covers three major topic areas, this book contains three chapters.

Preparing for the exam

Microsoft certification exams are a great way to build your resume and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. Although there is no substitute for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. We recommend that you augment your exam preparation plan by using a combination of available study materials and courses. For example, you might use the Exam Ref and another study guide for your "at-home" preparation and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you.

Note that this Exam Ref is based on publicly available information about the exam and the author's experience. To safeguard the integrity of the exam, authors do not have access to the live exam.

Microsoft certifications

Microsoft certifications distinguish you by proving your command of a broad set of skills and experience with current Microsoft products and technologies. The exams and corresponding certifications are developed to validate your mastery of critical competencies as you design and develop or implement and support solutions with Microsoft products and technologies both on-premises and in the cloud. Certification brings a variety of benefits to the individual and to employers and organizations.

MORE INFO **ALL MICROSOFT CERTIFICATIONS**

For information about Microsoft certifications, including a full list of available certifications, go to *http://www.microsoft.com/learn*.

Quick access to online references

Throughout this book are addresses to webpages that the author has recommended you visit for more information. Some of these links can be very long and painstaking to type, so we've shortened them for you to make them easier to visit. We've also compiled them into a single list that readers of the print edition can refer to while they read.

Download the list at *MicrosoftPressStore.com/ExamRefAZ9003e/downloads*

The URLs are organized by chapter and heading. Whenever you come across a URL in the book, find the hyperlink in the list to go directly to the webpage.

Errata, updates & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

MicrosoftPressStore.com/ExamRefAZ9003e/errata

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit *MicrosoftPressStore.com/Support*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

CHAPTER 1

Describe cloud computing

Cloud computing has been part of information technology (IT) for more than 20 years. During that time, it has evolved into a complex collection of cloud services and cloud models. Before you begin the process of moving to the cloud, it's important that you understand key concepts and services related to the cloud.

There are many reasons for moving to the cloud, but one of the primary benefits is removing some of the IT burden from your company. The cloud allows you to take advantage of a cloud provider's infrastructure and investments, and it makes it easier to maintain consistent access to your applications and data. You'll also gain the benefit of turnkey solutions for backing up data and ensuring your applications can survive disasters and other availability problems. Hosting your data and applications in the cloud is often more cost effective than investing in infrastructure and on-premises IT resources.

Once you decide that you want to take advantage of all the cloud offers, you need to understand the different cloud models and which is the right choice for you. You also need to understand the different types of cloud services and the benefits and drawbacks of each.

This chapter covers cloud computing in general, describes the benefits of the cloud, and discusses the different cloud models and service types that are available in cloud computing.

Skills covered in this chapter:

- Describe cloud computing
- Describe the benefit of using cloud services
- Describe cloud service types

Skill 1.1: Describe cloud computing

The concept of cloud computing is quite a bit more complex than many people realize. Many people believe moving to the cloud means not having to manage any IT infrastructure or systems, but that's not necessarily true. Whether you are operating on-premises or in the cloud, you'll likely be responsible for at least part of your application or system.

It may also surprise you that the cloud offers several different models, and these models allow the flexibility to keep using on-premises systems while still using the cloud. You can even use the cloud in scenarios where you are completely disconnected from the internet!

This section covers:

- Cloud computing
- Shared responsibility model
- Cloud models
- The consumption-based model
- Comparing cloud models

Cloud computing

Before we go any further, let's first agree on what we mean by *cloud computing*. Let's begin by saying that cloud computing means that we're using computers and other infrastructure running applications that are accessible over the internet. That's probably a fair starting point, but it certainly doesn't clearly differentiate cloud computing from other types of computing. What if we say that cloud computing means that the previously mentioned computers are all connected by a network? I think that probably gets us closer (although that's not technically a requirement in cloud computing), but to settle on a solid definition for cloud computing, we need to add a few other components.

Cloud computing means there is a cloud provider involved. If you're using Microsoft Azure, that provider is Microsoft. Your cloud provider takes on some of the responsibility for cloud resources. Cloud computing also provides financial benefits because you can pay for only what you need versus having to invest heavily in infrastructure, IT staff, and so forth.

We'll explore some other aspects that cloud computing represents later in this chapter, but let's first dig into the concepts I just introduced, starting with the shared responsibility model.

Shared responsibility model

Let's consider a typical on-premises scenario for hosting an application used by multiple people in a company. To host the application, we first need some computers running server software. Depending on the complexity of the application, we might also need a web server, a database server, and so forth. We also need to connect all these computers to our network, and that requires us to have network infrastructure like routers, switches, network cables, and so on. All this infrastructure isn't going to manage itself, so we also need an IT department, some support personnel, database and web administrators, and network professionals.

Once our application is up and running, we're responsible for troubleshooting and correcting any problems. Day or night, we likely need someone available to jump into action if

things go awry. Not only is this a huge responsibility, but it comes at great expense. The cost of infrastructure alone can be enormous. Add the payroll expenses of all the employees you need, and you're dealing with some serious money.

This kind of scenario is exactly why many companies are moving to the cloud. When you move to the cloud, the cloud provider takes on some of the responsibility for you. For example, if we use virtual machines in the cloud instead of having on-premises servers, we can shift the responsibility of the network infrastructure that connects our computers to the cloud provider. We are still responsible for the operating system and our application running on the computers, but the cloud provider takes on the responsibility for everything else.

Depending on what type of cloud service we choose, we may be able to shift even more responsibility to the cloud provider. For example, if we choose to use a database offering from our cloud provider instead of running our own database server, we might be able to shift responsibility for the configuration and performance of the database server to the cloud provider, saving ourselves money and the headache of managing the server.

When you move to the cloud, you share responsibility with your cloud provider. How much responsibility you shift to the provider depends on what type of cloud service you're using, but the responsibility for any cloud deployment is always shared between you and the cloud provider.

Cloud models

The decision to move to the cloud can be a complicated one. You may have legacy systems that can't be moved to the cloud, or you might have some security requirements that restrict what you can deploy to the cloud. These are just two of the many scenarios that companies face when considering the cloud.

Fortunately, the cloud makes it easier to deal with these challenges by offering three different cloud models: the public cloud, the private cloud, and the hybrid cloud.

The public cloud

The most common cloud model is the public cloud. In a public cloud model, you use shared infrastructure that is accessible on a public network. The network, storage, and virtual machines (VMs) that your application uses are provided by a cloud provider and shared between all consumers of the public cloud. Microsoft Azure, Amazon Web Services (AWS), and Google Cloud are examples of public clouds.

> ***NOTE*** **CLOUDS AND THE INTERNET**
>
> **Many cloud services might provide access from the internet, but that doesn't necessarily mean they are available to anyone on the internet. In most cases, access requires authentication.**
>
> **You'll learn more about securing cloud resources in Chapter 2, Skill 2.4, "Describe Azure identity, access, and security."**

The public cloud model is beneficial because it makes it easy and fast to move to the cloud. Because the cloud provider already has the infrastructure in place and configured for you, all you must do is decide on the type of cloud service you want and you're off and running.

Another advantage to the public cloud model is that you can control costs more efficiently because you only pay for the resources you are using. If you need to add more VMs, the cloud provider has them available and waiting for you. You don't have to maintain a pool of resources yourself. Instead, you take advantage of the resources the cloud provider has invested in.

IMPORTANT **MULTITENANT ENVIRONMENT**

Because you are sharing resources in a public cloud with other people who are using that public cloud, you'll often see public clouds referred to as a *multitenant environment*.

MORE INFO **MORE INFORMATION ON PUBLIC CLOUDS**

For more information on public clouds and Azure, see *https://bit.ly/az900-publiccloud*.

The private cloud

The private cloud model provides many of the attractive benefits of the cloud in a private environment that is dedicated to a single company. A private cloud can be hosted in an on-premises environment, but it can also be hosted on a third-party hosting provider.

IMPORTANT **SINGLE-TENANT ENVIRONMENT**

Because the resources in a private cloud are dedicated to a single organization, you will often see the private cloud referred to as a *single-tenant environment*.

Two of the main reasons why companies choose a private cloud are privacy and regulatory concerns. Unlike the public cloud, private clouds operate on a private network that is only accessible by a single organization.

You'll often hear that a private cloud consists of infrastructure that is owned by an individual company, but that's not always true. If a company runs a private cloud on-premises, they will usually own the hardware and infrastructure used for the private cloud, but it's also possible to host a private cloud in a third-party data center. In that situation, the infrastructure is owned by the hosting provider, but it's still completely dedicated to the single company paying for the private cloud. The bottom line is that the difference between a public and a private cloud is the privacy of infrastructure and data. It doesn't really matter who owns the infrastructure.

MORE INFO **MORE INFORMATION ON PRIVATE CLOUDS**

For more information on private clouds, see *https://bit.ly/az900-privatecloud*.

The hybrid cloud

As you might expect, hybrid clouds are a mixture of public and private clouds. In a hybrid cloud environment, you might have an application that is running within the public cloud, yet it accesses data that is securely stored on-premises. You might also have a scenario where your application and most of its resources are located on a private cloud, but you want to use services or infrastructure that are in a public cloud. Indeed, the various scenarios that are suitable for a hybrid model are almost endless.

Hybrid cloud models are often a company's first foray into the cloud. Many companies have legacy on-premises systems that are expensive to move to the cloud, yet they might want to take advantage of some of the benefits of the cloud. In such a scenario, a company might move only part of a particular system to the cloud, leaving the legacy system on-premises until a later time.

Not all companies adopting a hybrid cloud model are doing so because of legacy systems. In some situations, a company might want to maintain complete control over part of their infrastructure or data. They might decide to build out on-premises infrastructure in tandem with building their public cloud presence.

> ***IMPORTANT*** **HYBRID DOESN'T ALWAYS INCLUDE ON-PREMISES**
>
> **Remember, a private cloud is a cloud dedicated to a single organization. It doesn't have to be located on-premises. It can also be hosted at a third-party data center, so a hybrid cloud model might be the combination of a third-party data center and a public cloud.**

The consumption-based model

Cloud providers take savings a step further by offering the ability to use only those computing resources you require at any particular time. This is typically referred to as a *consumption-based model*, and it's often applied at many levels in cloud computing.

The cloud allows you to scale your application to use only the number of VMs you need, and you can choose how powerful those VMs are. You can adjust their number and power as your needs require. However, many cloud providers also offer services that allow you to pay only for the time that you consume computer resources. For example, you can have application code hosted in a cloud provider and pay only for the time that the code is executing on a VM. When no one is using the application, you don't pay for any resources.

> ***MORE INFO*** **CONSUMPTION-BASED COMPUTING**
>
> **For an example of a consumption-based model, see "*Compute types*" in Chapter 2, Skill 2.2, "Describe Azure compute and networking services."**

As you can see, the cloud model offers many economic benefits over the on-premises model, and that's just one reason why businesses are rapidly moving to the cloud.

Comparing cloud models

Now that you know about cloud models and some other benefits of the cloud, how do you decide which cloud model is best for any given scenario? Each model has some advantages and disadvantages, so let's compare each model.

As stated earlier, the public cloud is the most popular cloud model. In the public cloud, you can shift the maximum amount of responsibility to the cloud provider, and you can also benefit financially by taking advantage of the consumption-based model.

While the flexibility and convenience of the public cloud is attractive, it comes with some disadvantages. First, you give up some control of the infrastructure when using the public cloud. The amount of control you give up depends on the type of cloud service you choose, but no matter what, the cloud provider will control some portion of your infrastructure.

There might also be security concerns with operating in the public cloud. The network involved in the public cloud is the public internet, and it's available to anyone with an internet connection. That means you will need to have security measures in place to avoid unauthorized access to your application and data. Cloud providers realize this, and they provide security measures to help protect you, but those measures might not meet your security requirements.

Another disadvantage of the public cloud is that it locks you into the specific configuration defined by the cloud provider. For example, suppose you have an application that needs a large amount of disk storage, but you only need a single CPU system to run it. In order to meet your disk space requirements, the cloud provider might require you to use a high-powered, multi-CPU VM, thereby increasing your costs unnecessarily.

Some of the disadvantages of the public cloud can be offset by choosing the private cloud, especially where security is concerned. Businesses like banks and medical providers might have regulations in place that require certain data to be inaccessible from the internet, and in those situations, a private cloud might be a good choice.

Another common consumer of private clouds is the cruise ship industry. Cruise ships operate in remote areas where internet access isn't always available, but they still want to take advantage of the benefits of the cloud for day-to-day operations of complex ship systems.

There are also some disadvantages to a private cloud. If you are hosting your private cloud on-premises, you will likely spend as much on IT as you would in a non-cloud environment. You will have to pay for hardware and virtualized systems for your cloud, and you'll need IT staff who can manage the software and infrastructure for your cloud.

Avoiding IT costs is one of the primary reasons that companies choose to use a third-party hosting provider for private clouds, but that choice also has some drawbacks. For example, once you offload management of your private cloud to a third party, you lose control of important considerations such as the security of your data. It's often impossible to achieve full transparency when dealing with third-party providers, and you can't always guarantee that data on your private cloud network will remain secured in a way that you require.

Because of some of the challenges I've outlined, companies often choose the hybrid cloud. When companies adopt a hybrid model, they often require the capability of connecting the

private, on-premises network with the public cloud network. Cloud providers offer many technologies to make that possible. In Microsoft Azure, Virtual Networks, Hybrid Connections, and Service Bus are just some examples of such technologies.

MORE INFO **MORE INFORMATION ON AZURE NETWORKING**

We'll cover more features of Azure networking in Chapter 2, Skill 2.2.

While it might not be immediately obvious, a hybrid cloud model comes with several challenges. First, application development teams will need to ensure that data shared between the public and private cloud is compatible. This might require some specialized development skills and complex troubleshooting. The networking complexities in a hybrid environment can also be quite challenging, especially because network infrastructure at third-party providers might introduce problems that are difficult to troubleshoot. Finally, spreading application resources between a public and a private cloud might cause application slowdowns due to the geographical distance between systems running the application and the data the application uses. All these situations must be carefully evaluated when deciding to use a hybrid cloud model.

In order to make hybrid cloud easier for its customers, Microsoft provides Azure Stack. Azure Stack is sold as a package, including software and validated hardware to run it. Azure Stack allows you to run Azure services on-premises, making it easy to then transfer applications to the cloud with a minimal amount of work. Because the hardware is part of Azure Stack and has been validated by Microsoft, you don't have the burden of attempting to determine hardware needs in order to deploy Azure Stack, but you do have to manage the on-premises hardware.

Skill 1.2: Describe the benefit of using cloud services

Today's companies rely heavily on software solutions and access to data. In fact, in many cases, a company's most valuable assets are directly tied to data and applications. Because of that, investment in IT has grown tremendously over the past couple of decades. Reliance in on-premises IT departments worked well in the early days of IT, but access to data and applications has become such a critical part of day-to-day operations that localized IT systems have become inefficient on many levels.

When making decisions about what to move to the cloud, evaluate your decisions against the benefits that cloud computing can provide.

This section covers:

- High availability and scalability
- Reliability and predictability
- Security and governance
- Manageability in the cloud

High availability and scalability

The availability of data and applications is a core requirement for any application, whether it is on-premises or in the cloud. If your data or application isn't available to you, nothing else matters. There are many reasons why you might lose availability, but the most common issues are:

- A network outage
- An application failure
- A system outage (such as a virtual machine outage)
- A power outage
- A problem with a reliant system, such as an external database

In a perfect world, you experience 100 percent availability, but if any of the above problems occur, that percentage will begin to decrease. Therefore, it's critical that your infrastructure minimizes the risk of problems that affect the availability of your application.

Cloud providers offer a *service-level agreement* (SLA) that guarantees a certain level of availability as a percentage. An SLA will usually guarantee an uptime of close to 100 percent, but it only covers systems that are controlled by the cloud provider.

An application hosted in the cloud might be one that is developed by your company, but it can also be one provided to you by the cloud provider.

Network outage

All applications require some level of network connectivity. Users of an application require network connectivity to the computers that run the application. The application requires network connectivity to required back-end systems such as database servers. Applications might also call into other applications using a network. If any of these network connections fail, they can cause a lack of availability.

> ***MORE INFO*** **PLANNING FOR NETWORK OUTAGES**
>
> **A network failure doesn't have to mean that your application or data is unavailable. If you plan carefully, you can often avoid an application problem when a network problem occurs. We'll cover that in more detail when we discuss fault tolerance later in this chapter.**

Cloud providers invest a lot of money in network infrastructure, and by moving to the cloud you gain the benefit of that infrastructure and the additional reliability that comes with it. If something within that infrastructure fails, the cloud provider diagnoses and fixes it, often before you even realize there's a problem.

Application failure

An application failure is often the result of a software bug, but it can also be caused by application design.

> ***MORE INFO*** **APPLICATION DESIGN AND THE CLOUD**
>
> **You don't need to understand application design concepts for the AZ-900 exam, but if you're interested in learning more about application design and the cloud, Microsoft has a good reference at *https://bit.ly/cloudappdesign*.**

In some cloud scenarios, you are still responsible for application failures, but your cloud provider likely provides you with tools that you can use to diagnose these failures more easily. For example, Azure offers a service called Application Insights that integrates with your application to give you detailed information about the performance and reliability of your application. Application developers can often use this information to get right to the code where a problem is happening, dramatically reducing the time needed for troubleshooting.

Cloud providers offer other features that can reduce availability problems caused by application failure. You can often test new versions of an application in a protected environment without affecting real users. When you're ready to move actual users to a new version, you can often move a small number of users first to ensure things are working correctly. If you discover problems, the cloud often makes it easy to roll things back to the prior version.

System outage

A system outage occurs when the computer running a particular system becomes unavailable. In the on-premises world, that computer might be a server running a database or another part of the application. In the cloud, these systems run inside of *virtual machines*, or VMs.

VMs are software-based computers that run on a physical computer. A single computer can run multiple VMs, and each VM has its own isolated operating system and applications. All VMs running on a computer share the CPU, memory, and storage of the host computer they run on.

> ***NOTE*** **VMs AREN'T JUST FOR THE CLOUD**
>
> **VMs make it easy to add additional computers when necessary, and they allow you to better manage computer resources such as CPU, disk space, and memory. For that reason, VMs are commonplace in most businesses.**

Depending on the cloud service you choose, you might or might not be responsible for maintaining VMs. However, whether you or your cloud provider maintains them, the cloud provider will constantly monitor the health of VMs and will have systems in place to recover an unhealthy VM.

Power outage

Reliable electricity is critical to availability. Even a quick power flicker can cause computers to reboot and systems to restart. When that happens, your application is unavailable until all systems are restored.

Cloud providers invest heavily in battery-operated power backups and other redundant systems in order to prevent availability problems caused by power outages. In a situation where a large geographic area is affected by a power outage, cloud providers offer you the ability to run your application from another region that isn't affected.

Problems with a reliant system

Your application might use systems that aren't in the cloud or that are hosted by a different cloud provider. If those systems fail, you might lose availability. By hosting your application in the cloud, you gain the benefit of troubleshooting, alerting, and diagnostics tools that the cloud provider offers.

Other examples of a reliant system are the VMs running your application, the web server that's hosting your website, and so on. If something on your VM uses all the CPU resources on the machine or if your web server receives more traffic than it can handle, your application can become unavailable. In these situations, *scaling* can avoid a problem.

Scalability

Computing resources aren't free. Even if you're using virtual machines, the underlying resources such as disk space, CPU, and memory cost money. The best way to minimize costs is to use only the resources necessary for your purposes. The challenge is that resource needs can change often and quickly.

Consider a situation where you are hosting an application in the cloud that tracks sales data for your company. If your sales staff regularly enters information on daily sales calls at the end of the day, you might need additional computing resources to handle that load. Those same resources aren't needed during the day when the sales staff is making sales calls and not using the application.

You might also host a web application in the cloud that is used by external customers. Depending on the usage pattern, you might want to add additional computing resources on certain days or during certain times. You might also need to quickly adapt to more users if your company receives unexpected publicity from the media or some other means.

Scaling allows you to easily deal with these kinds of scenarios. Scaling is the process of adding additional resources or additional power for your application. There are two variations of scaling: horizontal scaling (often referred to as *scaling out*) and vertical scaling (often referred to as *scaling up*).

When you scale out, you add additional VMs for your application. Each VM you add is identical to other VMs servicing your application. Scaling out provides additional resources to handle additional load.

When you scale up, you move to a new VM with additional resources. For example, you might determine that you need a more powerful CPU and more memory for your application. In that case, scaling up will allow you to move your application to a more powerful VM.

> ***NOTE*** **SCALING UP OFTEN ADDS FEATURES**
>
> **When you scale up, you often not only add more CPU power and memory, but you also often gain additional features because of the added power. For example, scaling up might give you solid-state disk drives or other features not available at lower tiers.**

Figure 1-1 shows an example of scaling up a web application hosted in Azure.

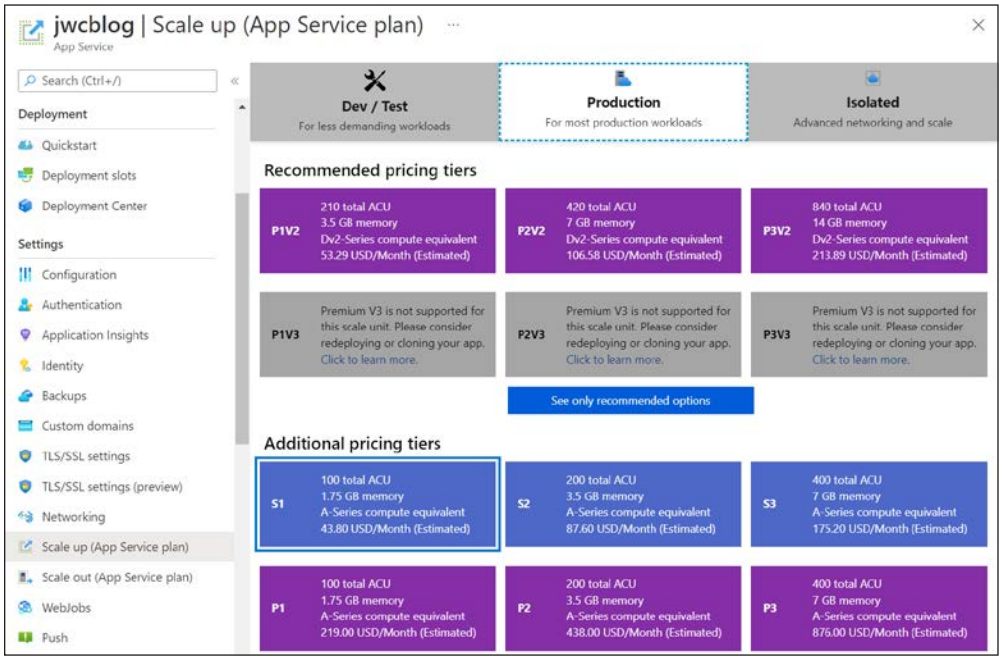


FIGURE 1-1 Scaling up a web application in Azure

> ***REAL WORLD*** **SCALING GOES BOTH WAYS**
>
> **In addition to scaling out and scaling up, you can also *scale in* and *scale down* to decrease resource usage. In a real-world situation, you would want to increase computing resources when needed and reduce them when demand goes down. The ease of scaling in both directions is often referred to as *elasticity*.**

NOTE **THE AZURE PORTAL**

Microsoft changes the Azure portal frequently, so what you see in the Azure portal may differ slightly from the figures shown in this book.

Cloud providers make it easy to scale your application, and they offer the ability to scale automatically based on the usage pattern for your application. You can scale automatically based on things like CPU usage and memory usage, and you can also scale based on other metrics that are specific to the type of application. The concept of automatically scaling is referred to as *elasticity*.

EXAM TIP

In Azure, you can scale automatically by configuring Auto-Scale. Auto-Scale is an Azure service that can automatically scale applications running in many Azure services based on usage patterns, resource utilization, time of day, and much more.

One major benefit of the cloud is that it allows you to quickly scale. For example, if you are running a web application in Azure and you determine that you need two more VMs for your application, you can scale out to three VMs in seconds. Azure takes care of allocating the resources for you. All you must do is tell Azure how many VMs you want, and you're up and running. This kind of speed and flexibility in the cloud is often called cloud *agility*.

MORE INFO **MORE INFORMATION ON SCALING BEST PRACTICES**

For more information on scaling in Azure, see the documentation at *https://docs.microsoft.com/azure/architecture/best-practices/auto-scaling*.

Reliability and predictability

In a complex cloud environment, things are bound to go wrong from time to time. To maintain a high level of availability, cloud providers implement systems that monitor the health of cloud resources and take action when a resource is determined to be unhealthy, thereby ensuring that the cloud is *fault tolerant*.

EXAM TIP

Don't confuse fault tolerance with scaling. Scaling allows you to react to additional load or resource needs, but it's always assumed that all the VMs you are using are healthy. Fault tolerance happens without any interaction from you, and it's designed to automatically move you from an unhealthy system to a healthy system if things go wrong.

In addition to monitoring the health of VMs and other resources, cloud providers design their infrastructure in such a way as to ensure fault tolerance. For example, if you have an application running on two VMs in Azure, Microsoft ensures that those two VMs are allocated within the infrastructure so that they are unlikely to be affected by system failures.

MORE INFO **FAULT TOLERANCE IN AZURE**

You don't have to understand the technical details of how Azure implements fault tolerance for the AZ-900 exam, but if you're interested in learning more, check out *https://msdnmicrosoft.com/magazine/mt422582.aspx*.

Fault tolerance is designed to deal with failure on a small scale; for example, fault tolerance can move you from an unhealthy VM to a healthy VM. However, there are times when much larger failures can occur. For example, natural disasters in a region can affect all resources in that region. Not only can something like that impact availability, but without a plan in place, disasters can also mean the loss of valuable data.

REAL WORLD **DISASTER RECOVERY AND GOVERNMENTS**

Depending on what kind of data you store, you might be required to have a disaster recovery plan in place. Cloud providers typically comply with standards imposed by laws such as the Health Insurance Portability and Accountability Act (HIPAA), and they often provide compliance tools you can use to ensure compliance. You'll learn more about compliance and Azure in Chapter 3, "Describe Azure management and governance."

Disaster recovery not only means having reliable backups of important data, but it also means that the cloud infrastructure can replicate your application's resources in an unaffected region so that your data is safe, and your application availability isn't affected. Disaster recovery plans are commonly referred to as *Business Continuity and Disaster Recovery* (BCDR) plans, and most cloud providers have services that can help you develop and implement a plan that works for your needs.

By taking advantage of the fault tolerant systems and implementing a disaster recovery plan, you can realize the predictability afforded by the cloud. These systems are in place to provide a high level of reliability for your cloud deployments, and they represent one of the major benefits of moving to the cloud.

Security and governance

Computing environments are fraught with security threats of all kinds. Once you connect computers to the internet, these threats increase exponentially. In a cloud model, these threats can impact both you and the cloud provider. For that reason, cloud providers invest significantly in the security of cloud services.

Cloud providers carefully monitor the networks and systems used by the cloud, and when security threats are identified, they are usually immediately and effectively eliminated. In most situations, you don't have to do anything to benefit from this level of protection. It's a default benefit of using the cloud. In situations where you do need to take some action for the sake of security, cloud providers will often provide tools and resources to make things as simple as possible for you.

Cloud providers will also offer governance features and services that make it easy to control who can access your resources and what level of access they have. That governance applies not only to access of resources but also to how users use those resources. For example, you can define policies that prevent users from creating resources that carry significant costs.

Manageability in the cloud

As we've already discussed, moving to the cloud offsets some of your responsibility for management of computing resources, but you do still share some of the responsibility. Managing resources can be complicated by moving to the cloud because you no longer have direct access to the infrastructure. Cloud providers offset this complication by providing tools to make managing cloud resources easier.

Applications that are running in the cloud can be monitored using tools like Application Insights, and alerts can be configured so that the right people are notified if things start going wrong. Tools such as Autoscale make it easy to automatically scale your cloud deployments based on specific metrics, ensuring high availability and controlling costs.

Tools are also available to assist you in configuring your cloud resources for security and availability, and to help you keep costs controlled, you can configure spending limits, budgets, and more.

Given the large investment that cloud providers make in manageability, moving to the cloud can make managing complex systems easier and more effective than it is on-premises.

MORE INFO **MONITORING APPLICATIONS**

For more information on Application Insights and monitoring cloud deployments in Azure, see Skill 3.4, "Describe monitoring tools in Azure."

Skill 1.3: Describe cloud service types

As you've learned, one of the benefits of moving to the cloud is that you offload some of the responsibility of your infrastructure to the cloud provider. Moving to the cloud, however, is not an all-or-nothing kind of thing. When you're evaluating your use of the cloud, you need to balance your need for controlling resources against the convenience of allowing the cloud provider to handle things for you.

In this skill section, we will discuss the three primary categories of cloud services: *Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS)*, and *Software-as-a-Service (SaaS)*. We'll cover how your choice of service type aligns with the shared responsibility model we discussed earlier in the chapter, and we'll talk about the use cases for each service type.

This section covers:

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)
- Use cases for each cloud service type

Infrastructure-as-a-Service (IaaS)

Infrastructure refers to the hardware that your application uses, and IaaS refers to the virtualized infrastructure offered by a cloud provider. When you create an IaaS resource, the cloud provider allocates a VM for your use. In some cases, the cloud provider might do the basic operating system install for you. In other situations, you might need to install the operating system yourself. In either case, you are responsible for installing other necessary services and your application.

Because you control the installation of the operating system and other services, IaaS gives you plenty of control over your cloud resources. However, it also means that you are responsible for making sure your operating system is patched with security updates, and if something goes wrong in the operating system, you're responsible for troubleshooting it. The cloud provider is only responsible for providing the VM. You do, however, benefit from the underlying infrastructure in the area of fault tolerance and disaster recovery that we discussed earlier.

MORE INFO **REMOTE ACCESS TO IaaS VMs**

You will have remote access to your IaaS VMs so that you can interact with them just as if you were using them in your on-premises environment. When you move to PaaS and SaaS services, you typically lose that capability because the infrastructure is managed by the cloud provider.

In Figure 1-2, you see an IaaS VM in the Azure portal. The Ubuntu Server, a Linux operating system, has been chosen for the VM. Once the VM is up and running, it will be using Ubuntu Server 20.04. Unless an update is installed, it will always be running that version, and Microsoft will never install patches or version updates.

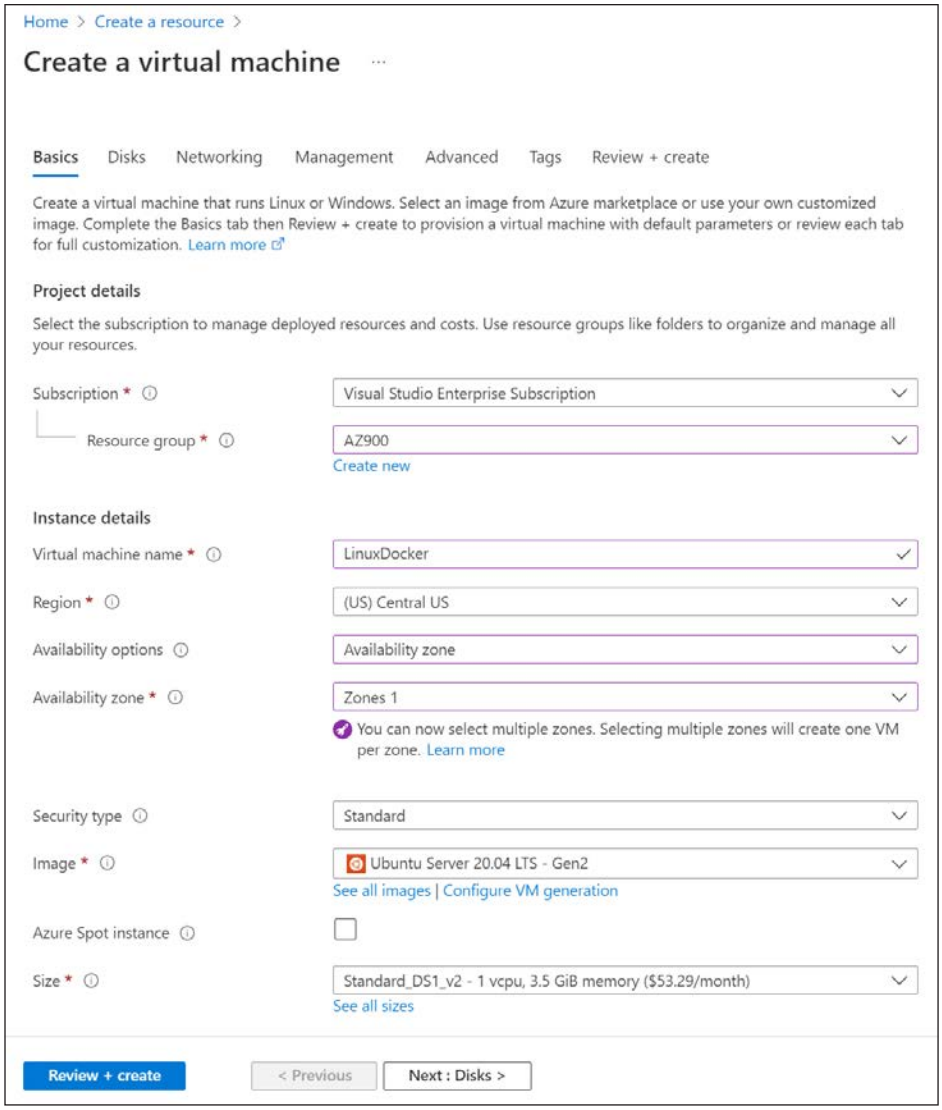


FIGURE 1-2 Creating an IaaS VM in Azure

Once you have an IaaS VM running in the cloud, you gain access to many services the cloud provider offers. For example, Microsoft offers Azure Security Center to ensure the security of your IaaS VMs, Azure Backup to make backing up data easy, Azure Log Analytics to help with troubleshooting any problems you might have, and much more.

MORE INFO **MORE INFORMATION ON IaaS AND AZURE**

For more information on IaaS and Azure, see the documentation at *https://bit.ly/az900-whatisiaas*.

IaaS services allow you to control costs effectively because you only pay for them when they are allocated to you. If you stop your IaaS VM, it is deallocated and billing stops for the resource. This makes IaaS an ideal choice if you need developers to have a platform for testing an application during release. Developers can start an IaaS VM, test the application as a team, and then stop the IaaS VM when testing is complete.

Another popular use of IaaS is when you need one or more powerful VMs for a temporary period. For example, you might need to analyze a large amount of data for a project. By utilizing IaaS VMs for your project, you can keep costs to a minimum, create resources quickly as you need them, and gain all the processing power you need.

IaaS services benefit from scaling and elasticity that we discussed earlier. If you need more VMs, you can scale out to accommodate that and then scale in when those resources are no longer needed. If you need more CPU power, more memory, or more disk space, you can quickly scale up to gain those benefits and then scale down when they're no longer needed.

In a nutshell, IaaS services are a great choice if you want to let someone else manage the hardware infrastructure (which can include both the computers and the network) related to your application, but you want to maintain control of what's installed in the operating system. In an IaaS environment, the cloud provider will not install something on the operating system for you, so the current state of what's installed on your VMs is always known to you. If this is important for your needs, IaaS might be the right choice for you. Also, IaaS is a great choice if you occasionally need high-end VMs for specific needs.

IaaS is also a great choice if you want your application and configuration in the cloud, but you want the option of not paying for it when you aren't using it. By stopping your VM, you can avoid the costs associated with it, and when you need to use your application again, you can simply start your VM and pick up right where you left off.

Platform-as-a-Service (PaaS)

In a PaaS environment, a cloud provider still provides the infrastructure for you, but they also provide the operating system, software installed in the operating system to help you connect to databases and network systems (often referred to as *middleware*), and many features that enable you to build and manage complex cloud applications.

PaaS services offer you the flexibility of controlling the application, but they offload management and control of the underlying systems to the cloud provider. If you are deploying your own application to the cloud and you want to minimize your management investment, a PaaS service is often the best choice.

> ***NOTE*** **PaaS AND VMs**
>
> **A PaaS service also uses VMs provided by the cloud provider. However, a user typically has limited visibility into those VMs. In most cases, they're entirely managed by the cloud provider.**

Suppose you need to run a web application that uses the PHP framework to connect to a back-end database system. If you were to choose IaaS for your application, you'd need to ensure that you install and configure PHP on your VM. You'd then need to install and configure the software necessary to connect to your back-end database. In a PaaS scenario, you simply deploy your web application to the cloud provider, and everything else is taken care of for you.

In Figure 1-3, we have a web application in Azure App Service, one of the PaaS offerings in Azure. It has been created on a VM that's maintained by Microsoft. Notice the option to choose either Linux or Windows, but the operating system is still managed by Microsoft.

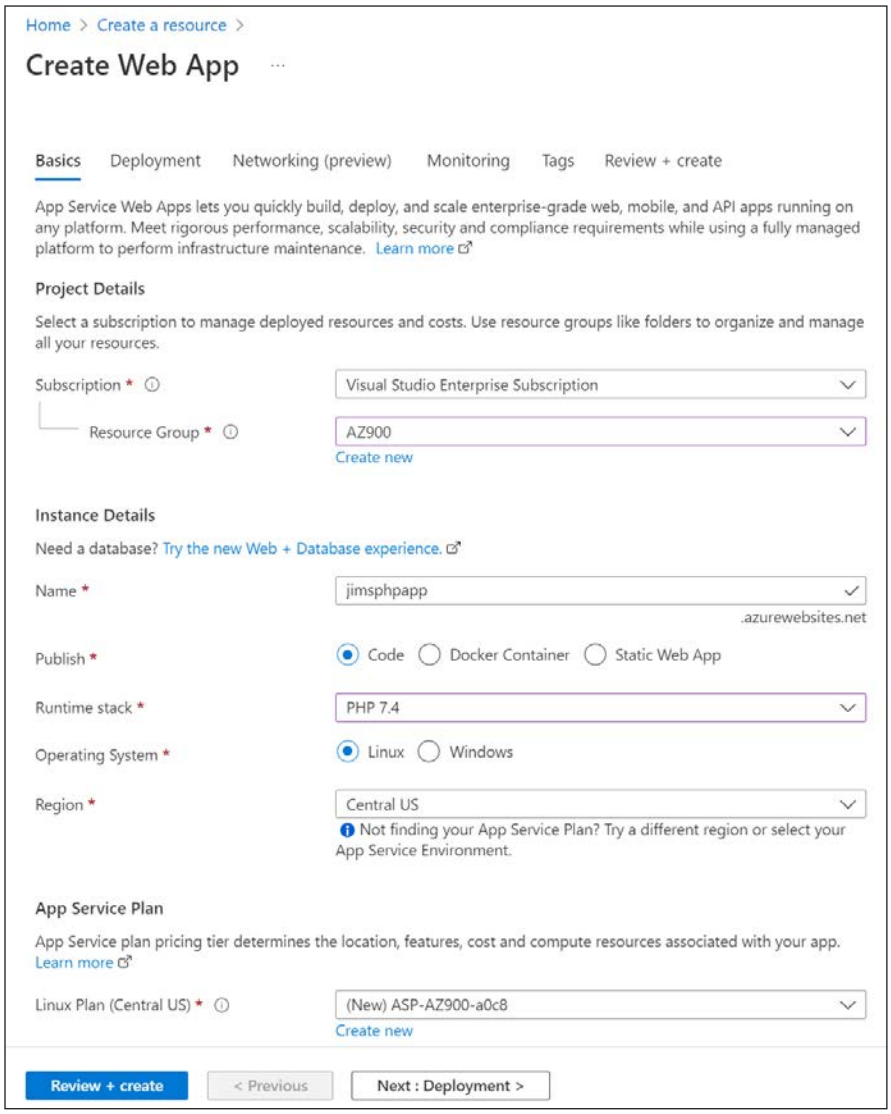


FIGURE 1-3 Creating a web app in Azure App Service

One more interesting thing in Figure 1-3 is the option to publish either your code or a Docker container. Docker is a technology that makes it easy to package your application and the components that it requires into an image that you can then deploy and run on another computer in another environment, as long as that computer has Docker installed on it. In Azure App Service, you don't have to worry about Docker installation or configuration. It's automatically included on all App Service VMs as part of Microsoft's PaaS offering, and it's completely managed and maintained by Microsoft.

Some of the other PaaS services are:

- Azure CDN
- Azure Cosmos DB
- Azure SQL Database
- Azure Database for MySQL
- Azure Storage
- Azure Synapse Analytics

In a PaaS web app offering, cloud providers offer numerous application frameworks such as PHP, Node.js, ASP.NET, .NET Core, Java, Python, and more. The cloud provider usually provides multiple versions of each framework, so you can choose a version that you know is compatible with your application. The cloud provider will also ensure that common components necessary for data connectivity from your application to other systems are installed and configured. That usually means that your application code works without you having to do any kind of complex configuration. In fact, this is one of the main benefits of using a PaaS service; you can often move your application from on-premises to a cloud environment by simply deploying it to the cloud. This concept is often referred to as *lift-and-shift*.

Because the cloud provider controls the operating system and what's installed on the VM, they can provide additional capabilities to you by adding their own features. For example, suppose you want to add a log-in feature to your web application, and you want to allow users to log in with a Microsoft, Facebook, or Google account. If you want to add this capability on-premises or in an IaaS environment, you'll need some developers to build it for you—a task that isn't easy and one that requires specialized knowledge. You must either have developers in your company who already have those skills, or you'll have to hire them. However, cloud providers often offer features like this in their PaaS services, and enabling them is as easy as flipping a switch and doing some minor configuration specific to your app.

A PaaS service also benefits from all the other enhancements offered by the cloud: you get fault tolerance, elasticity, easy and quick scaling, backup and disaster recovery features, and more. In fact, features such as backing up and restoring data are often more user-friendly and feature-rich in a PaaS environment because the cloud provider installs customized software on the PaaS VMs to add functionality.

As you can see, there are real benefits to allowing the cloud provider to control what's installed on the VMs running your application, but there can also be drawbacks. For example, the cloud provider controls when patches and updates are applied to both the operating system and to other components installed on the VMs. You'll usually be given advance notice of major changes so that you can test your application on-premises first and avoid any downtime, but you do lose the flexibility and control of deciding when to update the VM.

MORE INFO **MORE INFORMATION ON PaaS AND AZURE**

For more information on PaaS offerings in Azure, see *https://bit.ly/az900-whatispaas*.

Software-as-a-Service (SaaS)

As you've learned, IaaS requires you to control both the operating system and middleware components along with your application. When you move to PaaS, you offload the control of the operating system and middleware components to the cloud provider, and you're responsible only for your application code. As you move into the SaaS realm, the cloud provider controls everything. In other words, a SaaS service is software provided by a cloud provider that's installed on infrastructure completely controlled by the hosting provider.

SaaS services offer you the flexibility of a pay-as-you-go model. Essentially, you rent your software from a service provider. Users of the software usually access the software from a web browser, but they might also install applications that will only work if you are paying for the SaaS service. One huge benefit of web-based software is that it works from just about any device, including smart phones. Because of that, SaaS services enable connectivity and productivity for field staff using devices they already own.

When using a SaaS service, not only do you benefit from using software written and maintained by someone else, but you can also benefit from allowing the cloud provider to maintain and configure the application. For example, if your company offers corporate email, you can choose to use the Microsoft 365 SaaS service. By using the Exchange Online service in Microsoft 365, you can take advantage of enterprise-ready email solutions without having to hire IT staff and build infrastructure to support it. Instead, Microsoft maintains the system for you. Not only do you benefit from the flexibility and reliability of the cloud, but you can also rest easy knowing that Microsoft is ensuring your Exchange services are always available to your users.

SaaS services aren't just for the enterprise. In fact, most people use SaaS services all the time without even realizing it. If you use Outlook web mail, Gmail, or another online email service, you're using a SaaS service. The cloud provider hosts the email software in the cloud, and you log in and use that software using your web browser. You don't have to know anything about the software. The cloud provider can offer new features with software updates, and those new features are available to you automatically without any action on your part. If the cloud provider finds a problem with the software, they can resolve it with a patch without you even realizing anything happened.

Some of the SaaS services Microsoft makes available are:

- Microsoft 365
- Xbox Live
- OneDrive
- Power Automate (previously Microsoft Flow)

MORE INFO **MORE INFORMATION ON SaaS AND AZURE**

For more information on SaaS services and Azure, see *https://bit.ly/az900-whatissaas*.

Use cases for each cloud service type

We've already discussed some of the advantages and disadvantages of each type of cloud service. Now let's look at the use cases for each of these cloud service types. As we do that, keep the shared responsibility model in mind. A good way to visualize how the shared responsibility model translates to these cloud service types is the cloud pyramid shown in Figure 1-4. The bottom of the cloud pyramid represents the greatest amount of control, but also the greatest responsibility. As you move up in the cloud pyramid, your responsibility is decreased, but so is your control.

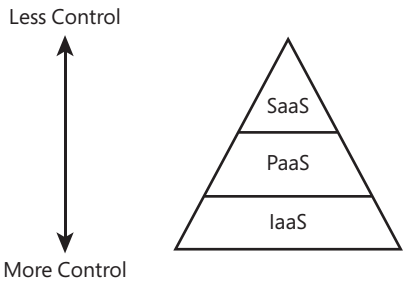


FIGURE 1-4 The cloud pyramid

As you've learned, IaaS provides you with the greatest flexibility. You can install your own software and your own components, and you control when the software and operating system are updated. An additional benefit is that you pay for your resources only when they're allocated to you, so IaaS can reduce your operational expenses. Even though you can save costs by turning off VMs you aren't using, the higher costs associated with installing and maintaining your VMs might offset that benefit.

PaaS services offer you some of the same flexibility as IaaS services without the need to manage the infrastructure. In a PaaS service, you are responsible only for the application that's installed in the cloud. This can be your own application, or an application developed by someone else (for example, a WordPress system or an e-commerce solution), but in either case, you are

responsible for the application. PaaS services are popular for developer teams who are looking to move on-premises applications to the cloud easily and quickly, and they typically offer many different deployment options to make that as easy as possible. PaaS services also offer more built-in features than IaaS services because the cloud provider installs their own software and features on the platform. Any application running in a PaaS service, however, can be affected by updates and version changes in the underlying software, and that can mean increased costs associated with testing an application before the cloud provider rolls out changes.

SaaS services are quite a bit different than IaaS or PaaS services because they are completely managed and maintained by the cloud provider. You don't have the option of installing any of your own software with a SaaS service, so the deciding factor is related entirely to whether the provided software meets your needs. The benefit of a SaaS service is that it largely removes the IT burden from your company, and it enables everyone in your company to access the software on multiple devices from just about anywhere internet access is available. You also benefit from data backup that the cloud provider includes in their infrastructure. If you need to customize the application or have any control over its configuration, however, SaaS might not be a good choice for you.

REAL WORLD **DEALING WITH THE COMPLEXITIES OF MODERN IT**

Deciding on a particular cloud service type can be straightforward in some cases, but it can also be complicated depending on your needs. For example, you might be in an industry that requires some of your information to be stored only on-premises. You might also have some older systems that aren't ready to move to the cloud, but you need your cloud applications to use those older systems. In the next skill section, you'll learn more about how to deal with such complexities.

Thought experiment

Let's apply what you've learned in this chapter. You can find the answers in the section that follows.

You work for Contoso Medical Group (CMG), and your manager is frustrated with one of your commonly used applications. The CMG IT department is resource-constrained, and they are having difficulty ensuring the application is always available.

The development team has been updating the application frequently, but due to a lack of knowledge of deployment methods, they only have the option of directly copying files, and this is causing problems with tracking changes that are being made. At the same time, the development team has no data to show whether the application is running correctly.

The problem became critical two days ago when a deadline was approaching for updating medical records. The application experienced far more usage than normal, and the system was

quickly overloaded and became unresponsive. The IT team determined the problem was the server running low on resources, but it took them two hours to build a second server to handle the load.

Your manager has come to you asking for a solution that addresses all these issues. Whatever solution you offer must consider that the medical data in this application is covered under HIPAA, and your manager wants CMG to retain all control of the data. Your manager also wants to carefully control costs.

You've decided that CMG should move the application to the cloud, but you need to sell the idea to your manager.

Answer the following questions:

1. What type of cloud service would you recommend?
2. How would you justify your choice related to the problems being encountered by the IT team?
3. How would you justify your choice related to the problems being encountered by the development team?
4. What other benefits will please your manager if your advice is followed?
5. How can you meet the requirements related to the medical records and the need to control them?

Thought experiment answers

In this section, we'll discuss the answers from the previous section.

1. A PaaS service makes the most sense in this situation. An IaaS environment would require your IT department to manage the VMs, and that would not meet your requirements. A SaaS service provides the software to you, and in this case, you need to run your company's custom application in the cloud.
2. The IT department is short on resources and is challenged in keeping the application available. In a PaaS service, the management of the VMs running the application is offloaded to the cloud provider. The cloud provider also offers an SLA so that your application is always available. The IT team will also benefit from easy scaling offered in a cloud environment, and instead of two hours, they can add more servers almost instantly.
3. In a PaaS service, the cloud provider offers flexible deployment options that make it easy to deploy an application using the method you prefer. They also provide logging so that the development team can track changes made to the application. Diagnostic features in a PaaS service (such as Azure's Application Insights) provide detailed data on how an application is performing and can alert you to code problems in an application.

4. Your manager wants to lower costs, and moving to the cloud should meet that need. Your IT department has already built a second server so that when additional need is required, you can meet it. However, the increased usage was temporary. Even so, it was related to a deadline for filing records, and the next time that deadline occurs, you'll need that second server. By moving to the cloud, you benefit from easy scaling and elasticity so that you can scale out when you need the second server to handle load, and then you can easily scale back in to reduce your costs.
5. By adopting a hybrid cloud model, you can keep your sensitive medical data on-premises, while benefiting from the application itself running in the cloud.

Chapter summary

In this chapter, you learned some of the general concepts related to the cloud. You learned about the advantages of moving to the cloud, you learned about the different cloud service types, and you learned about the different cloud models available to you. Here are the key concepts from this chapter:

- Increased control over your resources means a larger responsibility on your part. Decreased control results in more responsibility on the cloud provider's part. This concept is called the *shared responsibility model*.
- The public cloud model is sometimes referred to as a *multitenant environment*. Multiple companies and users share the same infrastructure. VMs and other infrastructure are allocated to users as they need them, and when they no longer need them, they are returned to the pool to be used by other users. The network is available publicly over the internet, but you do have the ability to put security methods in place to control access to your resources.
- The private cloud model is sometimes referred to as a *single-tenant environment*. All infrastructure is private to an individual or a company, and the network is only available within the private cloud itself. It is not exposed to the internet. In many cases, the infrastructure used in a private cloud is owned by the company, but not always. It's possible to host a private cloud in a third-party data center.
- A hybrid cloud model is a mixture of public and private cloud models. Hybrid clouds are often used when a company needs to use on-premises resources in a cloud application.
- Moving to the cloud can help avoid downtime caused by network, system, and power outages. It can also help you if you need to diagnose problems with an application or problems with an external system that your application uses.
- You can scale up (or vertically) when you want to add additional CPUs or more memory using a more powerful VM.
- You can scale out (or horizontally) if you want to add more VMs to handle the additional load.

- Cloud providers give you ways to automatically scale based on usage patterns, resource utilization, and times of day. This is referred to as *elasticity*.
- Cloud providers monitor the health of the infrastructure. When a VM becomes unhealthy, the cloud provider can automatically move you to a healthy VM without you having to do anything. This is called *fault tolerance*.
- Cloud providers also operate across multiple data centers in different regions of the world. If a natural disaster (or any other disaster) happens in one region, you can switch to another, assuming you have replicated your environment in multiple regions. This kind of planning is called "Business Continuity and Disaster Recovery planning," and cloud providers often have features in place to make implementing a plan easy. This is often referred to as *disaster recovery*.
- Cloud providers monitor the cloud infrastructure to detect threats and keep your resources safe and secure.
- Cloud providers offer governance features to allow you control over who can access your resources and what they can do with them.
- Cloud providers offer tools that make it easy to monitor and manage your cloud resources.
- Infrastructure-as-a-Service (IaaS) offers infrastructure running in the cloud, but you must maintain the operating system and what's installed on that infrastructure. IaaS services offer you the most control in the cloud, but they also carry the largest management burden.
- Platform-as-a-Service (PaaS) offloads the management of the infrastructure, and it also offloads the operating system and components installed on the VMs to the cloud provider. You are responsible for your application. PaaS services also offer many additional features that make it easy to add functionality to an application without having to write complex code. Development teams also have a wide variety of deployment methods available, and the cloud provider often automates much of that process.
- Software-as-a-Service (SaaS) provides a hosted application in the cloud that is most commonly accessed using a web browser. In a SaaS service, the cloud provider manages everything for you. You are essentially renting the use of the software from the cloud provider. A big benefit of SaaS is that it makes applications easily accessible by employees in the field on any device.

CHAPTER 2

Describe Azure architecture and services

In Chapter 1, "Describe cloud computing," you learned about the cloud and how you can benefit from using cloud services. Microsoft Azure was mentioned, but not in much detail.

In this chapter, we dive into the many services and solutions that Azure offers. You'll gain an understanding of the key concepts in Azure's architecture, which apply to all Azure services. We cover Azure datacenters and ways that Microsoft implements fault tolerance and disaster recovery by spreading Azure infrastructure across the globe. You'll also learn about availability zones, which are Microsoft's solution for ensuring your services aren't affected when a particular Azure datacenter experiences a problem.

You'll also discover how to manage and track your Azure resources and how you can work with resources as a group using Azure resource groups. You'll learn how to use resource groups to plan and manage Azure resources and how resource groups can help you categorize your operational expenses in Azure.

Once you have the foundational understanding of Azure, you'll dig into some of the compute and networking services in Azure. You'll learn about Azure Virtual Machines and Virtual Machine Scale Sets. You'll learn about some of the application hosting options such as Azure App Service, and you'll learn about networking services in Azure, such as Azure Virtual Networks, Azure DNS, and Azure VPN Gateway.

We'll then look at some of the storage services in Azure. You'll learn about storage tiers and redundancy options, and we'll look at how you can move files into Azure Storage easily.

We'll close out the chapter with a discussion of identity, access, and security. You'll learn about Azure Active Directory and authentication methods in Azure. You'll also learn about how you can control access to your Azure resources and your options for keeping them secure.

If you think that's a lot to cover, you're right! It's important for you to understand all these topics to pass the AZ-900 exam. With the foundational knowledge of the cloud from Chapter 1, "Describe cloud computing," you'll find that understanding Azure-specific concepts will be easier than you think.

Skills covered in this chapter:

- Describe the core architectural components of Azure
- Describe Azure compute and networking services
- Describe Azure Storage services
- Describe Azure identity, access, and security

Skill 2.1: Describe the core architectural components of Azure

If you were to ask any CEO to list the five most important assets of their company, it is likely that the company's data would be near the top of the list. The world we live in revolves around data. Just look at companies like Meta (the company that owns Facebook) and Google. These companies offer services to us that we like. Everyone likes looking at pictures from friends and family on Facebook (mixed in with things we don't like so much), and many people use Google to look for things on the internet. Meta and Google don't offer those services because they want to be nice to us. They offer those services because it's a way for them to collect a large amount of data on their customers, and that data is their most valuable asset.

Meta and Google aren't alone. Most companies have vast amounts of data that is key to their business and keeping that data safe is at the cornerstone of business decisions. That's why some companies are hesitant to move to the cloud. They're afraid of losing control of their data. Not only are they afraid that someone else might gain access to sensitive data, but they're also concerned about losing data that would be difficult (or even impossible) to re-create.

Microsoft is keenly aware of those fears, and Azure has been designed from the ground up to instill confidence in this area. Let's look at some core architectural components that help Microsoft deliver on the cloud promise.

This section covers:

- Azure regions, regional pairs, and sovereign regions
- Availability zones
- Azure datacenters
- Azure resources and resource groups
- Azure subscriptions
- Management groups
- Hierarchy of resource groups, subscriptions, and management groups

Azure regions, regional pairs, and sovereign regions

The term "cloud" tends to make people think of Azure as a nebulous entity that you can't clearly see, but that would be a mistake. While there certainly are logical constructs to Azure, there are also physical components to it. After all, at the end of the day, we're talking about computers!

In order to provide Azure services to people around the world, Microsoft has created boundaries called *geographies*. A geography boundary is oftentimes the border of a country, and there's good reason for that. There are often regulations for data handling that apply to an entire country, and having a geography defined for a country allows Microsoft to ensure that data-handling regulations are in place. Many companies (especially ones that deal with sensitive data) are also much more comfortable if their data is contained within the confines of the country in which they operate.

There are numerous geographies in Azure. For example, there's a United States geography, a Canada geography, a UK geography, and so on. Each geography is broken out into two or more regions, each of which is typically hundreds of miles apart. As an example, within the United States geography, there are many regions, including the Central US region in Iowa, the East US region in Virginia, the West US region in California, and the South Central US region in Texas. Microsoft also operates isolated regions that are completely dedicated to government data because of the additional regulations that governmental data requires.

Within each geography, Microsoft has created another logical boundary called a *regional pair*. Each regional pair contains two regions within the geography. When Microsoft must perform updates to the Azure platform, they perform those updates on one region in the regional pair. Once those updates are complete, they move to the next region in the regional pair. This ensures that the availability of your services operating within a regional pair aren't impacted by updates.

MORE INFO **REGIONAL PAIRS**

To benefit from regional pairs, you should make sure to deploy resources redundantly to each region within the pair. You can find a list of all regional pairs by browsing to *https://bit.ly/az900-regionpairs*.

EXAM TIP

The fact that each geography contains at least two regions separated by a large physical distance is important. That's how Azure maintains disaster recovery, and it's likely this concept will be included on the exam.

Microsoft provides tools to control the use of Azure resources to meet corporate policies, but some compliance requirements can't be met by simply applying policies. For example, some US government compliance scenarios require that data stays within the United States of

America and that only citizens of the United States have any access to systems used to store that data. You can't meet this requirement with policies. In fact, you can't meet that requirement at all in the public cloud. To address this type of issue, Microsoft has several sovereign clouds that are separated from their public cloud offerings.

To address concerns related to the government of the United States, Microsoft developed completely isolated Azure datacenters that make up the Azure Government cloud. Azure Government datacenters are separate from public datacenters. All employees working in Azure Government are screened and are citizens of the US. Even Microsoft employees who provide technical support to Azure Government customers are required to be US citizens.

Because Microsoft also wanted to allow for compliant communication between the Azure Government cloud and on-premises government systems, they also developed dedicated networking infrastructure that is completely isolated from other Azure networks and that uses its own dedicated fiber-optic components.

Azure Government isn't only for federal government agencies. Cities and municipalities also take advantage of Azure Government for compliance. When a customer signs up for Azure Government, Microsoft vets that user to ensure they are representative of a government agency. Only then are they given a subscription to Azure Government.

The Azure Government cloud has all the same features and services as the public cloud, but there are small differences. For example, the portal for Azure Government is located at *https://portal.azure.us* instead of *https://portal.azure.com*. URLs for Azure services also use the `.us` top-level domain, so if you create an App Service web app in Azure Government, your default domain name is *https://webapp.azurewebsites.us*. However, outside of that difference, everything else is the same, so developers who have a skill set in cloud development in Azure will find that their skills transfer directly to Azure Government.

The United States Department of Defense has additional compliance requirements called DoD Impact Level 5 Provisional Authorization. Compliance with this relates to controlled unclassified information that requires additional levels of protection. These additional DoD requirements are met by a subset of datacenters within Azure Government that are approved for DoD usage.

Microsoft also understands that the strict requirements in the EU need a unique approach, so they developed another sovereign cloud called Azure Germany. Much like Azure Government, Azure Germany is a distinct cloud system that's designed to meet specific compliance needs. Azure Germany is available to customers doing business in the EU, the European Free Trade Association, and the UK.

Azure Germany datacenters are physically located in Germany and are operated under strict security measures by a local company named T-Systems International (a subsidiary of Deutsche Telekom) that operates as a data trustee. The data trustee has full control over all data stored in Azure Germany and all the infrastructure used to house that data. Microsoft is involved in managing only those systems that have no access at all to customer data.

Another region where Azure has specific requirements is China. Microsoft operates another separate cloud in China called Microsoft Azure China. Azure China is operated by Shanghai Blue Cloud Technology Co., Ltd. (frequently referred to as simply BlueCloud). BlueCloud is owned by Beijing 21Vianet Broadband Data Center Co., Ltd. (often called 21Vianet), an internet and datacenter service provider in China. Because of this relationship, you may see Azure China referred to as "Microsoft Azure operated by 21Vianet" or simply "Azure 21Vianet."

Azure China doesn't offer the full set of features offered in other Azure clouds, but Microsoft is working hard to add additional features and services. For all the details on what is and isn't offered in Azure China, browse to *https://bit.ly/az900-azurechina*.

Availability zones

The fact that regions are physically separated by hundreds of miles protects Azure users from data loss and application outages caused by disasters in a particular region. However, it's also important that data and applications maintain availability when a problem occurs at a particular building within a region. For that reason, Microsoft developed availability zones.

> ***NOTE*** **AVAILABILITY ZONE AVAILABILITY**
>
> **Availability zones aren't available in all Azure regions, nor are they available for all Azure services in regions that support them. For the most up-to-date list of availability zone-enabled regions and services, see *https://bit.ly/az900-azones*.**

There are at least three availability zones within each enabled region, and each availability zone has a water supply, cooling system, network, and power supply that is isolated from other zones. By deploying an Azure service in two or more availability zones, you can achieve high availability in a situation where there is a problem in one zone.

EXAM TIP

Availability zones provide high availability and fault tolerance, but they might not help you with disaster recovery. If there is a localized disaster, such as a fire in a datacenter that houses one zone, you will benefit from availability zones. Because availability zones are in the same Azure region, if there is a large-scale natural disaster such as a tornado, you might not be protected. In other words, availability zones are just one facet to an overall disaster recovery and fault-tolerant design.

Because availability zones are designed to offer enhanced availability for infrastructure, not all services support availability zones. For example, Azure has a service called App Service Certificates that allows you to purchase and manage an SSL certificate through Azure. It wouldn't make any sense to host a certificate in App Service Certificates within an availability zone because it's not an infrastructure component.

By deploying your service to two or more availability zones, you ensure the maximum availability for that resource. In fact, Microsoft guarantees 99.99 percent uptime for Azure virtual machines only if two or more VMs are deployed into two or more zones. Figure 2-1 illustrates the benefit of running in multiple zones. As you can see, even though availability zone 3 has gone offline for some reason, zones 1 and 2 are still operational.

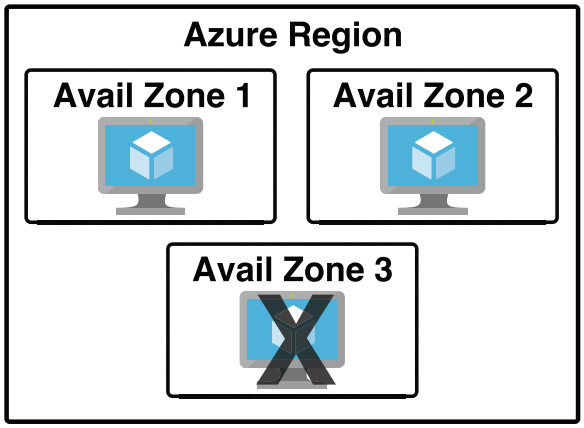


FIGURE 2-1 Azure virtual machine inside of three availability zones

> ***NOTE*** **THE STATUS OF AZURE**
>
> **Microsoft operates a website that shows the status of all Azure services. If you notice a problem with your resources, you can check the Azure Status page at *https://status.azure.com*.**

EXAM TIP

Don't confuse availability zones with availability sets. Availability sets allow you to create two or more virtual machines in different physical server racks in an Azure datacenter. Microsoft guarantees a 99.95 percent SLA with an availability set.

An availability zone allows you to deploy to two or more distinct datacenters (physical buildings) within a region. Microsoft guarantees a 99.99 percent SLA with availability zones.

There are two categories of services that support availability zones: *zonal* services and *zone-redundant* services. Zonal services are services such as virtual machines, managed disks used in a virtual machine, and public IP addresses used in virtual machines. To achieve high availability, you must explicitly deploy zonal services into two or more zones.

> ***NOTE*** **MANAGED DISKS AND PUBLIC IP ADDRESSES**
>
> **When you create a virtual machine in Azure and you deploy it to an availability zone, Azure will automatically deploy the managed disk(s) and public IP address (if one is configured) to the same availability zone.**

Zone-redundant services are services such as zone redundant storage and SQL databases. To use availability zones with these services, you specify the option to make them zone redundant when you create them. (For storage, the feature is called ZRS, or zone-redundant storage. For SQL database, there is an option to make the database zone redundant.) Azure takes care of the rest for you by replicating data automatically to multiple availability zones.

Azure datacenters

At each region, Microsoft has built datacenters (physical buildings) that contain the physical hardware that Azure uses. These datacenters contain climate-controlled buildings that house the server racks containing physical computer hardware. Each region also operates on its own network infrastructure, and Microsoft has designed the networks for low latency. Therefore, any Azure services you have in a particular region will have reliable and fast network connectivity with each other.

MORE INFO **CUSTOMERS ONLY SEE REGIONS**

When a customer is creating Azure resources, only the region is visible. The concept of geographies is an internal implementation of Azure that customers don't really have visibility of when using Azure. Customers also don't have visibility into the concept of regional pairs, but they can see each region within a regional pair.

Each datacenter has an isolated power supply and power generators in case of a power outage. All the network traffic entering and exiting the datacenter goes over Microsoft's own fiber-optic network on fiber owned or leased by Microsoft. Even data that flows between regions across oceans travels over Microsoft's fiber-optic cables that traverse the oceans.

MORE INFO **DATACENTER POWER**

A 2018 study found that Microsoft Azure was up to 93 percent more efficient than using on-premises services, and by 2025, Microsoft is committed to 100 percent renewable energy in Azure datacenters.

To remove reliance on third-party power providers, Microsoft is also investing in the development of natural gas-powered, fully integrated fuel cells for power. Not only do fuel cells provide clean power, but they also remove the power fluctuations and other disadvantages of relying on the power grid. In late July 2020, Microsoft announced that it had developed a hydrogen-fueled cell that could run an Azure datacenter for 48 consecutive hours.

To ensure that data in Azure is safe from disasters and failures caused by possible problems in a particular region, customers are encouraged to replicate data in multiple regions. For example, if the South Central US region is hit by a devastating tornado (not out of the question in Texas), data that is also replicated to the North-Central US region in Illinois is still safe and available. To ensure that applications are still performing as quickly as possible, Microsoft guarantees round-trip network performance of 2 milliseconds or less between regions.

Azure resources and resource groups

You might think that moving to the cloud isn't as simple as it first seemed. Creating a single resource in Azure is pretty simple, but most Azure services are made up of many resources. For example, when you create a virtual machine, you're creating a virtual network, a network adapter, an IP address, a disk, and many other resources. Virtual machines aren't unique in this respect. A single Azure deployment is typically made up of many Azure resources, some of which you explicitly create and others that are created implicitly by Azure.

Now add the complexity when you're dealing with enterprise-level applications that consist of a complex array of Azure services or applications that involve multiple layers of complexity spread across multiple Azure regions. Things can certainly get chaotic quickly.

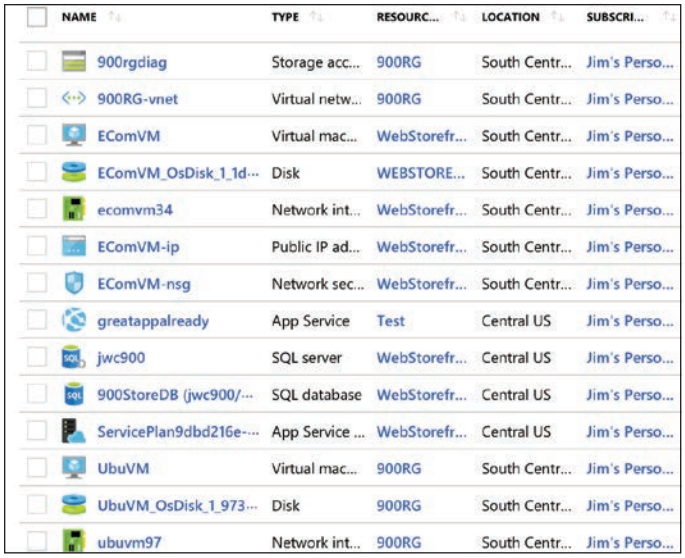
Fortunately, Azure provides a feature that helps you deal with this kind of problem: the resource group. A *resource group* is a logical container for Azure resources. By creating all Azure resources associated with a particular application in a single resource group, you can then deploy and manage all those resources as a single entity.

Organizing Azure resources in a resource group has many advantages. You can easily set up deployments using a feature known as an Azure Resource Manager (ARM) template. *ARM template* deployments are typically for a single resource group. You can deploy to multiple resource groups but doing so requires you to set up a complicated chain of ARM templates.

> ***MORE INFO*** **MORE ON ARM TEMPLATES**
>
> **You'll learn more about ARM templates in Skill 3.3, "Describe features and tools for managing and deploying Azure resources," in Chapter 3.**

Another advantage to resource groups is that you can name a resource group with an easily recognizable name so that you can see all Azure resources used in a particular application at a glance. This might not seem so important until you start deploying Azure resources and realize that you have many more resources than you first thought. If you're looking at all your Azure resources, it can be hard to differentiate which resources go with which app. Resource groups solve that problem.

In Figure 2-2, you can see a lot of Azure services. Some of these were automatically created by Azure to support other services, and in many cases, Azure gives the resource an unrecognizable name.



NAME	TYPE	RESOURC...	LOCATION	SUBSCRI...
900rgdiag	Storage acc...	900RG	South Centr...	Jim's Perso...
900RG-vnet	Virtual netw...	900RG	South Centr...	Jim's Perso...
EComVM	Virtual mac...	WebStorefr...	South Centr...	Jim's Perso...
EComVM_OsDisk_1_1d···	Disk	WEBSTORE...	South Centr...	Jim's Perso...
ecomvm34	Network int...	WebStorefr...	South Centr...	Jim's Perso...
EComVM-ip	Public IP ad...	WebStorefr...	South Centr...	Jim's Perso...
EComVM-nsg	Network sec...	WebStorefr...	South Centr...	Jim's Perso...
greatappalready	App Service	Test	Central US	Jim's Perso...
jwc900	SQL server	WebStorefr...	Central US	Jim's Perso...
900StoreDB (jwc900/···	SQL database	WebStorefr...	Central US	Jim's Perso...
ServicePlan9dbd216e-···	App Service ...	WebStorefr...	Central US	Jim's Perso...
UbuVM	Virtual mac...	900RG	South Centr...	Jim's Perso...
UbuVM_OsDisk_1_973···	Disk	900RG	South Centr...	Jim's Perso...
ubuvm97	Network int...	900RG	South Centr...	Jim's Perso...

FIGURE 2-2 All my Azure resources

In Figure 2-3, you can see resources that are in the `WebStorefront` resource group. These are the Azure resources used in the e-commerce storefront.

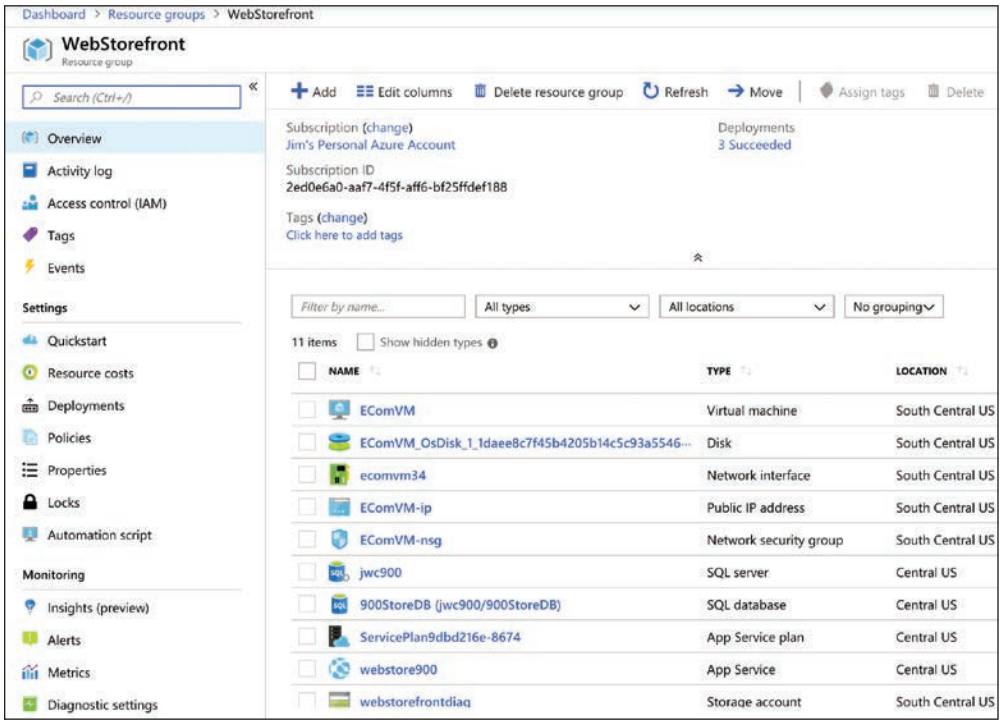


FIGURE 2-3 An Azure resource group

It's convenient to see all the resources associated with a particular app, but you aren't locked into that paradigm. This is a useful example, because it's a common use of resource groups; however, you can organize your resource groups any way you choose. Notice in Figure 2-3 that you see resources in several different Azure regions (Regions are in the Location column). If you have access to multiple Azure subscriptions, you can also have resources from multiple subscriptions in a single resource group.

If you look at the left side of Figure 2-3, you'll see a menu of operations that you can perform on your resource group. We won't go into all of these because it's out of scope for the AZ-900 exam, but there are a few that clarify the benefit of resource groups.

If you click **Resource Costs**, you can see the cost of all the resources in this resource group. Having that information at your fingertips is especially helpful in situations where you want to make sure certain departments in your company are charged correctly for the resources they use. In fact, some companies will create resource groups for each department rather than creating resource groups scoped to applications. Having a Sales and Marketing resource group or an IT Support resource group, for instance, can help you immensely when reporting and controlling costs.

EXAM TIP

An Azure resource can only exist in one resource group. In other words, you can't have a virtual machine in a resource group called `WebStorefront` and in a resource group called `SalesMarketing`, because it must be in one group or the other. You can move Azure resources from one resource group to another.

MORE INFO **MOVING AZURE RESOURCES**

Moving Azure resources between resource groups or subscriptions isn't without risk. Microsoft has documented some things you can do to avoid problems when moving resources. You can read that guidance by browsing to *https://bit.ly/az900-movingresources*.

You can also click **Automation Script**, and Azure will generate an ARM template that you can use to deploy all these Azure resources. This is useful in a situation where you want to deploy these resources later or when you want to deploy them to another Azure subscription.

When you delete a resource group, all the resources in that resource group are automatically deleted. This makes it easy to delete multiple Azure resources in one easy step. Suppose you are testing a scenario and you need to create a couple of virtual machines, a database, a web app, and more. By placing all these resources in one resource group, you can easily delete that resource group after your testing and Azure will automatically delete all the resources in it for you. This is a great way to avoid unexpected costs associated with resources you are no longer using.

Azure subscriptions

You get an Azure subscription automatically when you sign up for Azure, and all the resources you create are created inside that subscription. You can, however, create additional subscriptions that are tied to your Azure account. Additional subscriptions are useful in cases where you want to have some logical groupings for Azure resources or if you want to be able to report on resources used by specific groups of people.

Each Azure subscription has limits (sometimes called quotas) assigned to it. For example, you can have up to 250 Azure storage accounts per region in a subscription, up to 25,000 virtual machines per region, and up to 980 resource groups per subscription across all regions.

MORE INFO **SUBSCRIPTION LIMITS**

You can find details on Azure subscription limits at *https://bit.ly/az900-sublimits*.

EXAM TIP

Microsoft support can increase limits in some scenarios if you have a good business justification. Some limits, however, cannot be increased.

Figure 2-4 shows an Azure subscription in the Azure portal.

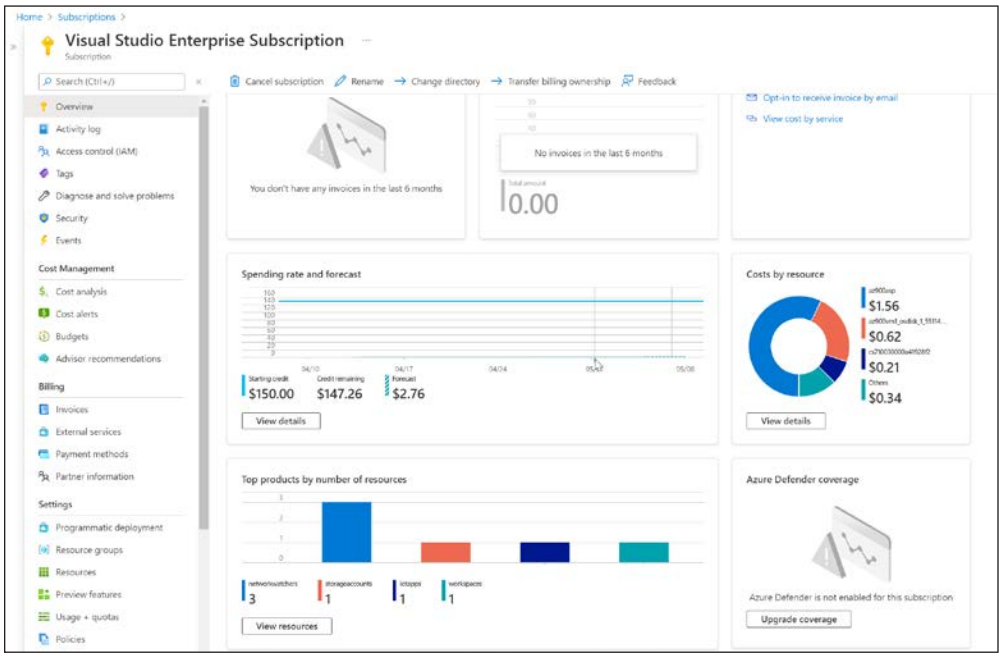


FIGURE 2-4 Azure subscription in the Azure portal

On the **Overview** blade, you can see a chart of your spending rate and forecasted costs, along with a cost breakdown for each of the resources. If you click the **View Details** button on the **Costs By Resource** tile, you can see a further breakdown of the Azure expenses, as shown in Figure 2-5. In this view, you see costs by **Service Name**, **Location (Azure region)**, and **Resource Group**, along with a graph of the costs for the month.

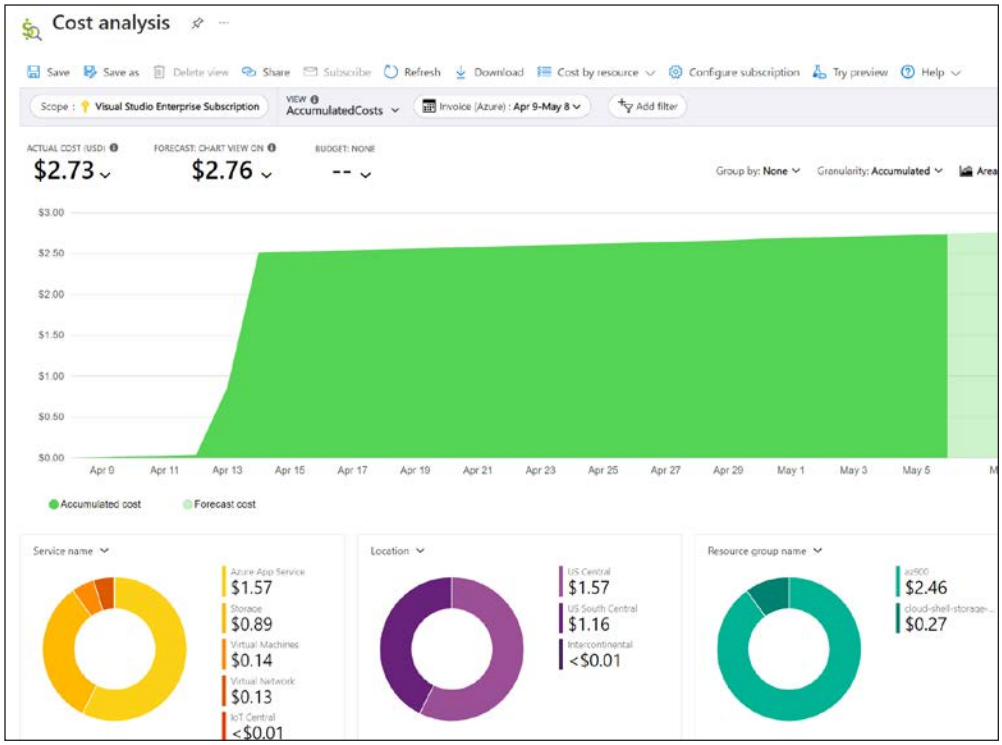


FIGURE 2-5 Azure subscription cost analysis

MORE INFO **CREATING BUDGETS**

You can manage your costs in Azure by creating budgets. You'll learn more about that in Skill 3.1, "Describe cost management in Azure," in Chapter 3.

Azure invoices are also available for the subscription from within the Azure portal. You can see your current Azure invoice as well as all past invoices by clicking Invoices in the menu for the subscription.

You can create additional Azure subscriptions in your Azure account. This is useful in cases where you want to separate costs or if you are approaching a subscription limit on a resource. To create a new Azure subscription, type **subscription** in the search box and click **Subscriptions,** as shown in Figure 2-6.

FIGURE 2-6 Azure subscriptions

To create a new subscription, click **Add** in the **Subscriptions** blade, as shown in Figure 2-7.

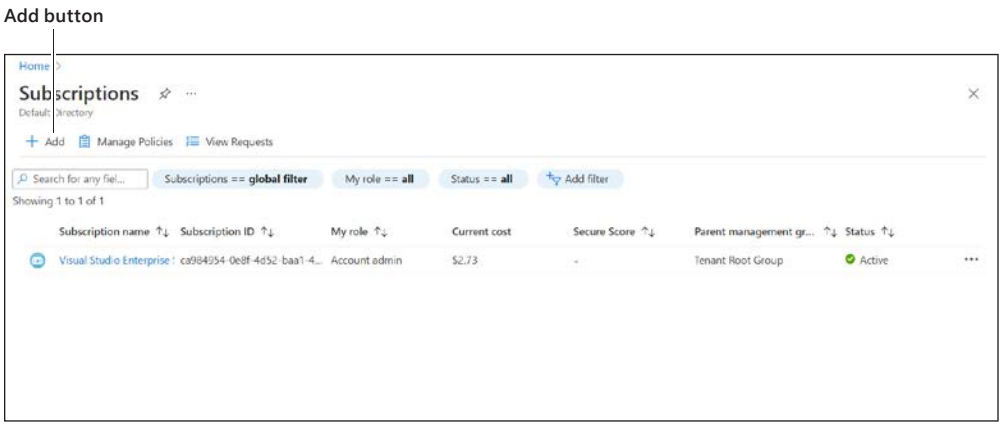


FIGURE 2-7 Creating a new subscription

After you click **Add**, you need to choose which type of subscription you want to create. There are several types of Azure subscriptions:

- **Free Trial** Provides free access to Azure resources for a limited time. Only one free trial subscription is available per account, and you cannot create a new free trial if a previous one has expired.
- **Pay-As-You-Go** You pay only for those resources you use in Azure. There's no up-front cost, and you can cancel the subscription at any time.
- **Azure For Students** A special subscription designed for exploring Azure services. It provides free Azure credits and access to many services free for 12 months.

NOTE **AZURE SUBSCRIPTION TYPES**

Depending on the type of Azure account you have, you might have additional subscription options.

EXAM TIP

Each subscription is associated with a globally unique identifier called a *subscription ID*. You can give each subscription a descriptive name to help you identify it, but Azure will always use the subscription ID to identify your subscription. When you talk to Microsoft about your Azure account, they'll also often ask for your subscription ID.

You now understand Azure subscriptions and how you can create additional subscriptions if needed. Once you've created additional subscriptions and resources in those subscriptions, you might find that managing all your resources becomes more cumbersome. To help with that, Microsoft has developed a feature called management groups.

Management groups

Management groups are a convenient way to apply policies and access control to your Azure resources. Much like a resource group, a management group is a container for organizing your resources. However, management groups can contain only Azure subscriptions or other management groups.

NOTE **AZURE IDENTITY AND GOVERNANCE**

At this point, you aren't expected to understand concepts such as access control and policies. Access control is introduced in Skill 2.4, "Describe Azure identity, access, and security," and policies are discussed in Skill 3.2, "Describe features and tools in Azure for governance."

In Figure 2-8, three management groups have been created for a company. The Sales Dept. management group contains subscriptions for the sales department. The IT Dept. management group contains a subscription and another management group, and two additional subscriptions are within that management group. The Training Dept. management group contains two subscriptions for the training department.

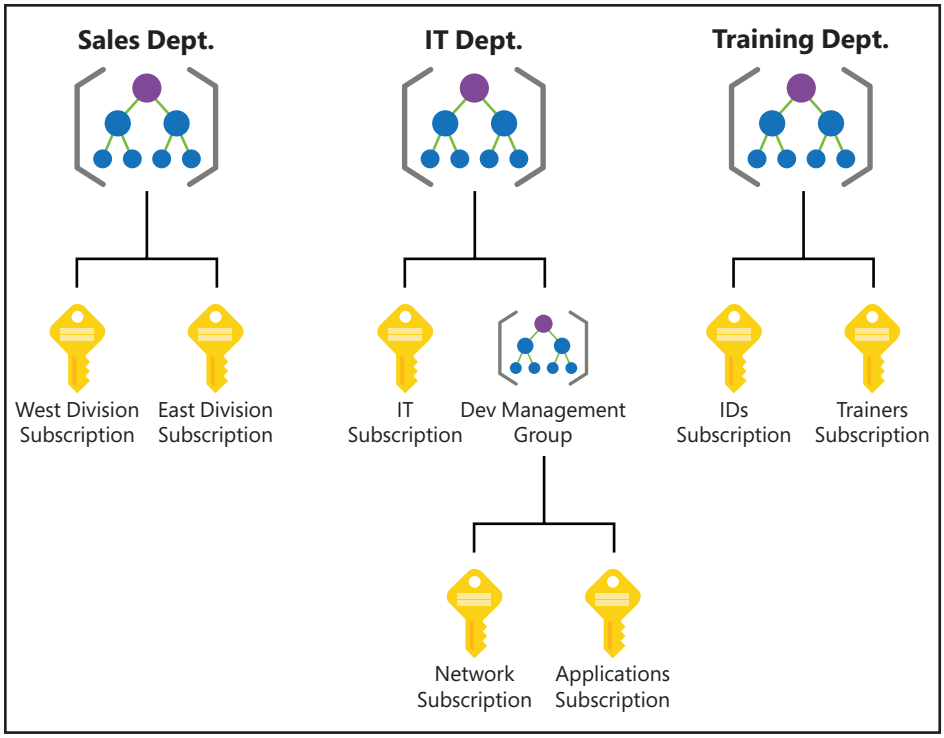


FIGURE 2-8 Management groups organizing subscriptions and other management groups

By organizing the subscriptions using management groups, you can have more precise control over who has access to which resources. You can also control the configuration of resources created within those subscriptions.

After you create a management group, you can move any of your subscriptions into that management group. You can also move a management group into another management group. There are, however, a few limitations:

- You're limited to a total of 10,000 management groups.
- A management group hierarchy can only support up to six levels.
- You cannot have multiple parents for a single management group or subscription.

Hierarchy of resource groups, subscriptions, and management groups

You should have a general sense of the hierarchy of resource groups, subscriptions, and management groups, but it's important for you to fully grasp how they are all related to each other.

At the top of the hierarchy is the management group. As you saw in the last section, you can create multiple management groups, but even if you never create one, you still have one by default called the Tenant Root Group. This default management group is part of your Azure Active Directory tenant.

> ***MORE INFO*** **AZURE ACTIVE DIRECTORY**
>
> **You'll learn about Azure Active Directory in Skill 2.4, "Describe Azure identity, access, and security."**

Your Azure subscription is inside a management group. It can be a management group that you explicitly created or the Tenant Root Group management group.

Azure resources that you create must be inside of a resource group, and that resource group gets created inside of your Azure subscription. You can't create an Azure resource without first specifying which resource group you want it to be created in. Unlike management groups, there isn't a default resource group. You must explicitly create each resource group in your subscription.

Understanding this hierarchy becomes important when we start talking about things such as access control, policies, and so on. We will refer to this hierarchy at that point, but feel free to revisit this section if you need a refresher.

Skill 2.2: Describe Azure compute and networking services

We've talked about the fact that Azure offers many different services to fit many different needs. In this section, we'll cover some of those services in detail. We'll also talk about some of the choices you have when using Azure services.

This section covers:

- Compute types
- Options for Azure virtual machines
- Resources required for virtual machines
- Application hosting options
- Virtual networking

Compute types

Any cloud service that consumes resources such as CPU and memory is categorized as a *compute* service. We've talked a little about virtual machines, one of the most common types of compute services, but there are other compute types available in Azure.

Container instances

Container instances refer to an application that runs in a Docker container runtime. You can run a containerized application on a virtual machine, but you can also use a service like Azure Container Instances (ACI) to run a containerized application on a VM that isn't directly allocated to you.

It's becoming commonplace for companies to move applications between "environments," and this type of thing is even more prevalent when it comes to the cloud. In fact, one of the most complicated aspects of moving to the cloud is dealing with the complexities of moving to a new environment. To help with this problem and to make it easier to shift applications into new environments, the concept of *containers* was invented.

A container is created using a zipped version of an application called an *image*, and it includes everything the application needs to run, including the user-mode portions of the operating system. That might include a database engine, a web server, and so on. The image can be deployed to any environment that supports the use of containers. Once there, the image is used to start a container the application runs in.

> ***NOTE*** **CONTAINERS USE THEIR OWN OPERATING SYSTEM**
>
> **The user-mode components of the operating system for a container is part of the image. It's important that your VM uses an operating system that's compatible with your container. A Docker image that was built for Linux will not run on a Windows host and vice versa.**

To run an application in a container, a computer needs to have a container runtime installed on it. The most popular container runtime is Docker, a runtime developed and maintained by Docker, Inc. Docker not only knows how to run applications in containers, but it also enforces certain conditions to ensure a secure environment.

> ***MORE INFO*** **DOCKER IMAGES**
>
> **You aren't limited to your own images. In fact, Docker runs a repository of images that you are free to use in your own applications. You can find it at *https://hub.docker.com*.**

Each container typically operates within an isolated environment. It has its own network, its own storage, and so on. Other containers running on the same machine cannot access the data and systems used by another container unless the developer of the image takes explicit steps to allow it. This makes containerized applications an ideal solution when security is a concern.

You can use containers in Azure by installing a container runtime such as Docker on a VM and configuring everything yourself. However, an easier option is to use Azure Container Instances (ACI) or Azure Kubernetes Service (AKS).

ACI makes it easy to start a container with minimal configuration. You simply tell ACI where to find the image (using either a Docker tag or a URL to the image) and some basic configuration for the VM you want the container to run on.

Azure creates server resources as needed to run your container, but you're not paying for an underlying VM. Instead, you pay for the memory and CPU that your container uses. That translates into extremely low costs in most cases. For example, if your ACI app is running on a machine with 1 CPU and 1 GB of memory and you use the app for 5 minutes a day, your cost would be less than 5 cents at the end of the month!

ACI is designed to work with simple applications. You can define a container group and run multiple containers within an ACI instance, but ACI isn't a good choice for you if you have an application that is used heavily by many people and that might need to take advantage of scaling. Instead, Azure Kubernetes Service (AKS) would be a better choice.

When you create a container instance in ACI, you specify a name for the container, the image you want to use, and the size of the VM you want to run your container. If you don't have an image handy, Microsoft provides multiple sample images you can use. In Figure 2-9, an ACI instance named `jimsaciapp` is being created in the East US region using one of the sample Quickstart images.

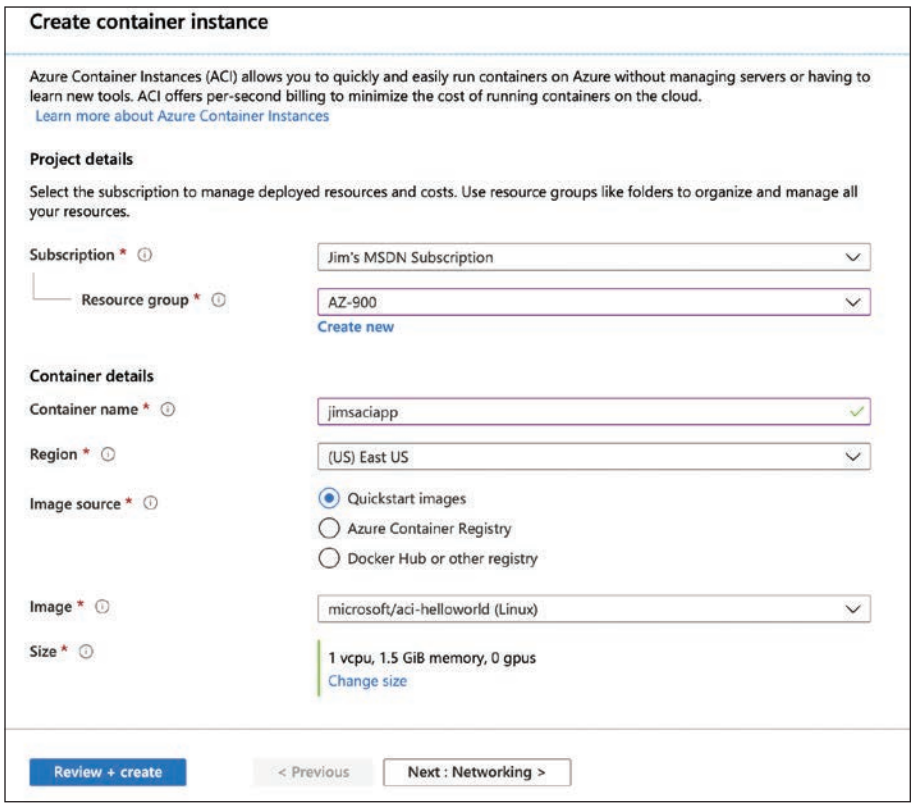


FIGURE 2-9 Creating an ACI instance with a Quickstart image

To make this instance accessible on the internet, you'll need to set the DNS name label for the instance. This setting is accessible by clicking **Next: Networking** at the bottom of the screen, as shown in Figure 2-9. In Figure 2-10, the **DNS Name Label** for this instance is set to `jimsaciapp`. After the instance is created, it can be accessed by browsing to *http://jimsaciapp.eastus.azurecontainer.io*.

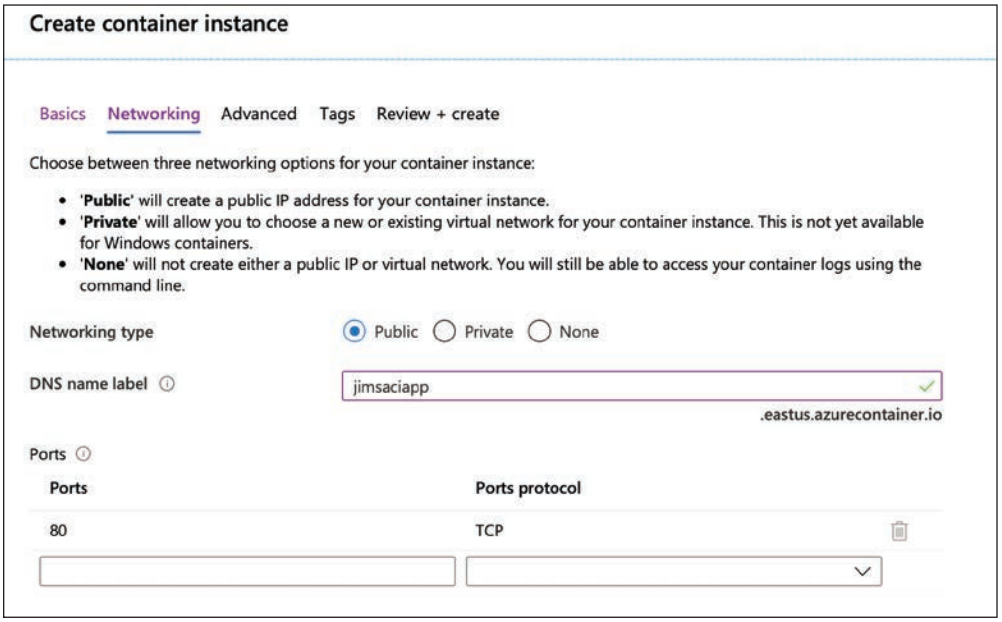


FIGURE 2-10 Setting a DNS Name Label so the instance can be reached using a URL

EXAM TIP

You can't change the DNS Name Label after the instance is created. You also can't change the image your instance uses. If you want to change these settings, you'll need to delete the instance and re-create it. However, doing so might mean that you lose your public IP address, so it's best to plan ahead before you create your instance.

ACI is a great option when you want a simple container to run something quickly and easily, but if you want a scalable containerized environment for running full workloads, Azure Kubernetes Service is a better choice.

MORE INFO **AZURE KUBERNETES SERVICE**

You'll learn more about Azure Kubernetes Service later in this chapter when we discuss application hosting options.

Virtual machines

Another compute type in Azure is the virtual machine, or VM. Typically, VMs are used when you need persistent machine availability. Unlike the container instance option, you pay for a VM if it's allocated to you, regardless of whether you're using it.

A virtual machine (VM) is a software-based computer that runs on a physical computer. The physical computer is considered the *host*, and it provides the underlying physical components such as disk space, memory, CPU power, and so on. The host computer runs software called a hypervisor that can create and manage one or more VMs, and those VMs are commonly referred to as *guests*.

The operating system on a guest doesn't have to be the same operating system that the host is running. If your host is running Windows 11, you can run a guest that uses Windows Server 2016, Linux, or many other operating systems. This flexibility makes VMs extremely popular. However, because the VMs running on a host use the physical systems on that host, if you have a need for a powerful VM, you'll need a powerful physical computer to host it.

VMs offer many more options than a container instance. You can choose a VM that has the CPU, memory, and disk configuration that best meets your needs, and the operating system that runs on your VM is the full operating system, so you can use the VM just as you would a physical computer sitting at your desk.

As you'll see in the next section, Azure offers many different options for VMs.

Functions

A common use of compute resources is *microservices*. You can think of a microservice as a small component that has a specific capability. A full solution would involve connecting many microservices together, along with other required components. These microservices take input, perform some operation on that input, and then return a result. They may also kick off an event such as sending an email, passing off data to another component, triggering a database update, and so forth.

The time that any single microservice runs is small, so it usually makes no sense to run a microservice on a VM. It is a common scenario to run a microservice in a container instance, but there are other options available in Azure.

Azure Functions is a service that runs on Azure App Service, and Functions is well-designed to accommodate a microservices architecture. When Functions runs a microservice, it can do so on a "spare" VM in App Service. This allows you to save costs because you only pay for the execution time of the function.

Options for Azure virtual machines

By using Azure virtual machines, you can take advantage of powerful host computers that Microsoft makes available when you need computing power, and when you no longer need that power, you no longer have to pay for it.

> ***NOTE*** **USING AZURE**
>
> **In the following steps, you'll create an Azure virtual machine. This requires that you have an Azure subscription. If you don't have an Azure subscription, you can create one at *https://azure.microsoft.com/en-us/free/*.**

To create an Azure virtual machine, log in to the Azure portal using your Azure account and then follow these steps, as shown in Figures 2-11 through 2-14.

1. Click **Create A Resource**.
2. Click **Compute**.
3. Under **Popular Marketplace Products**, click **Create** under **Ubuntu Server 20.04 LTS**. See Figure 2-11.

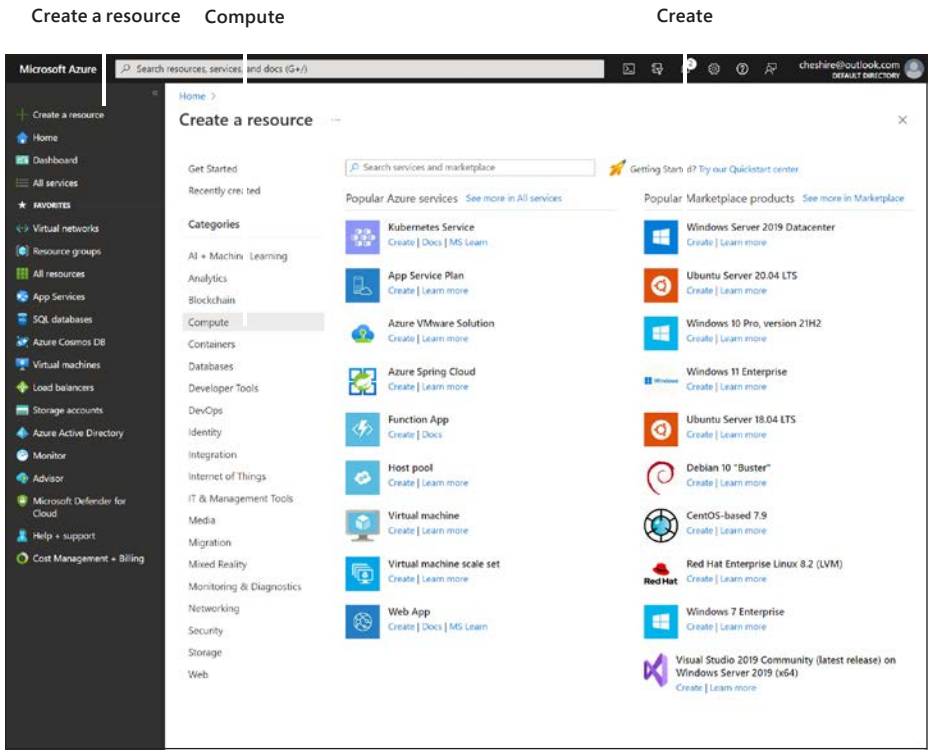


FIGURE 2-11 Creating a virtual machine

4. Next to **Resource Group**, click **Create New** to create a new resource group.
5. Enter **TestRG** as the resource group name and click **OK**.
6. Enter **TestVM** as your VM name, as shown in Figure 2-12.

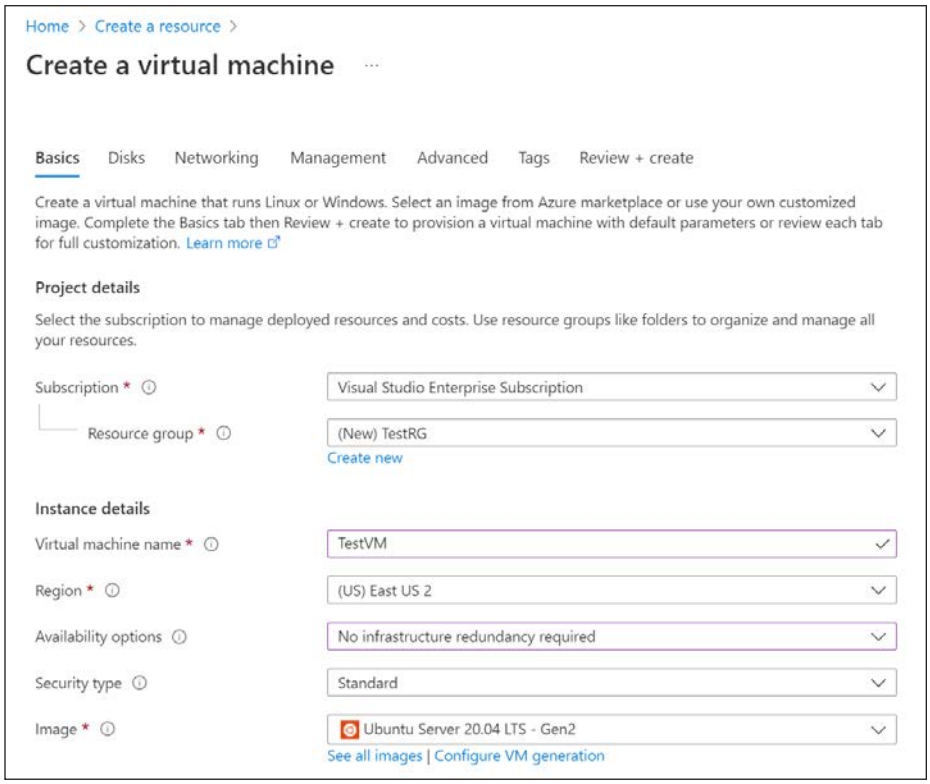


FIGURE 2-12 Virtual machine settings

7. Scroll down and select **Password** for the authentication type.
8. Enter a username for your administrator account.
9. Enter a password you'd like to use for your administrator account.
10. Confirm the password.
11. Leave all the other settings as they are and click the **Next** button three times to move to the **Management** screen.
12. In the **Monitoring** section, set **Boot Diagnostics** to **Disable**.

13. Click **Review + Create** to create your VM. See Figure 2-13.

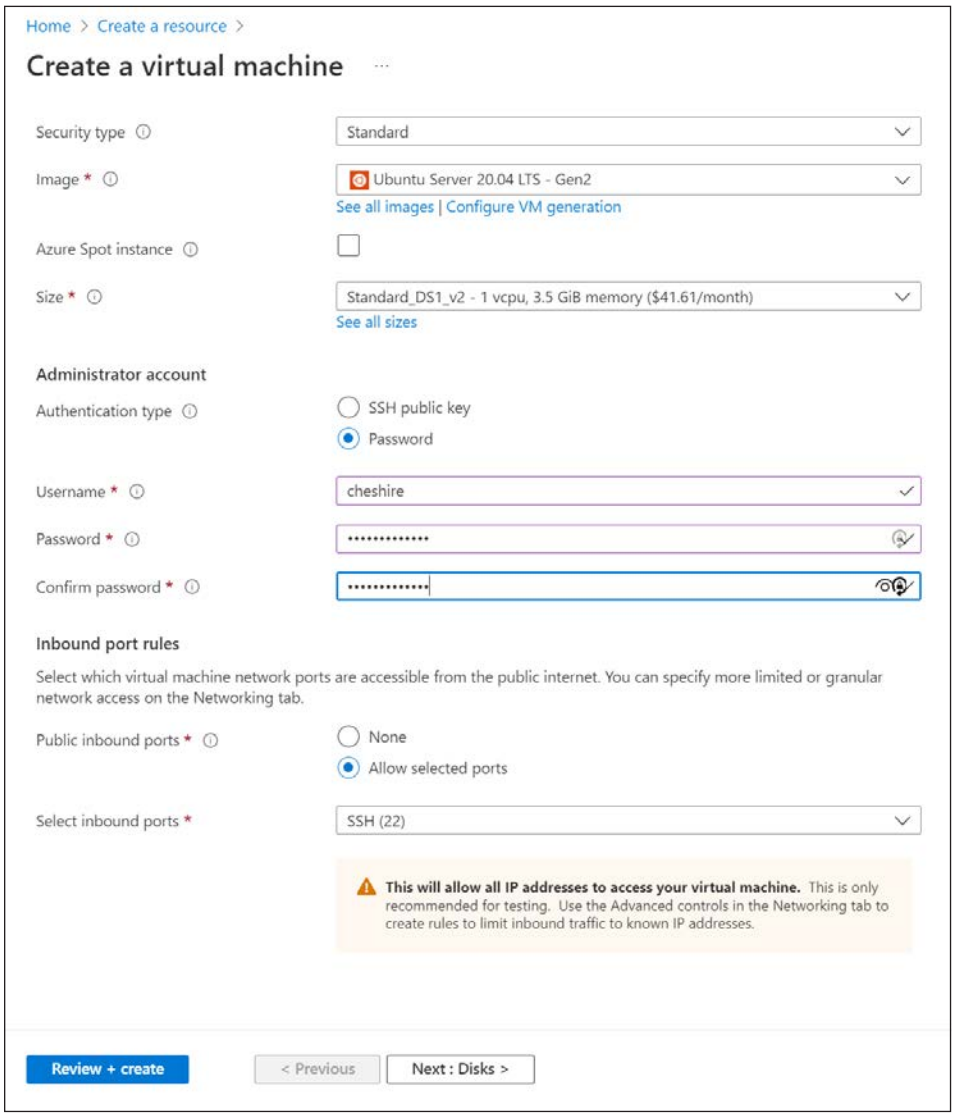


FIGURE 2-13 Virtual machine account settings

After you click **Review + Create**, Azure will validate your settings to make sure you haven't left anything out. Once your validation has passed, you will see a **Create** button. Click the **Create** button to start the deployment of your new VM, as shown in Figure 2-14.

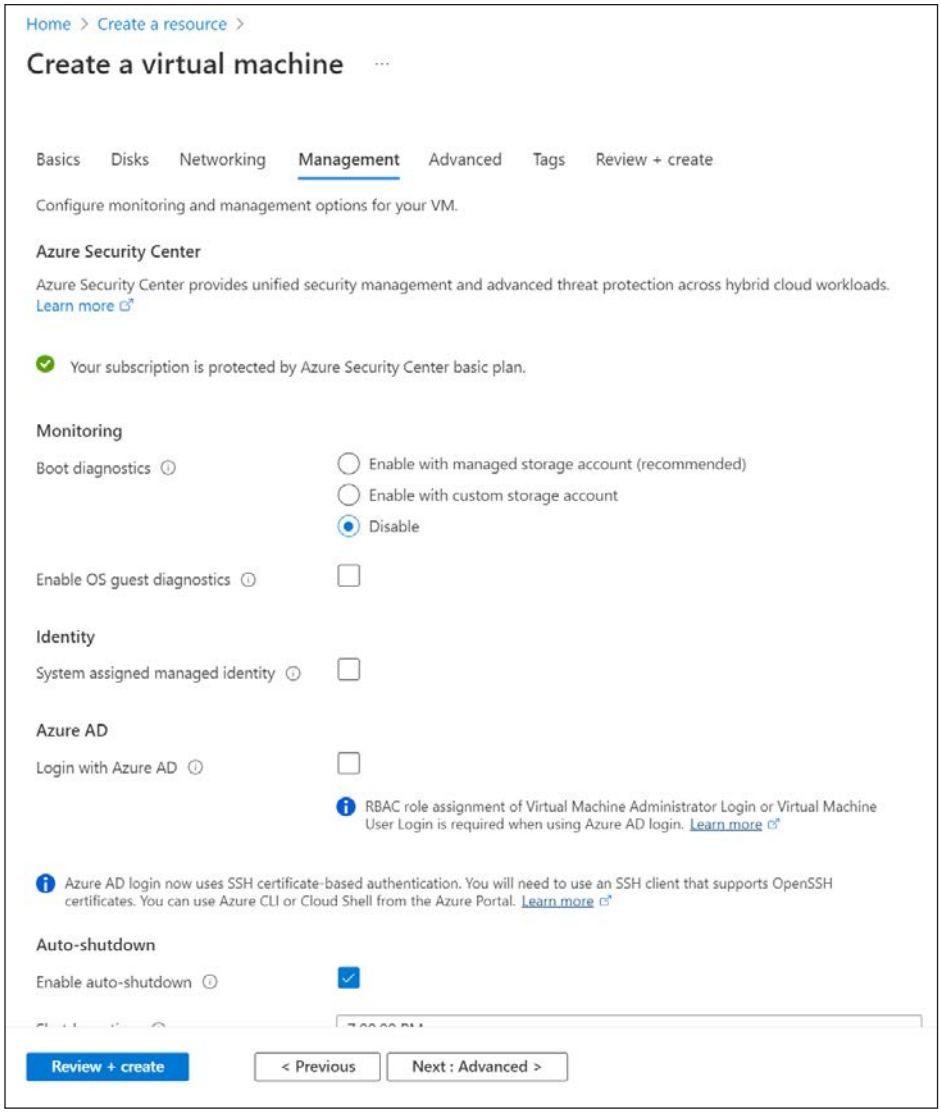


FIGURE 2-14 Virtual machine management settings

As your VM is being deployed, you'll see the status displayed in the Azure portal, as shown in Figure 2-15. You can see the Azure resources that are created to support your VM. You can see the resource name, the resource type, and the status of each resource.

Once all the resources required for your VM are created, your VM will be considered fully deployed. You'll then be able to click the **Go To Resource** button to see the management interface for your VM in the Azure portal, as shown in Figure 2-16.

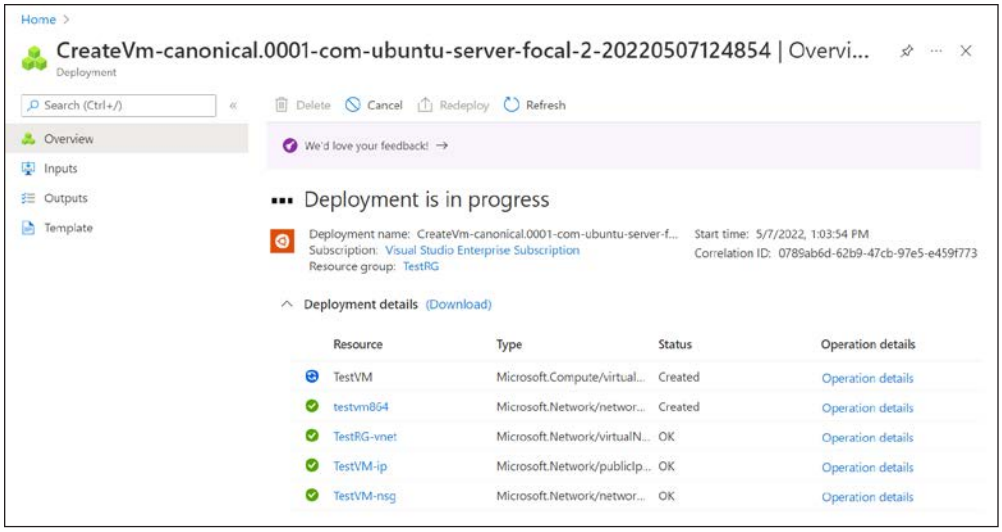


FIGURE 2-15 Virtual machine deployment

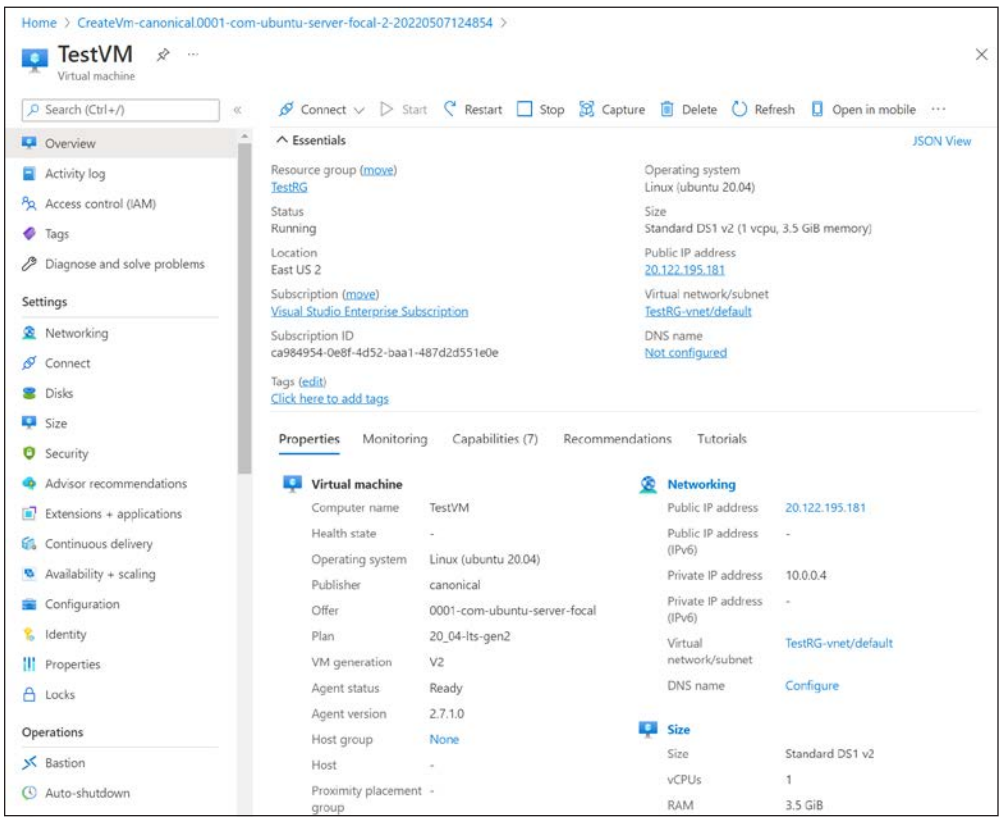


FIGURE 2-16 Viewing a virtual machine

The new VM is a guest on a physical computer in an Azure datacenter. In that datacenter is a physical rack of computer servers, and our VM is hosted on one of those servers. The host computer is managed by Microsoft, but you manage the VM because this is an IaaS offering in Azure.

> ***NOTE*** **VMs AND BILLING**
>
> **You are charged for Azure VMs as long as they are running, and using the default settings as we have here led to a few expensive options. To stop billing for this VM, click the Stop button at the top of the screen shown in Figure 2-16. Azure will save the current state of the VM, and billing will stop. You won't be able to use the VM while it's in a stopped state, but you will also avoid the billing of that VM. Keep in mind that unless you reserve the IP address for your VM, your IP address will likely change the next time you start it.**
>
> **You can also stop a VM from within the guest operating system on the VM, but when you do that, you will still be charged for the resources the VM uses because it's still allocated to you. That means you'll still incur charges for managed disks and other resources. Deleting the `TestRG` resource group will ensure you aren't charged for the VM.**

As of right now, this VM is susceptible to downtime due to three types of events: *planned maintenance, unplanned maintenance*, and *unexpected downtime*.

Planned maintenance refers to planned updates that Microsoft makes to the host computer. This includes things like operating system updates, driver updates, and so on. In many cases, updates won't affect your VM, but if Microsoft installs an update that requires a reboot of the host computer, your VM will be down during that reboot.

Azure has underlying systems that constantly monitor the health of computer components. If one of these underlying systems detects that a component within the host computer might fail soon, Azure will flag the computer for unplanned maintenance. In an unplanned maintenance event, Azure will attempt to move your VM to a healthy host computer. When it does this, it preserves the state of the VM, including what's in memory and any files that are open. It only takes Azure a short time to move the VM, during which time it's in a paused state. In a case where the move operation fails, the VM will experience unexpected downtime.

To ensure reliability when a failure occurs in a rack within the Azure datacenter, you can (and you should) take advantage of a feature called *availability sets*. *Availability sets* protect you from maintenance events and downtime caused by hardware failures. To do that, Azure creates some underlying entities in an availability set called *update domains* and *fault domains*. (In order to protect yourself in the event of maintenance events or downtime, you must deploy at least two VMs into your availability set.)

Fault domains are a logical representation of the physical rack in which a host computer is installed. By default, Azure assigns two fault domains to an availability set. If a problem occurs in one fault domain (one computer rack), the VMs in that fault domain will be affected, but VMs in the second fault domain will not. This protects you from unplanned maintenance events and unexpected downtime.

Update domains are designed to protect you from a situation where the host computer is being rebooted. When you create an availability set, Azure creates five update domains by default. These update domains are spread across the fault domains in the availability set. If a reboot is required on computers in the availability set (whether host computers or VMs within the availability set), Azure will only reboot computers in one update domain at a time and it will wait 30 minutes for computers to recover from the reboot before it moves on to the next update domain. Update domains protect you from planned maintenance events.

Figure 2-17 shows a representation of an availability set containing five VMs. There are two fault domains and three update domains. When VMs were created in this availability set, they were assigned as follows:

- The first VM is assigned Fault Domain 0 and Update Domain 0.
- The second VM is assigned Fault Domain 1 and Update Domain 1.
- The third VM is assigned Fault Domain 0 and Update Domain 2.
- The fourth VM is assigned Fault Domain 1 and Update Domain 0.
- The fifth VM is assigned Fault Domain 0 and Update Domain 1.

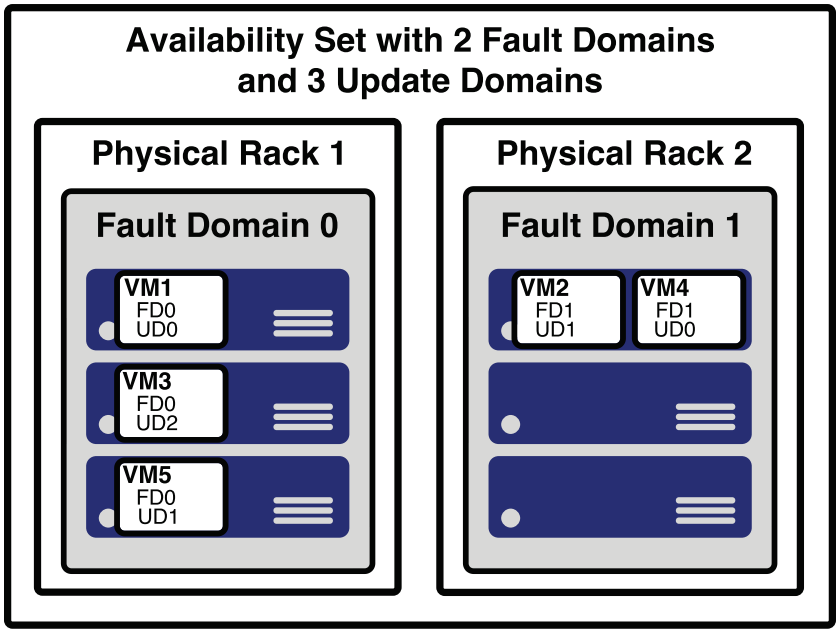


FIGURE 2-17 A representation of an availability set

You can verify the placement of fault domains and update domains by creating five VMs in an availability set with two fault domains and three update domains. If you then look at the availability set created in the Azure portal, as shown in Figure 2-18, you can see the same configuration depicted in Figure 2-17.

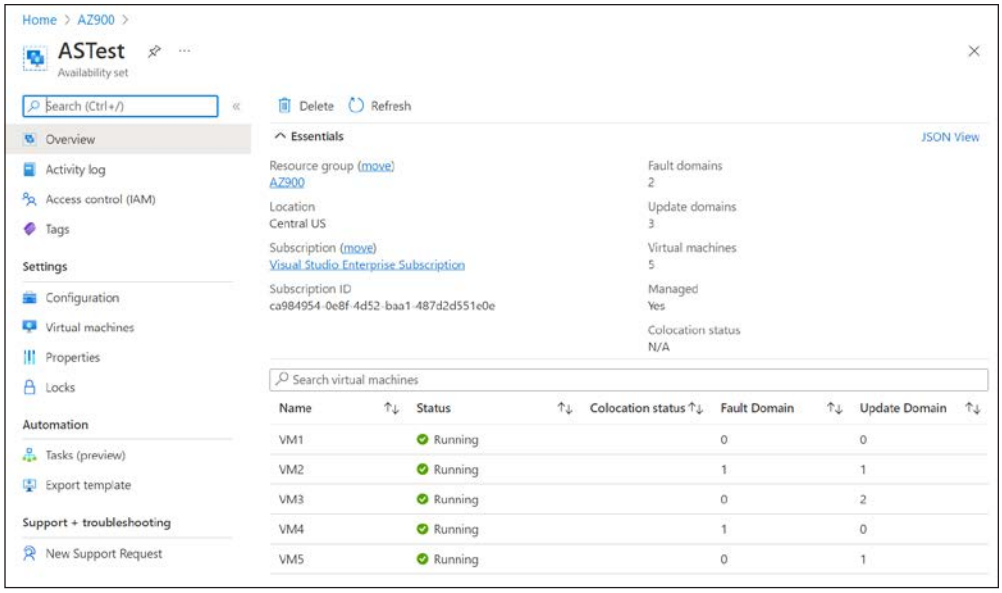


FIGURE 2-18 An availability set in the Azure portal showing fault domains and update domains

Notice in Figure 2-18 that the availability set is named `ASTest`. In this availability set, we run five VMs that are all running a web server and host the website for an application. Suppose you need a database for this application, and you want to host that database on VMs as well. In that situation, you would want to separate the database VMs into their own availability set. As a best practice, you should always separate your workloads into separate availability sets.

Availability sets certainly provide a benefit in protecting from downtime in certain situations, but they also have some disadvantages. First, every machine in an availability set must be explicitly created. While you can use an ARM template to deploy multiple virtual machines in one deployment, you still must configure those machines with the software and configuration necessary to support your application.

An availability set also requires that you configure something in front of your VMs that will handle the distribution of traffic to those VMs. For example, if your availability set is servicing a website hosted on the VMs, you'll need to configure a load balancer that will handle the job of routing users of your website to the VMs that are running it.

Another disadvantage of availability sets relates to cost. In a situation where your VM needs to be changed often based on things like load on the application, you might find yourself paying for many more VMs than you need.

Azure offers another feature for VMs called *scale sets* that solves these problems nicely. When you create a scale set, you tell Azure what operating system you want to run and then you tell Azure how many VMs you want in your scale set. You have many other options such as creating a load balancer or gateway and so forth. Azure will create as many VMs as you specify (up to 1,000) in one easy step.

MORE INFO **USING A CUSTOM IMAGE**

The default set of templates for VMs are basic and include only the operating system. However, you can create a VM, install all the necessary components you need (including your own applications), and then create an image that can be used when creating scale sets.

For more information on using custom images, see *https://bit.ly/az900-customvmimages*.

Scale sets are deployed in availability sets automatically, so you automatically benefit from multiple fault domains and update domains. Unlike VMs in an availability set, however, VMs in a scale set are also compatible with availability zones, so you are protected from problems in an Azure datacenter.

As you might imagine, you can also scale a scale set in a situation where you need more or fewer VMs. You might start with only one VM in a scale set, but as the load on that VM increases, you might want to automatically add additional VMs. Scale sets provide that functionality by using Azure's auto-scale feature. You define scaling rules that use metrics like CPU, disk usage, network usage, and so forth. You can configure when Azure should add additional instances and when it should scale back and deallocate instances. This is a great way to ensure availability while reducing costs by taking advantage of the elasticity that auto-scale provides.

MORE INFO **SCALING AND AVAILABILITY SETS**

Before the introduction of scale sets, you had the ability to configure auto-scale rules for an availability set. You'll probably still see third-party documentation and training that talks about scaling availability sets, but that functionality has been replaced with scale sets.

Microsoft guarantees an SLA of 99.95 percent when you use a multi-VM deployment scenario, and for most production scenarios, a multi-VM deployment is preferred. However, if you use a single-instance VM, and you use premium storage, Microsoft guarantees a 99.9 percent SLA. Premium storage uses solid-state drives (SSDs) that are located on the same physical server that is hosting the VM for enhanced performance and uptime.

An Azure VM is a great choice if you need the flexibility to run your own applications in a virtualized environment with the greatest amount of flexibility, but many businesses today are turning to desktop virtualization instead of running multiple VMs.

In a desktop virtualization model, a business installs an operating system and applications on one central server. The desktop virtualization infrastructure makes it possible for employees to access the operating system and applications from virtually any device, provided it has access to the network. The OS and applications aren't downloaded to the employee's device. Instead, the employee uses the applications in a virtualized environment that makes it feel like the applications are running locally.

Most businesses have applications that all their employees need to use. For example, employees might need access to Microsoft Word, Microsoft Excel, Microsoft Outlook, and so on. In many situations, businesses fill this need by purchasing a Microsoft Office license for all employees and installing Office apps on each computer.

This classic model of each employee using one computer with applications installed on it is not only inefficient for businesses, but it's also insecure. First off, it requires that the business purchase operating system and application licenses for each employee. It also requires that the IT department be available to troubleshoot any operating system or application issues. Users of local applications also have data that is stored on the local hard drive, and this represents a security risk if an unwanted person gets access to the machine.

MORE INFO **DESKTOP VIRTUALIZATION**

If you've ever used Windows Remote Desktop to remote into another computer, you've experienced something like desktop virtualization. The difference is that desktop virtualization allows you to virtually access the operating system and applications that an administrator has installed for remote access.

Desktop virtualization might sound like the perfect solution for many businesses, but in fact, it's quite complex to configure, and it requires many components to ensure a secure environment. For that reason, Microsoft developed a service called Azure Virtual Desktop.

Azure Virtual Desktop is a PaaS offering in Azure that provides desktop virtualization that is managed by Microsoft. It requires a bit of advanced configuration, but once you have it configured, the infrastructure is entirely managed by Microsoft.

To use Azure Virtual Desktop (AVD), you first create an AVD *tenant*. A tenant is a collection of one or more *host pools*, and a host pool consists of both *session hosts* and one or more app groups that represent the applications and OS desktops users should be able to access. These session hosts are simply Azure VMs that you've configured for AVD.

Once you've set up the tenant, you can add users from your Azure Active Directory so that they can access the OSes and apps in your tenant and assign permissions to them. Those users can then access AVD using the following methods.

- Using the AVD client app for Windows
- Using the AVD client app for macOS
- Using the AVD client for iOS
- Using the AVD client for Android
- Using the web-based client from any web browser

MORE INFO **AZURE VIRTUAL DESKTOP**

For more information on Azure Virtual Desktop, including requirements for using it and a guide to configuring it, browse to *https://bit.ly/AZ900-avd*.

Users accessing AVD see a list of OSes and applications they can use. When the user clicks an OS, they can interact with that OS just as though they were running it on their local machine. When the user clicks an app, the app launches in a virtual session, but it looks exactly as if it's running locally. Better yet, using technology Microsoft acquired called FSLogix, AVD provides a local profile while the user is using apps. This capability even allows for users to use files in Microsoft OneDrive along with AVD.

> ***MORE INFO*** **WINDOWS 10 MULTI-USER**
>
> **Microsoft developed a special version of Windows 10 called Windows 10 Multi-User to support the functionality of Azure Virtual Desktop.**

Resources required for virtual machines

In the previous section, we walked through the creation of an Azure virtual machine. The process was relatively simple, and once it was complete, we looked at the resulting virtual machine in the Azure portal. What might not have been obvious is that many resources were created automatically by Azure in addition to the virtual machine itself. Let's look at the different resources that are required for a virtual machine to function.

Figure 2-19 shows a resource group in the Azure portal. The only resource I've explicitly created in this resource group is a virtual machine.

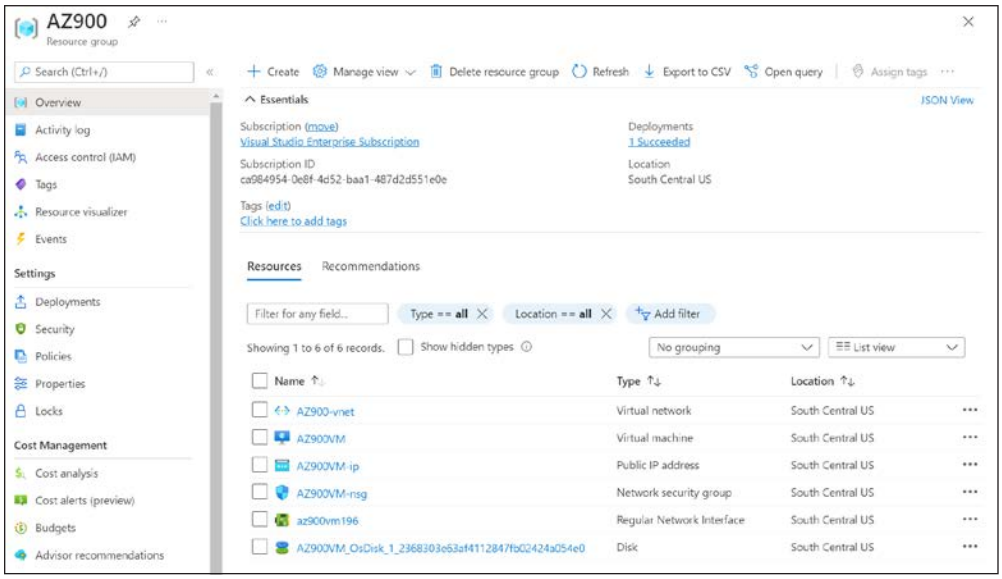


FIGURE 2-19 A resource group showing all the required resources for a VM

As you can see, there are many resources in the resource group other than the VM. Those resources include a virtual network, a public IP address, a network security group, a network interface, and a disk. All these resources work together to provide the functionality of my VM.

All but one of these additional resources are provided for network connectivity. Remember that a VM is a virtualized computer that runs on a host computer. For the VM to have network connectivity, it must have a network card, so Azure creates a virtual network interface when you create a VM. That network interface is connected to a virtual network, and that virtual network provides connectivity to the physical networks that Azure uses.

> ***MORE INFO*** **AZURE VIRTUAL NETWORKS**
>
> **You'll learn about Azure virtual networks in the "Virtual networking" section later in this chapter.**

To provide internet connectivity for the VM, it requires a public IP address. Azure creates a public IP address resource for the VM, which is bound to the virtual network interface.

Azure also creates a network security group and associates it with the virtual network interface. This network security group makes it possible to allow or deny traffic to the VM based on rules that you configure. Azure creates some built-in rules that allow the VM to communicate with the virtual network. If you configured one or more ports when you created your VM (for example, allowing RDP access over port 3389 on a Windows VM), Azure will add a rule to the network security group to allow that.

> ***MORE INFO*** **NETWORK SECURITY GROUPS**
>
> **You'll learn more about network security groups in Skill 2.4, "Describe Azure identity, access, and security."**

The one resource not related to networking is the disk that is created with your VM. This disk is the operating system disk for your VM, and its purpose is to store the files necessary for the operating system. This is not a persistent disk, and that means data written to the disk does not persist between VM restarts. If you want files to persist, you'll need to create a data disk for the VM.

> ***MORE INFO*** **DATA DISKS FOR VMs**
>
> **Data disks aren't technically required, but it's uncommon not to use a data disk with a VM. You'll find out about data disks in Skill 2.3, "Describe Azure Storage services."**

EXAM TIP

There are three other resources required for a virtual machine. As with any other Azure resource, a VM requires a management group (either the Tenant Root Group or one you explicitly create), an Azure subscription, and a resource group.

This might seem obvious, but it's easy to overlook this requirement, and I wouldn't rule out possibly seeing a question related to it on the exam.

Application hosting options

Azure offers many options for hosting applications ranging from simple, single-tiered applications to complex applications using multiple systems such as web servers, database servers, middleware components, and more. There are options available that give you complete control over the infrastructure using an IaaS approach, and there are numerous options that offload some of that responsibility to the cloud provider using a PaaS approach.

Web Apps on Azure App Service

Azure App Service is a PaaS offering in Azure that makes it easy to host a web app in the cloud. In addition to basic web hosting services, App Service also offers many additional features that you can easily add to your web app, often with the flip of a switch within the Azure portal.

When you create a web app in Azure App Service, your app runs on an Azure virtual machine that is preconfigured specifically for App Service. Depending on the tier of service you use when you create your app, it will either run on a VM that is shared among many users or a VM that is dedicated to you.

Figure 2-20 shows a diagram of the basic App Service architecture. This diagram is simplified, but it illustrates the basics of how App Service works. Azure Load Balancer distributes traffic to a special VM within App Service called a *front end*. The front end is running special software that allows it to effectively distribute traffic to the VMs that are actually running your web app. These VMs run inside of an *App Service plan*, a logical container for one or more VMs that are running your web app.

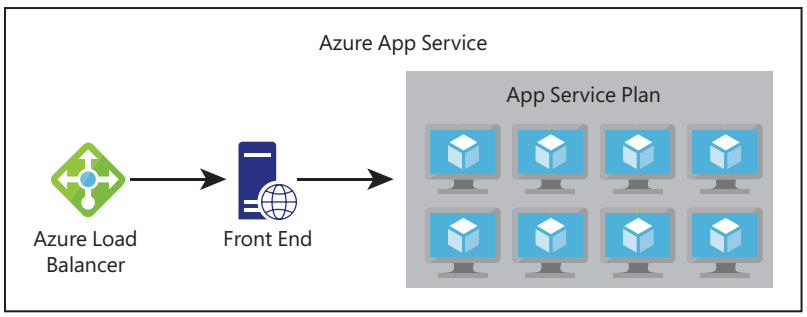


FIGURE 2-20 A high-level representation of Azure App Service

Every web app you create in App Service runs inside of an App Service plan. An App Service plan is created within a specific Azure region, and it specifies how many VMs your app runs on and the properties of those VMs.

> ***NOTE*** **APP SERVICE PLANS**
>
> **In the example in this chapter, a single web app is running in an App Service plan. However, multiple apps can run inside a single App Service plan. All apps in an App Service plan will share the same VMs in that App Service plan.**

In Figure 2-21, an App Service plan named `AZ900-Plan` is being created in the Central US region. The VMs in this App Service plan will run Windows and will be created in the Standard S1 App Service pricing tier. You can click **Change Size** to change the pricing tier before the App Service plan is created, and you can also scale the App Service plan at any point to change the size.

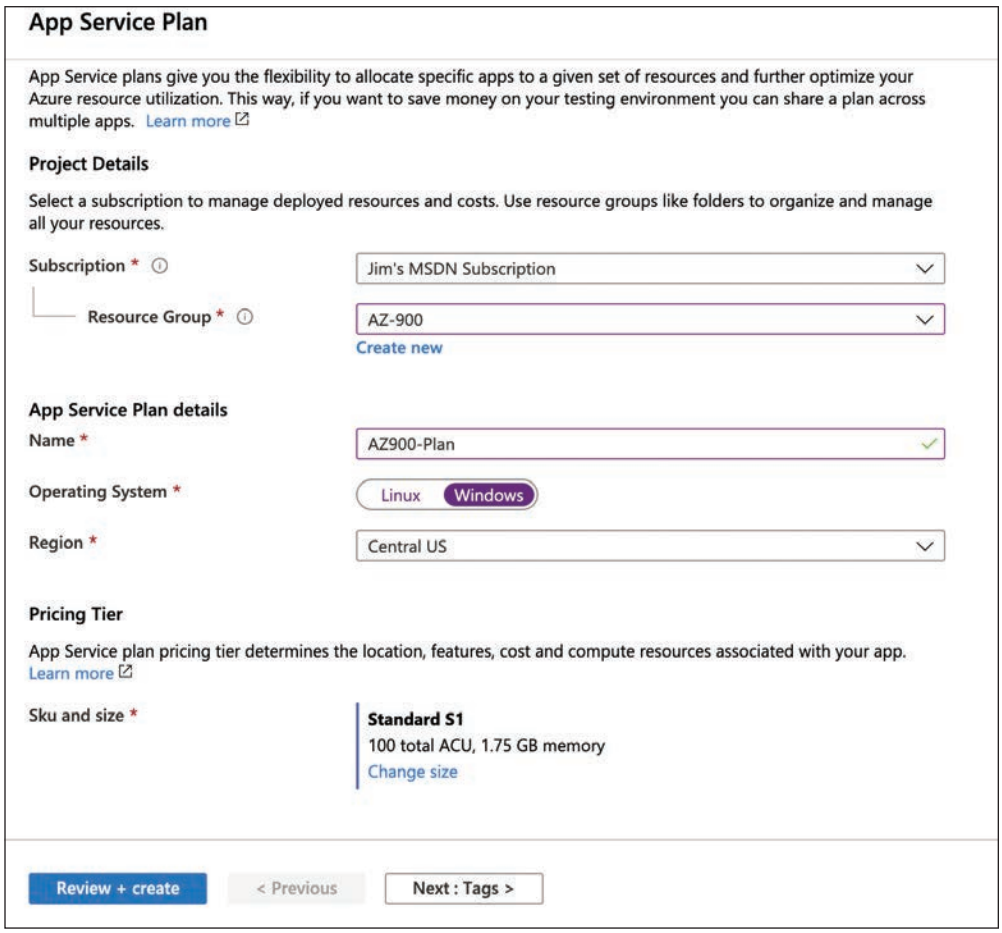


FIGURE 2-21 Creating an App Service plan in the Central US region

The following pricing tiers are available in App Service:

- **Free** A no-cost tier for testing only that runs on VMs shared with other App Service customers.
- **Shared** A low-cost tier for testing only with some additional features not offered in the Free tier. Runs on VMs shared with other App Service customers.
- **Basic, Standard, Premium, and PremiumV2** Higher-cost tiers that offer many additional features. Runs on dedicated VMs that are not shared with other customers.

EXAM TIP

You are charged for App Service plans even when no web apps are running in them. If you do have web apps in your App Service plan, you are still charged if you stop the web apps. The only way to avoid being billed for an App Service plan is to delete it.

When you move from a lower pricing tier to a higher pricing tier, you are scaling up. You can also scale down at any time by moving to a lower pricing tier. If you are running in the Basic, Standard, Premium, or PremiumV2 tier, you can also scale out to multiple VMs. The Basic tier allows you to scale to a maximum of 3 VMs (or *instances*), the Standard tier allows for 10 instances, and the Premium and PremiumV2 tiers allow for up to 20 instances.

MORE INFO **APP SERVICE VIRTUAL MACHINES**

Creating a web app in App Service is very fast, and scaling it out to multiple instances is also very fast. That's because the VMs that are running App Service web apps are already up and running. When you create a web app, you are simply allocating an existing VM for your use.

When you create a new web app, you can create it in an existing App Service plan, or you can create a new App Service plan for the app. All apps in an App Service plan run on the same VMs, so if you are already stressing the resources of an existing App Service plan, your best choice might be to create a new App Service plan for your new web app.

App Service allows you to choose between a VM preconfigured with a runtime stack (such as Java, .NET, PHP, and so forth), a Docker container, or a static web app. If you choose to run a preconfigured runtime stack, you can choose between multiple versions that App Service provides.

MORE INFO **STATIC WEB APPS**

Static web apps are web apps that are automatically connected to a code repository. They allow developers to automatically update the code a web app uses based on changes to source code running in GitHub or Azure DevOps.

Figure 2-22 shows a web app being created in the AZ900-Plan App Service plan. This new web app will run on a VM that is configured to run .NET Core 3.1 apps on a Windows VM.

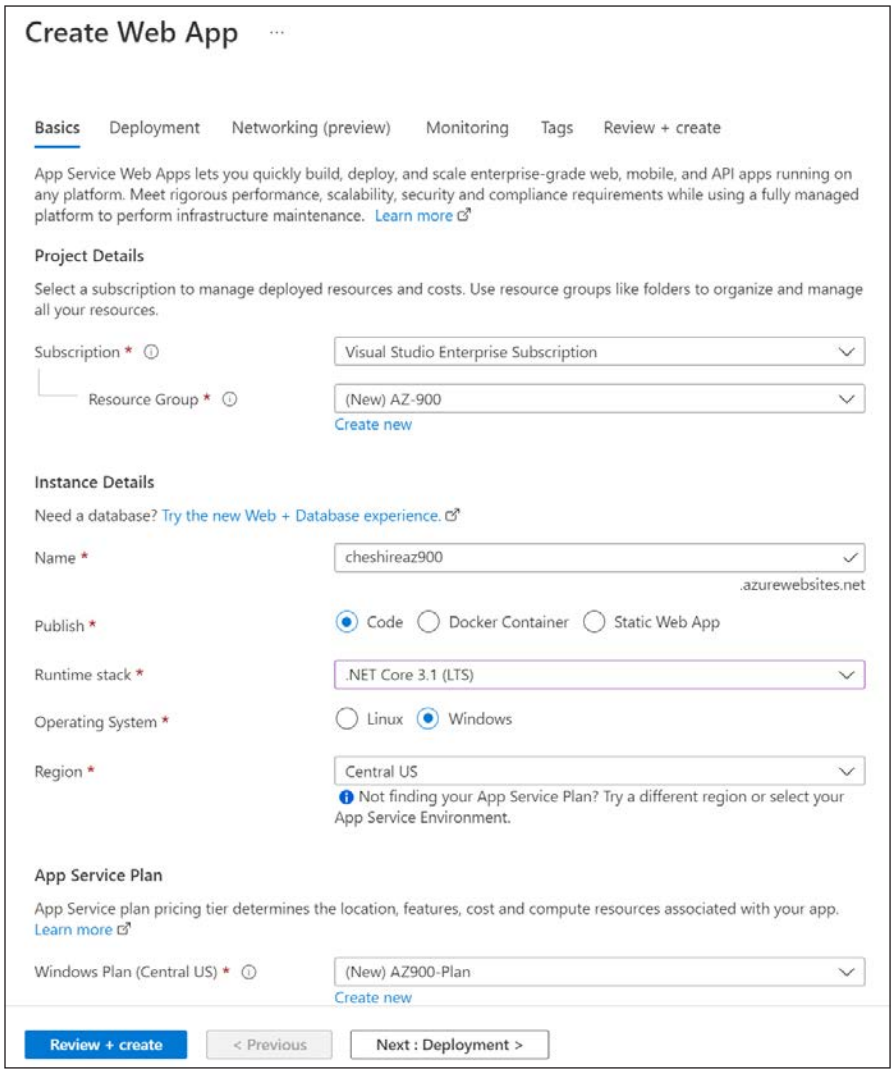


FIGURE 2-22 Creating a web app to run a .NET Core 3.1 website

Configuring and managing your web app is extremely easy. Because App Service is a PaaS service, you are only responsible for your code. Microsoft manages the features available to you. In Figure 2-23, you can see many of the features available in App Service, including the ability to quickly and easily scale out when needed.

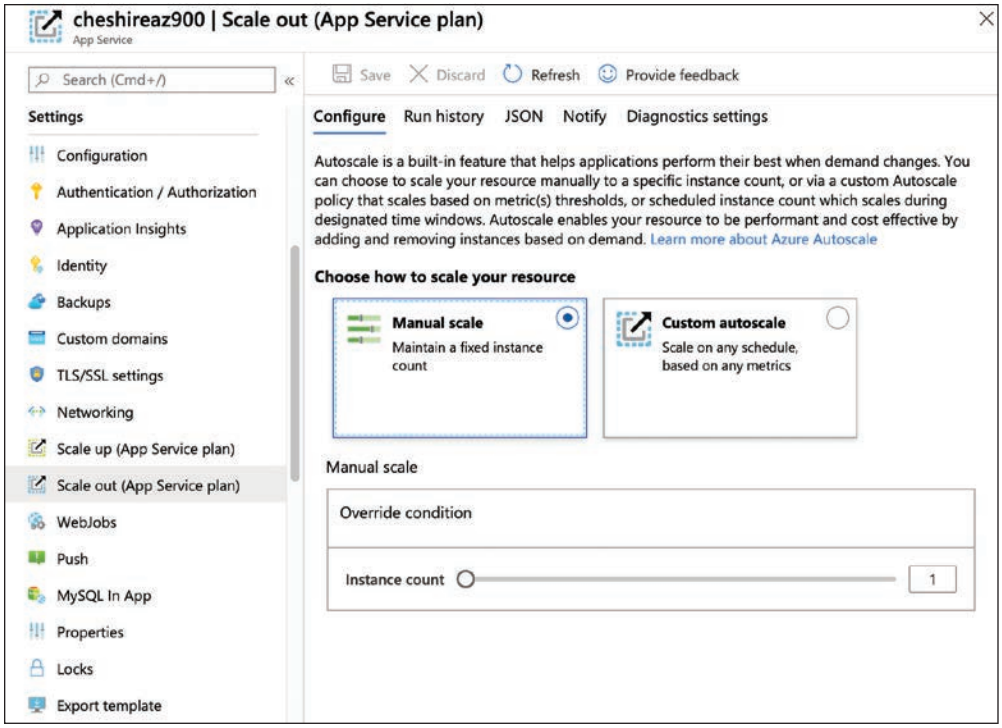


FIGURE 2-23 Settings for a web app make it easy to add features and scale your app

Azure Kubernetes Service (AKS)

In the previous section, we talked about using Azure Container Instances (ACI) to easily create a container to host your application. ACI makes running a container simple and fast, but there are limitations. If you want a powerful and scalable hosting option for containers, Azure Kubernetes Service (AKS) is a better option.

> ***NOTE*** **CONTAINERS IN APP SERVICE**
>
> **I mentioned earlier that you can host Docker containers in a web app running in App Service. While that's certainly one option for container hosting, AKS is still a better choice for enterprise hosting or hosting large deployments where maximum control is needed over scaling of your containers.**

Kubernetes is a container orchestration service. This means that it's responsible for monitoring containers and ensuring that they're always running. It can also scale to add additional containers when the needs require it to, and it can then scale back when the needs are reduced.

Kubernetes creates containers in a *pod*. A pod is a group of related containers, and containers within a pod can share resources. This is one of the advantages to using Kubernetes because it releases you from the resource-sharing restriction typically imposed in a multi-container environment. However, a container in one pod is not able to share resources with a container in another pod.

The computer that Kubernetes pods are running on is called a *node* or a *worker*. This computer must have a container runtime such as Docker running on it. In addition to pods, the node also runs several services that are required for Kubernetes to manage the pods and so on. There will typically be multiple nodes within a Kubernetes instance, and they are all controlled by a node called the Kubernetes *control plane*. The entire environment of the control plane and all its nodes is called a Kubernetes *cluster*.

A Kubernetes control plane contains all the configuration and services necessary to manage the orchestration of pods and other Kubernetes entities. Configuring a cluster can be complex, and it is by far the most laborious task of using Kubernetes. For that reason, services such as Azure Kubernetes Service (AKS) are becoming more popular.

AKS offloads the burden of dealing with the Kubernetes cluster to Microsoft. When you create a Kubernetes cluster in AKS, Azure creates the control plane and the nodes for you. All you have to do is deploy your containers and you're up and running with a managed Kubernetes cluster.

AKS simplifies the creation of a Kubernetes cluster, but it also makes it extremely easy to manage a cluster. Operations such as upgrading a cluster or scaling a cluster are simple using the Azure portal menu options. You can also get detailed information on your cluster, including each node that's running in the cluster.

While AKS makes adopting and managing Kubernetes easier, it doesn't completely obfuscate Kubernetes. To deploy your applications, you still need to understand how to use Kubernetes, and in some cases, you'll need to use the Kubernetes command line. Azure, however, makes it far easier than doing all the legwork and maintenance yourself. Even better, AKS in Azure is free. You only pay for the Azure compute resources you use within your cluster.

Azure Spring Cloud

If you're a Java developer, you've no doubt heard of (and probably have used) Spring. Spring is a powerful framework for easily developing and running Java apps. Spring also offers many powerful extensions, allowing Java developers to easily add powerful functionality to their apps.

Azure Spring Cloud is an Azure service that makes hosting Spring apps in Azure fast and easy. Spring Cloud offloads the management burden of the infrastructure to Azure, allowing you to focus only on your code. As with other PaaS services, Spring Cloud offers a lot of features to deploy, monitor, and manage your apps.

Spring Cloud also seamlessly integrates with other Azure services such as Azure SQL Database, Azure Storage, and so forth.

Virtual machines

The application hosting options mentioned up to this point are all PaaS services. They make application hosting easier because they remove your responsibility for creating and managing the infrastructure necessary for application hosting. However, if you want maximum control over the configuration and management of the underlying infrastructure when hosting your apps, you can use virtual machines and configure everything yourself.

When using virtual machines for application hosting, you'll need to ensure you create the resources and configure everything for maximum security, and all of that is left to you. For example, if you're going to host a web app using VMs, you'll need to install and configure a web server for your app, and you'll need to take the necessary steps to ensure security. You'll also likely want to install a firewall of some sort, possibly a load balancer to distribute load across multiple VMs, and so forth.

As you can see, VMs provide you with maximum flexibility and control, but they also represent a significant increase in responsibility for configuration and management, and if something goes wrong with any of the components of the application, fixing it will likely be your responsibility.

Virtual networking

An Azure virtual network (often called a *VNet*) allows Azure services to communicate with each other and with the internet. You can even use a VNet to communicate between your on-premises resources and your Azure resources. As we have already discussed, when you create a virtual machine in Azure, Azure creates a VNet for you. Without that VNet, you wouldn't be able to remote into the VM or use the VM for any of your applications. However, you can also create your own VNet and configure it any way you choose.

An Azure VNet is just like any other computer network. It's composed of a network interface card (a NIC), IP addresses, and so on. You can break up your VNet into multiple subnets and set up a portion of your network's IP address space for those subnets. You can then configure rules that control the connectivity between those subnets.

Figure 2-24 illustrates an Azure VNet that we might use for a multi-tier application. The VNet uses IP addresses in the 10.0.0.0 address range, and each subnet has its own range of addresses. IP address ranges in VNets are specified using classless inter-domain routing (CIDR) notation, and a discussion of that is far outside the scope of this exam. However, with the configuration shown in Figure 2-24, we have 65,536 IP addresses available in our VNet, and each subnet has 256 IP addresses allocated to it. (The first four IP addresses and the last IP address in the range are reserved for Azure's use, so you only have 251 addresses to use in each subnet.) This is a typical design because you still have many addresses available in your network for later expansion into additional subnets.

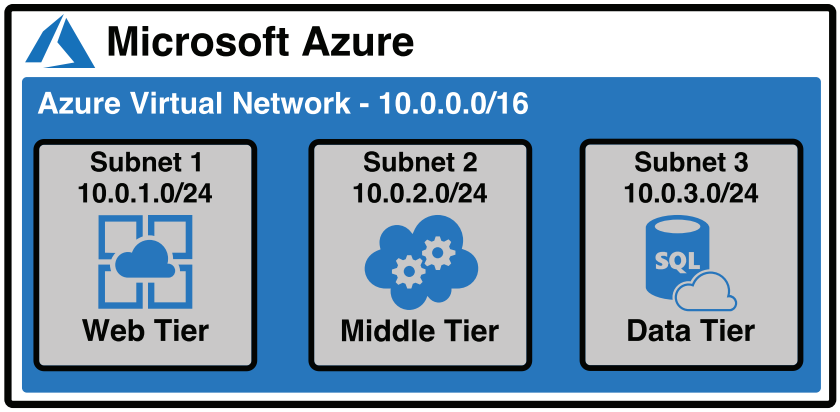


FIGURE 2-24 A multi-tier application in an Azure Virtual Network

In most cases, you create VNets before you create the resources that use them. As I said earlier, when you create a VM in Azure, a VNet is created automatically. Azure does that because you can't use a VM unless there's a network associated with it. While you can connect a VM you are creating to an existing VNet, you can't connect a VM to a VNet after it's been created. For that reason, if you wanted to use your own VNet instead of the one Azure creates automatically, you would create your VNet before you create your VM.

The web tier shown previously in Figure 2-24, on the other hand, is running in Azure App Service, a PaaS offering. This is running on a VM that Microsoft manages, so Microsoft creates and manages the VM and its network. To use that tier with the VNet, App Service offers a feature called VNet Integration that allows you to integrate a web app in App Service with an existing VNet.

The IP addresses within the VNet at this point are all private IP addresses. They allow resources within the VNet to talk to each other, but you can't use a private IP address on the internet. You need a public IP address to give the internet access to your web tier.

MORE INFO **OUTBOUND INTERNET CONNECTIVITY**

A public IP address doesn't have to be assigned to a resource for that resource to connect outbound to the internet. Azure maintains a pool of public IP addresses that can be dynamically assigned to a resource if it needs to connect outbound. That IP address is not exclusively assigned to the resource, so it cannot be used for inbound communication from the internet to the Azure resource.

Because the web tier is running on Azure App Service (a PaaS service), Microsoft manages the public-facing network for us. You get internet access on that tier without needing to do anything. If you want to run the web tier on an IaaS VM instead, configure the public IP address for the web tier. In those situations, Azure allows you to create a Public IP Address resource and assign it to a virtual network.

Virtual network peering

In scenarios where you need resources to communicate across VNets, you can connect your VNets to each other using peering. traffic between two VNets that are peered travels over Microsoft's private backbone infrastructure and not over the internet VNet.

EXAM TIP

You can peer VNets that are in the same region or in different regions. Microsoft refers to peering VNets between two Azure regions as *global virtual network peering*.

To connect two VNets using virtual network peering, open your VNet and click **Peerings** on the menu on the left in the Azure portal. You can then click **+Add** to add a virtual network peering, as shown in Figure 2-25.

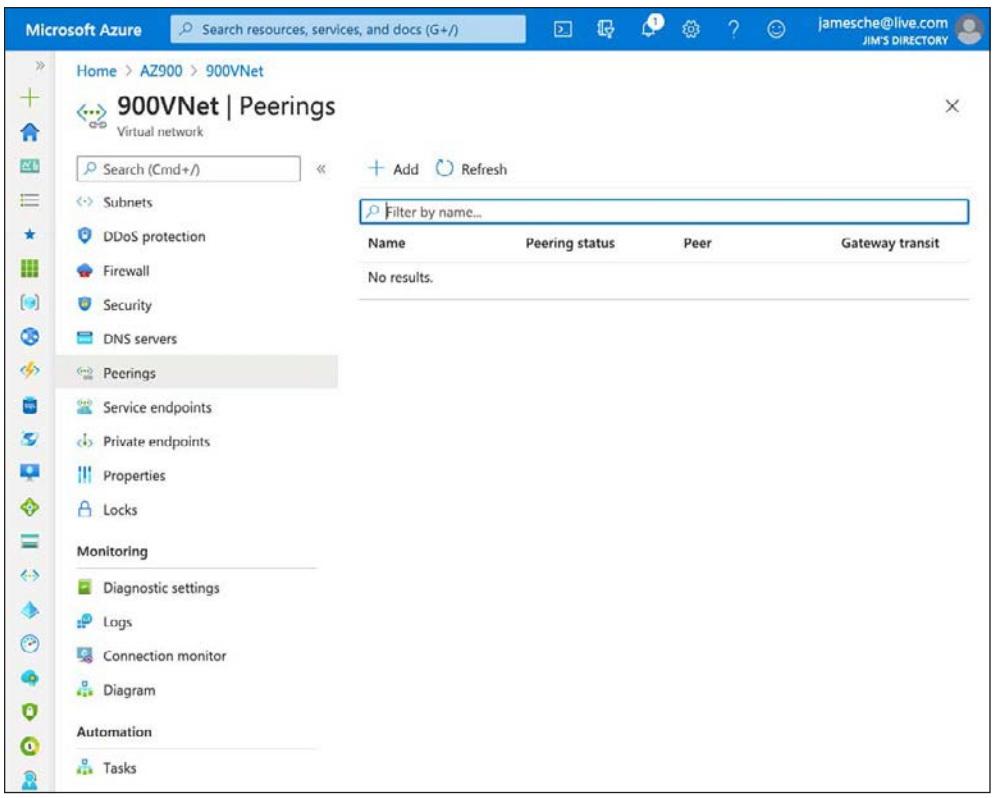


FIGURE 2-25 Adding a virtual network peering

For virtual network peering to work, a peering must be created in both networks. Once you've named your peering and selected the VNet you want to peer, you'll also need to name the peering that will be created in the second network you're peering to, as shown in Figure 2-26.

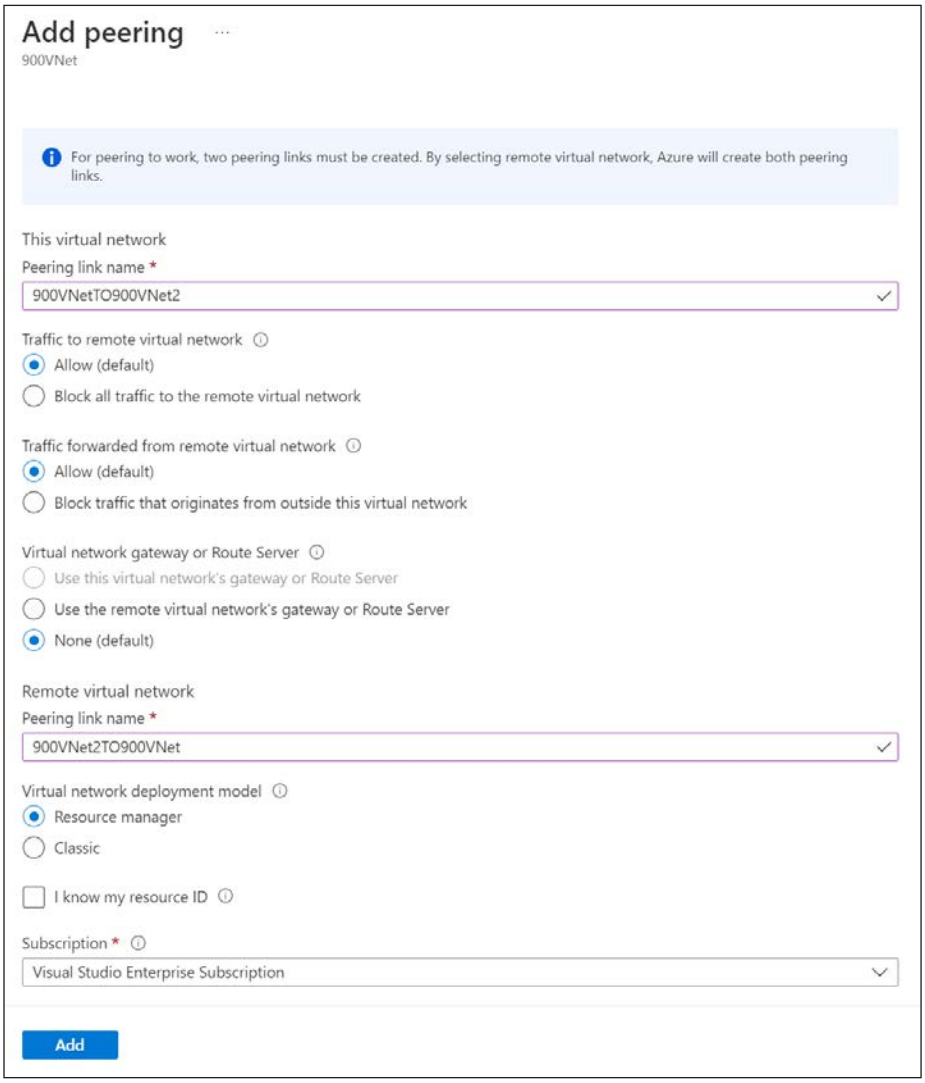


FIGURE 2-26 Configuring a virtual network peering

Once the peering has been added, you can view the status of the peering in the **Peerings** blade in the portal. Figure 2-27 shows the peering created earlier with a status of `Connected`.

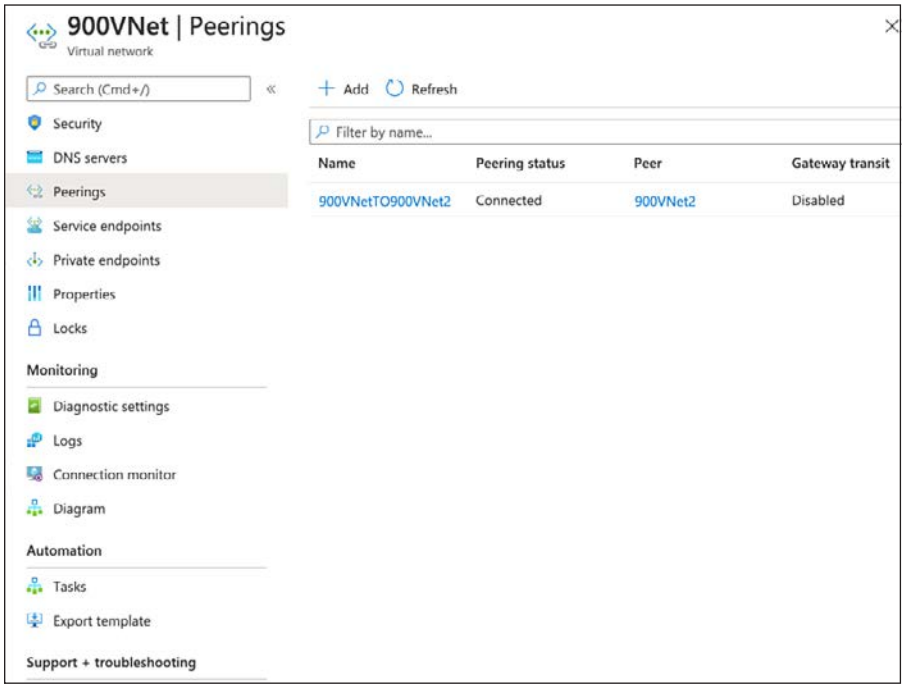


FIGURE 2-27 Virtual network peering status

There is one important limitation related to global virtual network peering. (Remember, global virtual network peering means peering two VNets in different Azure regions.) If you have resources you need to connect to that are behind Azure Load Balancer running in the Basic tier, you won't be able to connect to those resources using the public IP address of the load balancer. If that's a requirement for you, you'll need to change Azure Load Balancer to the Standard tier.

MORE INFO **AZURE LOAD BALANCER AND VIRTUAL NETWORK PEERING**

There are some Azure services that can use the Basic tier of Azure Load Balancer under the hood, so you won't be able to connect to those resources using the load balancer's public IP address when you are using global virtual network peering. You can find more information, including a list of those resource types, by browsing to *http://bit.ly/az900-peeringconstraints*.

Azure DNS

Azure DNS is a service for managing your DNS (Domain Name System) records with the added benefit of being integrated with Azure. To understand what that means, let's briefly discuss DNS in general.

As you may already know, computers on a network communicate with each other using an IP address. IP addresses can be iPv4 addresses such as `192.168.0.4` or iPv6 addresses such as `ff80::d432:9186:3d17:daee%5`. In many scenarios, what's efficient and simple for a computer isn't quite so simple for humans, and IP addresses are a great example of that. Just imagine if you had to know the IP address for every website you wanted to visit. Worse still, an IP address might change, and unless you were aware of such a change, you might not be able to access that website.

The DNS system solves this problem for us by mapping a name to the IP address. A full discussion of DNS is out of scope for this book, but in the most basic sense, when you enter a domain name such as *www.microsoft.com* into your browser, the DNS system is responsible for mapping that name to the IP address needed to reach the Microsoft website.

If you want to purchase your own domain name, you go to a domain registrar, and they register the domain to you. They also give you access to manage your DNS settings. Those settings consist of nameserver settings and DNS records. The nameserver setting is the IP address (typically more than one) of the DNS server that knows the IP address of your domain name. The DNS records are used to inform those nameservers how to reach your domain.

There are numerous DNS records. For example, an A (address) record maps your domain name to the iPv4 address of the computer hosting your domain. An AAAA record does the same for an iPv6 address. A CNAME (canonical name) record maps a name to another name or to an IP address. An MX (mail exchanger) record is used to map to a mail server for the domain.

> ***MORE INFO*** **HOSTNAMES AND SUBDOMAINS**
>
> **An A record (or AAAA record) maps a hostname (e.g., microsoft.com) to an IP address. A CNAME maps a subdomain or domain alias (e.g., *www.microsoft.com*) to a hostname. Without a CNAME record pointing *www.microsoft.com* to the microsoft.com hostname, you would not be able to use *www.microsoft.com* in a browser to reach Microsoft's website.**

It's perfectly fine to manage your DNS records at your domain registrar, but what if you could manage them inside the Azure portal instead? You'd then have access to all the tools Azure provides such as access control to allow others to manage your DNS records, logging to keep an eye on changes, policies to apply governance, and so forth. That's what Azure DNS provides.

To use Azure DNS, you start by creating a DNS zone. There are two types of DNS zones in Azure DNS.

- **Public zone** Zone used for DNS entries that are internet-facing.
- **Private zone** Zone used for DNS entries for use by an Azure virtual network.

EXAM TIP

In any networking discussion, you may see references to public and private endpoints. A public endpoint has an IP address that is accessible over the internet. A private endpoint has an IP address that is only accessible over a private network. It's important to understand that even if a resource has a private endpoint, it may also have a public endpoint. For example, an Azure VM has a private IP address (private endpoint) that isn't accessible from the internet, but it may also have a public IP address (public endpoint) that is accessible from the internet.

A public DNS zone contains entries for a public endpoint, and a private DNS zone contains entries for a private endpoint.

To use a public DNS zone, create a DNS Zone resource in the Azure portal, as shown in Figure 2-28. The name of your DNS zone should be the domain name that you are going to use with the DNS zone.

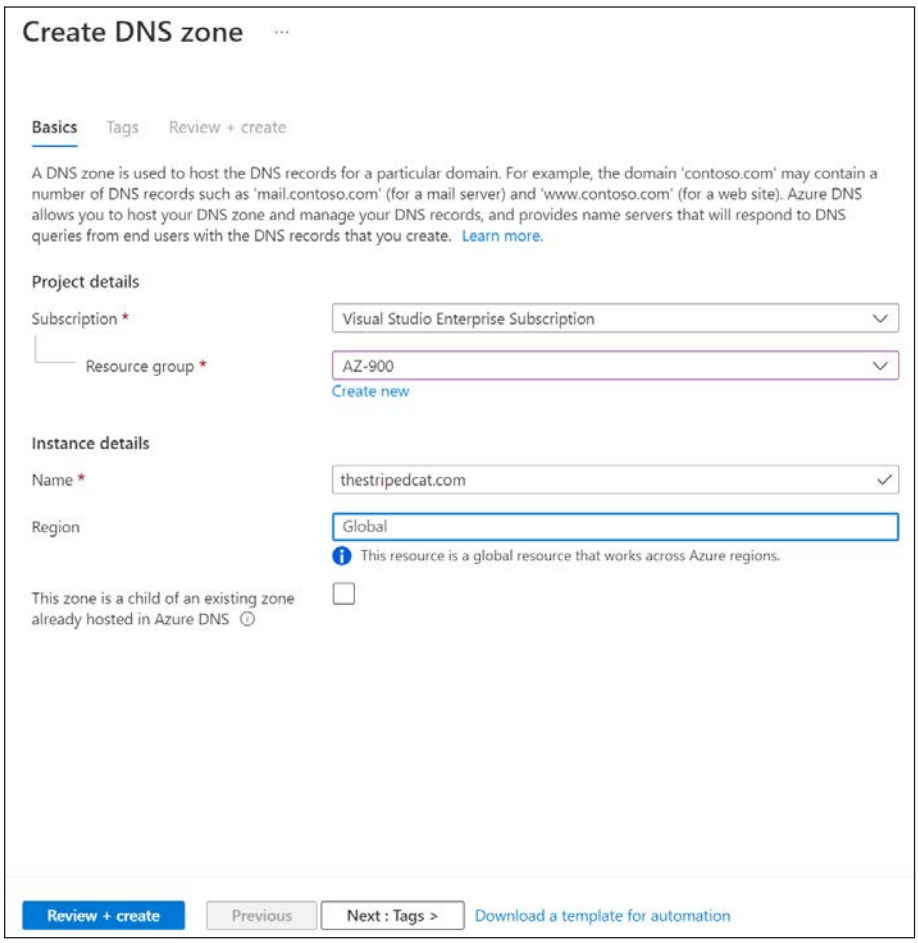


FIGURE 2-28 Creating a public DNS zone

After you've entered the zone name, click **Review+Create** to create the DNS zone.

> ***MORE INFO*** **PRIVATE AND PUBLIC ZONE CREATION**
>
> **You might have noticed that you didn't have to specify whether your new DNS zone is a public or a private zone. That's because a DNS Zone resource is always a public DNS zone. To create a private DNS zone, you use the Private DNS Zone resource, as you'll see later in this section.**

Figure 2-29 shows my *thestripedcat.com* DNS zone. Notice that there are four nameservers at the top of the screen. To use this DNS zone, I'll need to update the nameservers at my domain registrar with the nameservers shown in the portal.

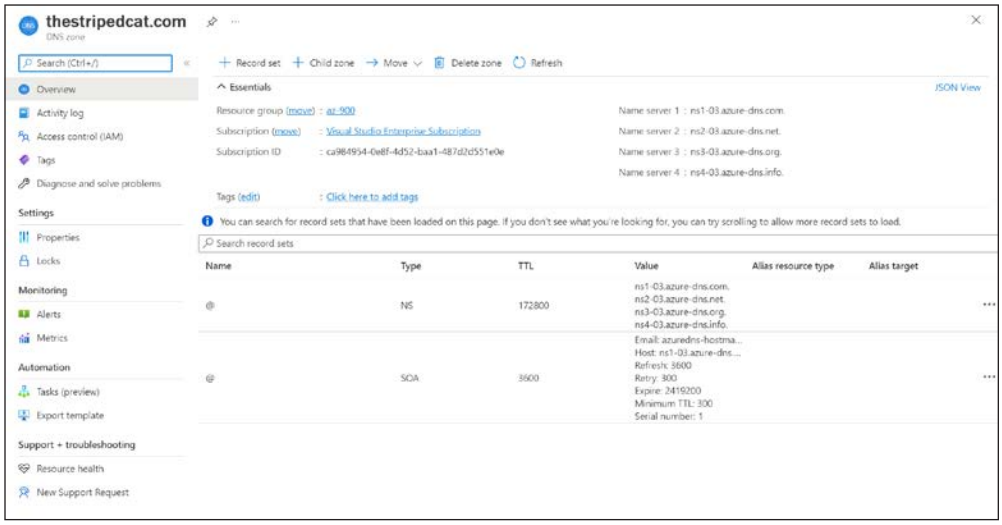


FIGURE 2-29 A DNS zone in the Azure portal

You can see two DNS records already in my zone. The NS (nameserver) record defines the authoritative nameservers for my domain. The SOA (start of authority) record contains administrative information about my domain. Both records are required, so Azure creates them for you automatically.

Suppose I am going to host my website for thestripedcat.com in Azure App Service, and I've already created a web app at stripedcat.azurewebsites.net. To use thestripedcat.com as my domain, I need to create two DNS records. I need an A record that points to the IP address of the server in App Service running my web app. I'll also need a CNAME record that enables me to use *www.thestripedcat.com*.

MORE INFO **CUSTOM DOMAINS IN APP SERVICE**

There are a few other steps I need to take in App Service to use a custom domain, but I'm not going to cover those here. This discussion is limited to configuring the records in a DNS zone.

To add those records, click the **+ Record Set** button shown previously in Figure 2-29. This opens the **Add Record Set** blade. In Figure 2-30, I am creating the A record that points to the IP address of my web app. Notice that I've used `@` for the name. The `@` symbol in DNS is a way to refer to the root domain—in my case, *thestripedcat.com*. Once I've configured my new A record, I click the **OK** button to create it.

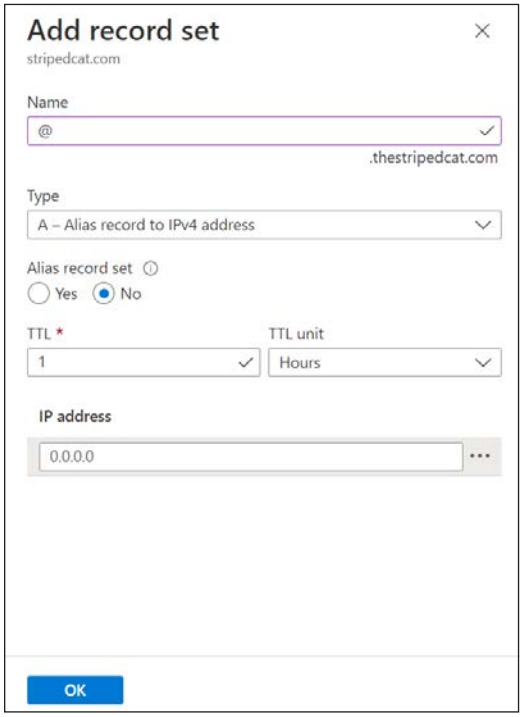


FIGURE 2-30 Creating an A record in my DNS zone

Creating the CNAME record is similar, as shown in Figure 2-31. Notice that the name of the record is `www`, and the alias is the URL in App Service where my website is hosted. Once the CNAME record has been created and that change has propagated, I can enter *www.thestripedcat.com*, and it will take me to the web app running in App Service.

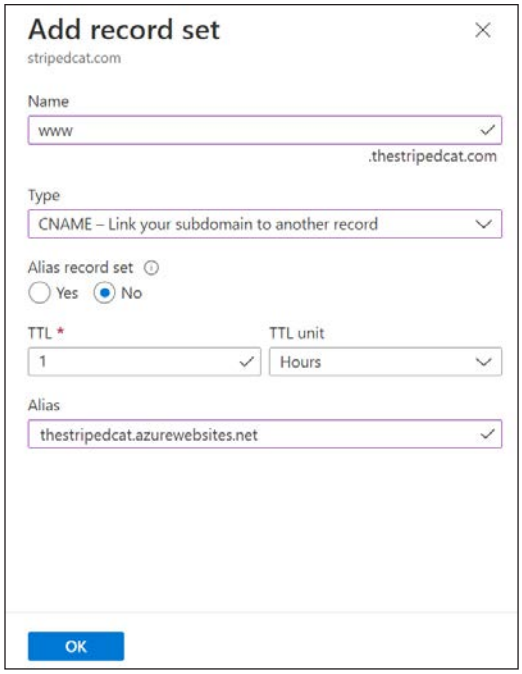


FIGURE 2-31 Creating a CNAME record in my DNS zone

Creating a private DNS zone is the same as creating a public zone except that the resource type is Private DNS Zone. Once you create your private DNS zone, you need to link your virtual network to it. To do that, you use the **Virtual Network Links** menu option, as shown in Figure 2-32. You then click the **Add** button to create a new link. In Figure 2-32, I've created a link to my 900VNet virtual network.

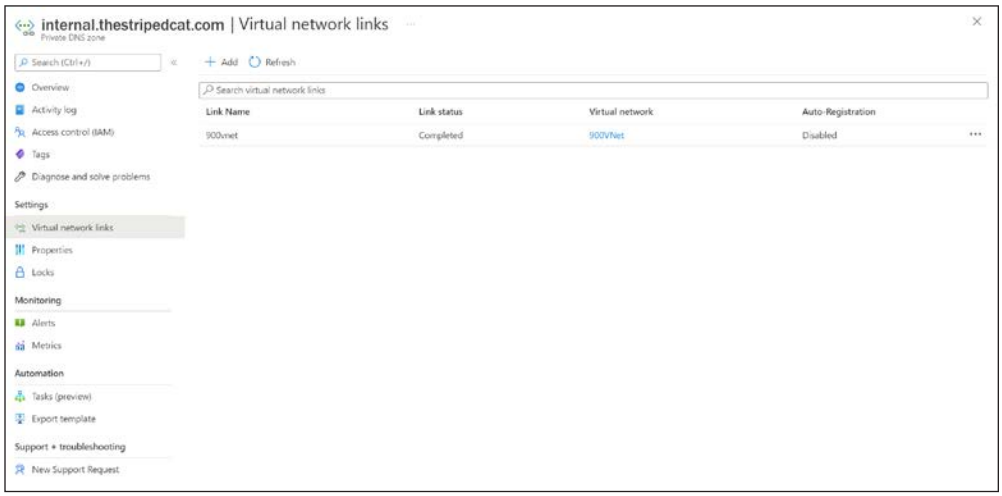


FIGURE 2-32 Linking a VNet to a private DNS zone

Once you've created a private DNS zone and linked your VNets, you can create DNS records for your Azure resources. For example, you can create an A record that points to the IP address of a VM that's in your VNet.

Azure VPN Gateway

In most cases, you'll want to connect your VNet to other networks. You might need to connect your VNet to another VNet in Azure, or you might also want to connect your VNet to on-premises resources. In both cases, network traffic is going to travel over the internet, and that incurs a certain amount of risk. To provide more security for your network traffic, you can use a secure virtual private network (VPN) connection.

A VPN is a network technology that allows connectivity between a private network and a public network. If you ever connect to resources on your company's private network from your home computer, you're likely using a VPN connection that allows your home network to connect to your company's network.

EXAM TIP

You may see VPN Gateway referred to as a virtual network gateway. Keep in mind that when you see the term "virtual network gateway," it's the same thing as VPN Gateway.

Azure VPN Gateway uses VPN connections to enable secure connectivity between an Azure VNet and other networks. VPN Gateway secures these connections using internet protocol security (IPSec) and the internet key exchange (IKE) protocol. This ensures that traffic flowing over the public internet is encrypted and secure.

A VPN Gateway uses two or more VMs that are created inside a subnet (called the *gateway subnet*) that is created explicitly for the VPN Gateway. These VMs run services that implement the functionality of VPN Gateway, and they also have network routing tables that enable them to properly route network traffic. You can't use these VMs for anything other than VPN Gateway, and you also can't remote into them or change their configurations.

Before you can create a VPN Gateway, you create the gateway subnet and specify the IP address range for that subnet. Azure makes this easy for you in the Azure portal. Once you open your VNet in the portal, click **Subnets** in the menu and then click the **+ Gateway Subnet** button, as shown in Figure 2-33.

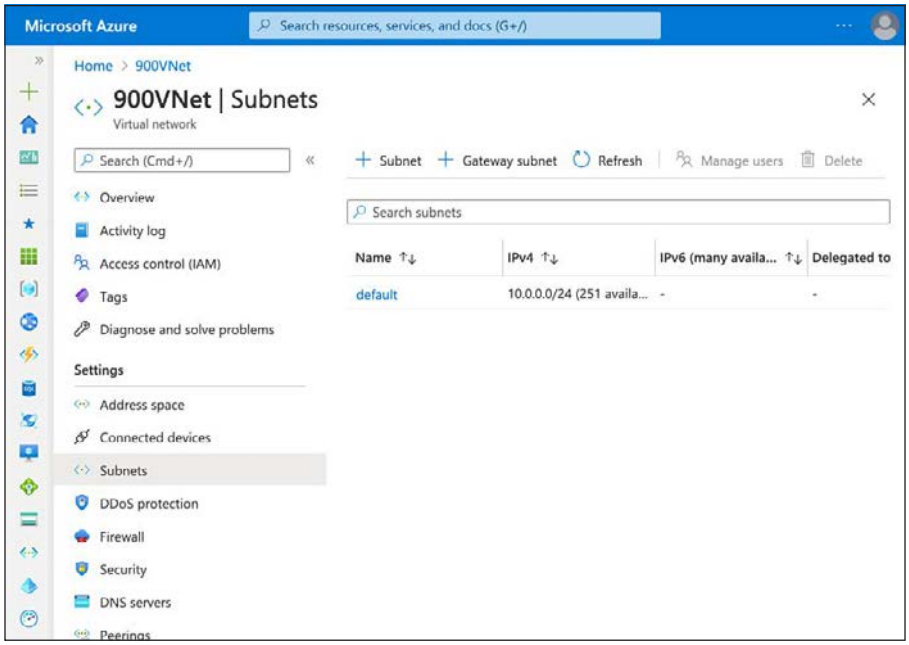


FIGURE 2-33 Creating a gateway subnet

> ***MORE INFO*** **GATEWAY SUBNETS**
>
> **Microsoft provides more information on the gateway subnet, including some guidance on choosing an IP address range for your subnet. You can find that information by browsing to *http://bit.ly/az900-gatewaysubnets*.**

Once you've created a subnet for VPN Gateway, you can create the VPN Gateway. To create a VPN Gateway, you create a new virtual network gateway and specify the gateway type as VPN, as shown in Figure 2-34. Notice that the portal has automatically selected the gateway subnet that exists in the selected VNet.

> ***MORE INFO*** **CREATING VPN GATEWAYS**
>
> **It can take a long time to create a VPN Gateway. Azure must create VMs in the gateway subnet to support the gateway, and it also must configure VPN Gateway services and network routing tables. It can take up to 45 minutes before your VPN Gateway is ready for use.**

After you create your VPN Gateway, you can configure a connection to it. There are three connection types supported by VPN Gateway: VNet-to-VNet, site-to-site, and point-to-site.

A VNet-to-VNet connection allows you to connect two Azure VNets to each other. Each VNet must have a VPN Gateway configured, and the VNets don't have to be in the same Azure region or even in the same Azure subscription. Communication between VNets that are using

a VNet-to-VNet connection is encrypted and travels over Microsoft's backbone infrastructure, not over the internet.

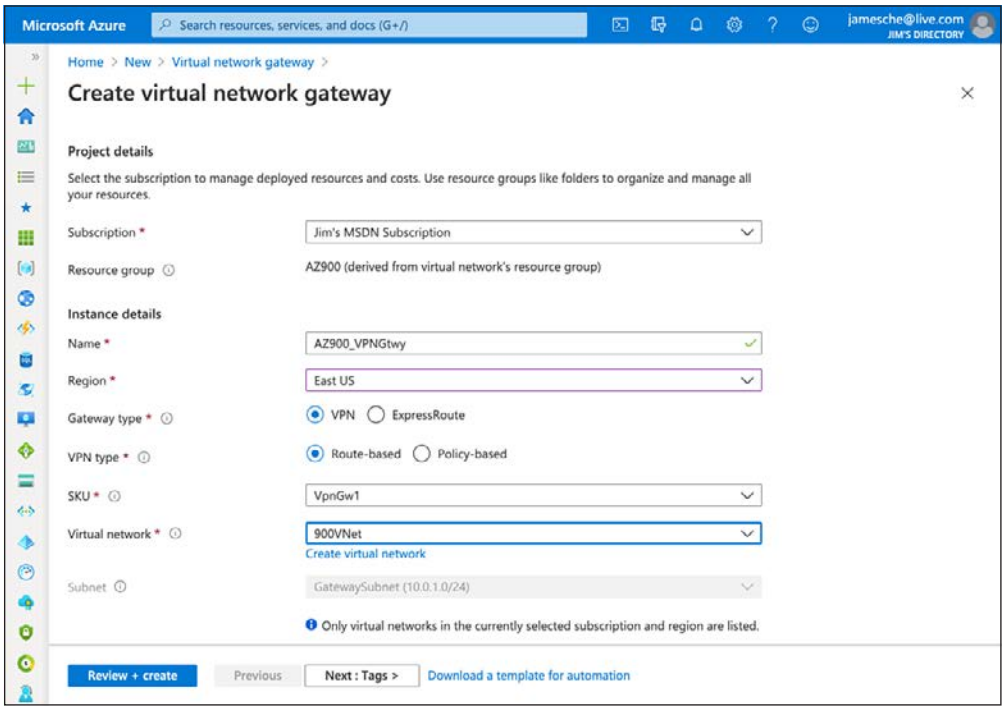


FIGURE 2-34 Creating a VPN Gateway

EXAM TIP

VPN Gateway has several pricing tiers, and each pricing tier has an associated bandwidth cap. When connecting two VNets using a VNet-to-VNet connection, make sure you can live with the bandwidth restrictions imposed by the VPN Gateway pricing tier you are using. If you need to avoid a bandwidth restriction, using VNet peering (covered in the next section) might be a better option for you.

A site-to-site connection allows you to connect your VNet to an on-premises network using an encrypted VPN connection. Site-to-site connections require you to configure a VPN device on your on-premises network, and that VPN device must have a public-facing IP address. Network traffic between your VNet and your on-premises network travels over an encrypted VPN connection.

MORE INFO **VPN DEVICES**

For more information on VPN devices, you can use with a site-to-site connection, browse to *https://bit.ly/az900-vpndevices*.

A point-to-site connection connects your VNet to a single device. That device can be a computer, but it can also be a mobile device, such as a tablet or a smartphone. When using a point-to-site connection, software on the device establishes a VPN connection to VPN Gateway, and all traffic is encrypted and travels over that connection.

> ***MORE INFO*** **SITE-TO-SITE AUTHENTICATION**
>
> **Site-to-site connections can authenticate using Azure certificate authentication, Remote Authentication Dial-In User Service (RADIUS) authentication, Azure Active Directory authentication, or OpenVPN.**

Azure ExpressRoute

As you've just learned, you can use Azure VPN Gateway to connect your Azure VNet to on-premises resources, and many customers use this method. However, there are some aspects of using a VPN that might not meet the requirements of some customers. For example, a VPN is limited to a maximum of 1.25 Gbps in network speed. If a customer needs more speed than that, VPN isn't a good choice. VPN Gateway also sends all traffic over the public internet, and that might not be a viable option for some people.

For these reasons, Azure offers a service called ExpressRoute that can offer speeds up to 10 Gbps over dedicated fiber-optic connections. When you use ExpressRoute, you connect from your on-premises network to a Microsoft Enterprise Edge router (MSEE), and that MSEE router then connects you to Azure. The MSEE router sits on the edge of Microsoft's network, and in most cases, your connection will also be from a router in your on-premises network that is on the edge of your network.

> ***MORE INFO*** **EDGE NETWORK DEVICES**
>
> **An edge device on a network refers to a device that operates as the access point into the network. If you think of a network as a circle and devices that are on that network as being inside that circle, you can think of an edge device as sitting on the line that makes up the circle.**

In most situations, customers connect to the MSEE router using a third-party service provider. The service provider is a major network service provider, often an internet service provider. The service provider has network connections directly into the MSEE router, and those connections have dedicated bandwidth. Figure 2-35 shows a typical ExpressRoute configuration.

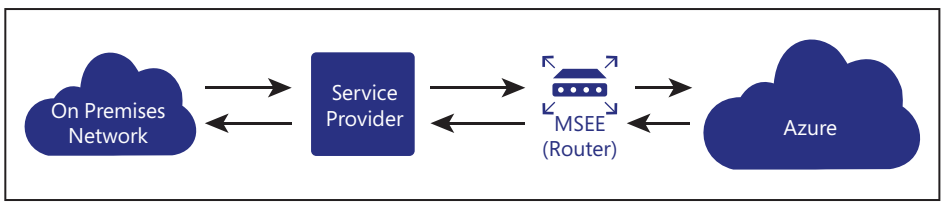


FIGURE 2-35 A typical ExpressRoute configuration

EXAM TIP

Microsoft calls an ExpressRoute connection a *circuit*.

Because data in ExpressRoute doesn't traverse the public internet, bandwidth is much more reliable. However, the ExpressRoute configuration you see in Figure 2-35 does require that you trust the service provider with the data flowing through the circuit. If you want to remove the service provider from the picture, you can use an offering called ExpressRoute Direct that allows you to connect directly to a physical port on the MSEE router. ExpressRoute Direct also provides for much higher bandwidth, if that's a concern for you.

MORE INFO **EXPRESSROUTE AND AVAILABILITY**

ExpressRoute is designed for high availability. Each circuit uses redundant connections, so if you create a circuit at 5 Gbps, the actual bandwidth allocated to that circuit is 10 Gbps. Microsoft will even allow you to use that extra bandwidth for short bursts.

Skill 2.3: Describe Azure Storage services

Most deployments in Azure need some type of storage. If you want persistent data stored on a VM, you'll need a disk to store the data on. However, disks are just one example of storage in Azure. Azure offers storage services for many different scenarios, and there are also plenty of options available for getting data into and out of Azure Storage.

This section covers:

- Azure Storage services
- Storage tiers
- Redundancy options
- Storage accounts and storage account types
- Moving files to and from Azure Storage
- Migrating files to Azure Storage

Azure Storage services

Just about every application needs to store and access data, and the format and makeup of that data varies widely. No matter what kind of data an application uses, you need access to that data to be fast and reliable. That means the infrastructure that stores your data must be scalable and highly available. It must have all the benefits we've already discussed related to the cloud.

Azure offers numerous storage solutions, all of which meet the needs of today's demanding cloud applications. Let's discuss the various storage services available to you in Azure.

Azure Blob storage

Azure Blob storage is designed for storing data with no defined structure. That includes text files, images, videos, documents, and much more. An entity stored in Blob storage is referred to as a *Blob*. There are three types of Blobs.

- **Block Blobs** These are used to store files used by an application.
- **Append Blobs** These are like Block Blobs, but Append Blobs are specialized for append operations. For that reason, they are often used to store constantly updated data like diagnostic logs.
- **Page Blobs** These are used to store virtual hard disk (.vhd) files that are used in Azure virtual machines.

Blobs are stored in storage *containers*. A container is used as a means of organizing Blobs, so you might have a container for video files, another container for image files, and so on. The choice, however, is entirely up to you.

Data that you store in Blob storage is stored in a specific storage tier. The requirements and cost of storage differ depending on the tier. We'll discuss this in more detail in the "Storage tiers" section later in this chapter.

Blob storage is a great option when you need an application to have easy and reliable access to binary data. For example, you might use Blob storage to store video files that your application needs to stream, or you might use it to store large log files used in your application.

Azure Disks

Azure Disks refers to disks that are used in virtual machines. Azure creates a disk that is automatically designated for temporary storage when you create a VM. This means data on that disk will be lost if there's a maintenance event on the VM. If you need to store data for a longer period that will persist between VM deployments and maintenance events, you can create a data disk that is stored in Azure Disks.

Azure Disks are available as both traditional hard disks (HDD) and solid-state drives (SSD). HDD disks are cheaper and designed for noncritical data. SSD disks are available in a Standard tier for light use and as Azure Premium Disk for heavy use.

When you create a new disk, you can choose to create it from a snapshot of another disk, a Blob in Blob storage, or you can create an empty disk. A snapshot is a full copy of another disk. You can create a snapshot of a disk by creating a new Snapshot resource in the Azure portal and selecting the disk you want a snapshot of.

EXAM TIP

You can create a snapshot of a VM's OS disk or a data disk. If you create a snapshot of a VM's OS disk, you can use that snapshot to create another VM or to simply create a backup of the VM at a specific point in time.

When you create a data disk for a VM, it is assigned a drive letter of your choosing. Each data disk has a maximum size of 32,767 gibibytes, and the maximum number of data disks you can attach to a VM depends on your VM SKU.

MORE INFO **GIBIBYTES**

In case you're wondering, the use of the term *gibibyte* is not a typo. A gibibyte is 1,073,741,824 bytes. In comparison, a gigabyte is 1,000,000,000 bytes.

Azure Files

Azure Disks are a good option for adding a disk to a virtual machine, but if you just need disk space in the cloud, it doesn't make sense to take on the burden of managing a virtual machine and its operating system. In those situations, Azure Files is the perfect solution.

NOTE **AZURE FILES AND AZURE STORAGE**

Azure Files shares are backed by Azure Storage, so you will need a storage account to create an Azure Files share.

Azure Files is a completely managed file share that you can mount just like any SMB file share. That means existing applications that use network attached storage (NAS) devices or SMB file shares can use Azure Files without any special tooling, and if you have multiple applications that need to access the same share, that will work with Azure Files, too.

EXAM TIP

You can mount Azure Files shares on Azure VMs and on-premises on Windows, Linux, and macOS. You can't, however, use Windows 7 or Windows Server 2008 to mount an Azure Files share on-premises because those operating systems only support SMB 2.1.

Also, because Azure Files shares use SMB, you'll need to make sure that TCP port 445 is open on your network. On Windows, you can use the `Test-NetConnection` PowerShell cmdlet to test connectivity over port 445. For more information, see *https://bit.ly/az900-azurefiles*.

Azure Queues

Azure Queues (also called Azure Queue Storage) is a storage service designed for queueing many messages. For example, suppose you have an application that processes video files that users upload. Each time a video file is uploaded, your application encodes the video into a format that is optimized for streaming across the internet. If many people are uploading videos, you can use a message queue to store the path to all video files that require encoding.

Messages in Azure Queues are accessible using a URL, and requests to the queue are authenticated. The maximum size of a single message is 64KB, and the maximum size of a message queue is 500 tebibytes (approximately 550 terabytes.)

MORE INFO **INTERACTING WITH AZURE QUEUES**

Message queues aren't created and managed using the Azure portal. Instead, queues and messages are typically managed programmatically within an application. Microsoft offers support for most programming languages, including .NET, Java, Python, Node.js, C++, PHP, and Ruby.

You can also use The Az PowerShell module to create and manage queues and messages.

Each queue is given a lowercase name, and the URL for accessing the queue uses that name. For example, if we use a queue name of `videos-to-encode` and our storage account name is `az900storage`, the URL to the message queue would be *https://az900storage.queue.core.windows.net/videos-to-encode*.

EXAM TIP

A message has a default time-to-live of seven days. However, when you add a message to a queue, you can specify a time-to-live of your choosing. Once the time-to-live expires, the message is deleted.

Storage tiers

Microsoft offers numerous storage tiers for Blob Storage that are priced according to how often the data is accessed, how long you intend to store the data, and so on. The Hot storage tier is for data you need to access often. It has the highest cost of storage, but the cost of accessing the data is low. The Cool storage tier is for data that you intend to store for a longer period and not access quite as often. It has a lower storage cost than the Hot tier, but the access costs are higher. You're also required to keep data in storage for at least 30 days.

Microsoft also offers an Archive storage tier for long-term data storage. Data stored in the Archive tier enjoys the lowest storage costs available, but the access costs are the highest. You must keep data in storage for a minimum of 180 days in the Archive tier or you can be subjected to an early deletion charge. Because data in the Archive tier isn't designed for quick and frequent access, it can take a very long time to retrieve it. In fact, while the Hot and Cool access tiers guarantee access to the first byte of data within milliseconds, the Archive tier only guarantees access to the first byte within 15 hours unless you use the high priority option.

EXAM TIP

If you need to access a Blob in the Archive tier, you must first move it into the Hot or Cool tier. The process of moving the Blob to the Hot or Cool tier is called *rehydration*.

Redundancy options

We discussed high availability, fault tolerance, and disaster recovery in Chapter 1, and those concepts certainly apply to Azure Storage services. Azure provides several options for data redundancy to ensure your data is always available when you need it.

The underlying resource for all storage services in Azure is the storage account. When you create an Azure storage account, you can specify the redundancy option you want for any data in that storage account.

> ***MORE INFO*** **STORAGE ACCOUNTS**
>
> **You'll learn more about storage accounts in "Storage accounts and storage types" in the next section of this chapter.**

Figure 2-36 shows a storage account being created in the Azure portal. The Redundancy dropdown has been clicked, and the four options for redundancy are displayed.

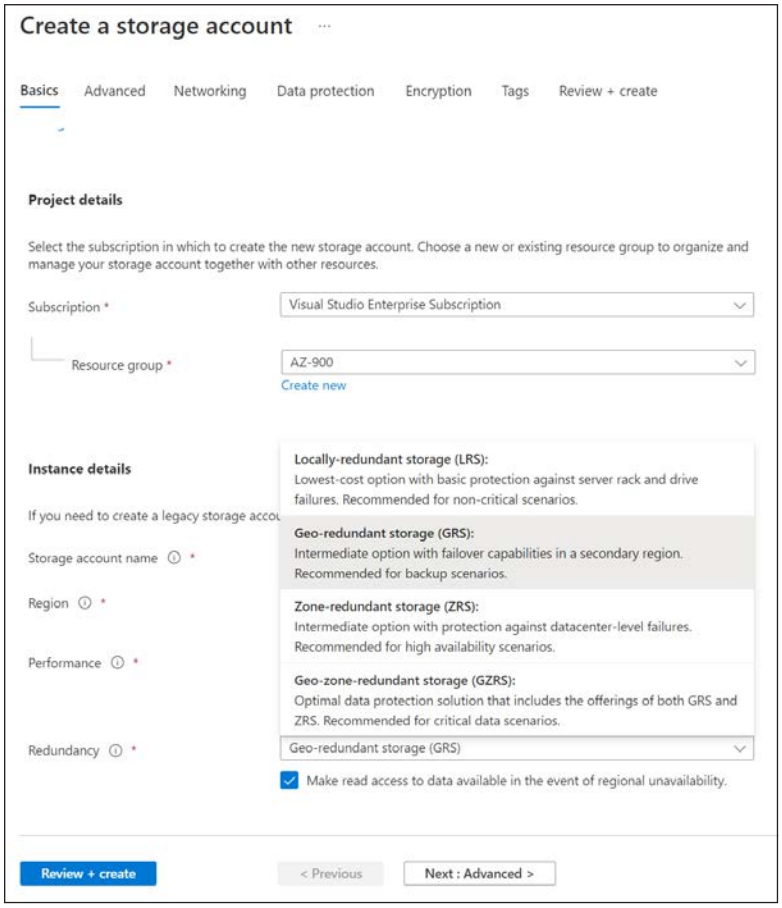


FIGURE 2-36 Storage account redundancy options

Data in Azure Storage can be replicated in the primary region (the region you select when you create the storage account) or across regions depending on which redundancy option you choose.

EXAM TIP

You cannot change the redundancy type of an Azure storage account after the storage account has been created.

Primary-region redundancy

There are two redundancy options designed to replicate your data only in the primary region. They are locally redundant storage (LRS) and zone-redundant storage (ZRS). When using LRS, Azure makes three copies of your data within a single datacenter. ZRS also makes three copies of your data, but each copy is in a separate availability zone in a separate datacenter.

MORE INFO **AVAILABILITY ZONES**

Not all Azure regions include support for availability zones. Those that don't will also not have support for ZRS. For more information on availability zones, see "Availability zones" in Skill 2.1, "Describe the core architectural components of Azure."

LRS is the least expensive redundancy option, but it's also the least durable. Microsoft recommends you use LRS only if you can easily recreate any data that is lost. This is because LRS only protects your data from a problem with a drive or a server rack where your data resides.

When you use ZRS, each copy of your data is in a separate datacenter because Azure uses availability zones for redundancy. ZRS has the added benefit of protecting your data from a problem impacting a single datacenter, but it won't protect you from a large-scale disaster in an Azure region.

LRS or ZRS are frequently chosen by companies that are required to keep all data within the boundaries of a country. When you move to data replicated across multiple regions, you may end up in a scenario where your data crosses geographical boundaries, and if regulations forbid that, LRS or ZRS may be your only viable option.

MORE INFO **WRITES ARE SYNCHRONOUS**

Data writes to Azure Storage in LRS and ZRS are synchronous. That means that a write to Azure Storage is only successful once the data is successfully written to all three copies.

Multiple-region redundancy

Azure can also distribute copies of your data across Azure regions. These options protect you from a large-scale impact in a particular Azure region. Just as with primary-region redundancy, there are two redundancy options for multiple-region redundancy. They are geo-redundant storage (GRS) and geo-zone-redundant storage (GZRS).

GRS creates three copies of your data in the primary region using LRS. It then creates an additional three copies of your data using LRS in a separate Azure region that's hundreds of miles away from the primary region.

GZRS creates three copies of your data in three availability zones in the primary region using ZRS. It then creates three copies of your data using LRS in a secondary region that is hundreds of miles away from the primary region.

EXAM TIP

When using GRS or GZRS, data in the primary region is replicated synchronously, but data is replicated to the secondary region asynchronously. Therefore, there is a small delay in synchronizing the data between the primary and secondary regions.

You might wonder why GZRS uses LRS for the redundant data in the secondary region. The reason is that the replicated data in the secondary region is there for backup purposes only by default. Access to the data in the secondary region is available only if there is a failure in the primary region. In that case, Azure will change the DNS information to your storage account so that any access requests are sent to the secondary region.

If you require it, you can enable read access to the data in the secondary region by checking the "Make read access to data available in the event of regional unavailability" checkbox shown previously in Figure 2-36. When using this option, the two multiple-region redundancy options are referred to as read access geo-redundant storage (RA-GRS) and read access geo-zone-redundant storage (RA-GZRS). This option is available for all storage services except Azure Files.

MORE INFO **USING READ ACCESS REDUNDANCY OPTIONS**

Earlier in this chapter, I said Azure will update the DNS records for your storage account in a failover situation where access to data is required in the secondary region. If you use one of the read access options, Azure creates a URL for accessing the secondary region by appending `-secondary` to the storage account name in the URL.

Storage accounts and storage types

As I said earlier, a storage account is the underlying Azure resource for all storage services. Azure offers several different storage account types, and the type of storage account you choose affects the services available to you as well as your costs.

By default, a storage account will use the Standard tier general-purpose v2 account type. This account type is recommended for most uses of Azure Storage, and it supports all the Azure Storage services.

If you require the highest level of performance, you can choose one of the three types offered in the Premium tier. As shown in Figure 2-37, they are Block Blobs, file shares, and Page Blobs.

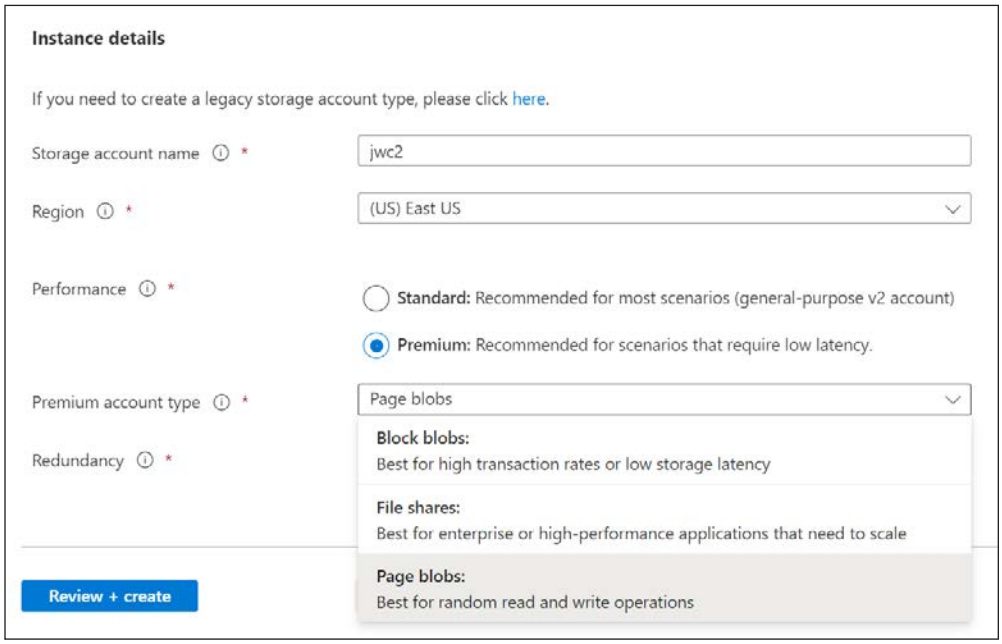


FIGURE 2-37 Premium storage account types in the Azure portal

Block Blobs are used for Block Blobs and Append Blobs in Blob storage. This account type offers maximum performance when you need to handle many transactions against Azure Storage. File shares is a premium offering targeting Azure Files, and it's a requirement if you need a file share that supports network file system (NFS) file shares. Page Blobs supports only Page Blobs for high performance. Premium storage account types use solid state drives for enhanced performance.

EXAM TIP

You cannot change the account type after a storage account has been created.

Moving files to and from Azure Storage

There are numerous methods to move files to and from Azure Storage. Perhaps the most used method is to use the AzCopy utility, a command-line tool for copying Blobs or files to and from Azure Storage.

> ***MORE INFO*** **AzCOPY**
>
> **For more information on AzCopy, including a download link for your OS, see *https://bit.ly/az900-azcopy*.**
>
> **AzCopy requires authentication using either an SAS token or Azure AD. For more information on authentication and AzCopy, see *https://bit.ly/az900-azcopyauth*.**

You can use AzCopy to copy individual files into Blob storage, or you can use it to upload an entire directory. To copy a single file, you use the `Copy` command. The following sample command copies a file named `MyDocument.docx` from my local hard drive to the Docs container in Blob storage.

```
azcopy copy "c:/Documents/MyDocument.docx" 'https://jwcstorage.blob.core.windows.net/
docs/MyDocument.docs'
```

> ***NOTE*** **USE FORWARD SLASHES**
>
> **Notice that I am using forward slashes (/) and not back slashes (\) in the local path. I am using AzCopy version 10.15.0, and it fails with an error unless I use forward slashes.**

You can also use AzCopy to copy an entire directory of files to Azure Storage. The following command will copy all the files in the Documents directory to Azure Storage. It will also create a Documents folder in the `docs` storage container.

```
azcopy copy "c:/Documents" "https://jwcstorage.Blob.core.windows.net/docs" --recursive
```

> ***NOTE*** **COMMAND PROMPT AND POWERSHELL**
>
> **The commands I'm showing are for use at the Windows command prompt. If you are using PowerShell, you should use single quotes instead of double quotes.**

If you want to download a file from Blob storage to your local machine, reverse the paths used in the `AzCopy` command. The following command copies a file from Azure Storage to my local drive.

```
azcopy copy "https://jwcstorage.blob.core.windows.net/docs/MyDocument.docx" "c:/
Documents/MyDocument.docx"
```

The Copy command can also be used to copy files and directories to and from Azure Files. The following command copies a file from my local file system to the `docs` share in Azure Files.

```
azcopy copy "c:/Documents/MyDoc.docx" "https://jwcstorage.file.core.windows.net/docs/
MyDoc.docx"
```

AzCopy is a powerful tool for managing resources in Azure Storage, especially in situations where you need to script the management of many resources. However, if you're looking for a simpler option, Azure Storage Explorer may be preferable. Azure Storage Explorer is a free application that makes it easy to manage storage resources.

> ***NOTE*** **STORAGE EXPLORER**
>
> **Storage Explorer is a multi-platform application available for Windows, macOS, and Linux. You can find out more and download Storage Explorer at *https://bit.ly/az900-storageexplorer*.**

Once you've installed Storage Explorer and added your Azure subscription, you easily interact with your storage resources. Figure 2-38 shows a Blob container in Storage Explorer, and one Blob is visible in the pane on the right.

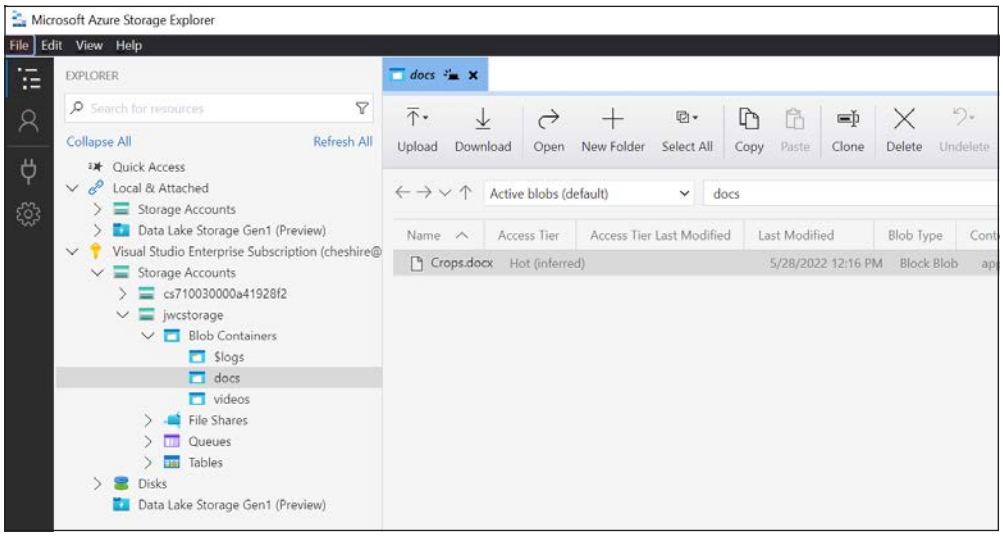


FIGURE 2-38 A Blob in Azure Storage Explorer

Storage Explorer might be simple in appearance, but it has some powerful features, including the ability to easily generate SAS tokens, create and manage containers, change the storage tier for Blobs, and much more. Figure 2-39 shows the context menu when right-clicking a Blob container.

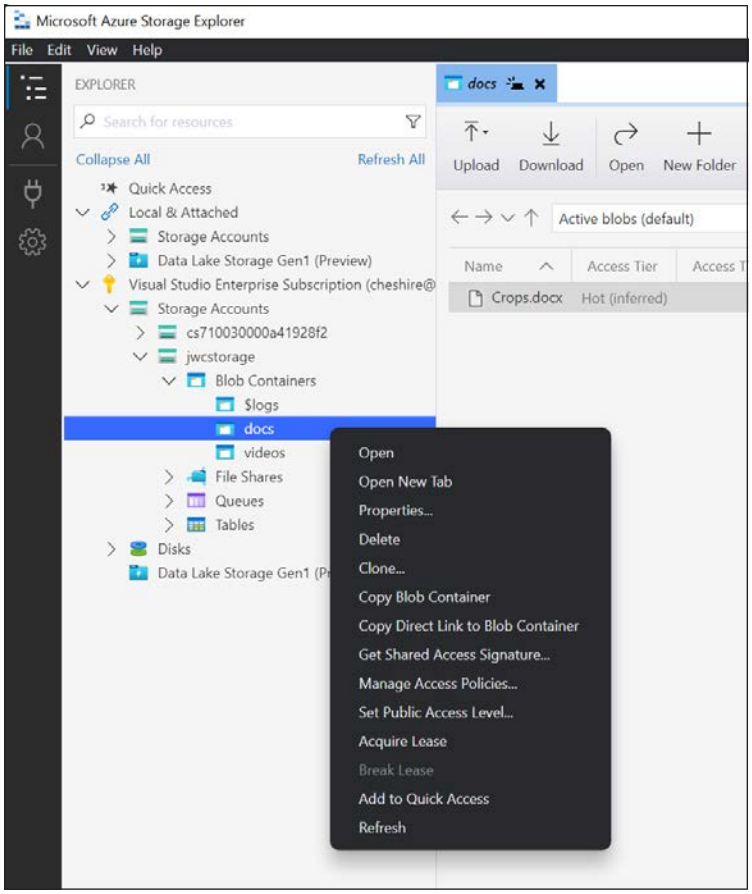


FIGURE 2-39 Context menu for a Blob in Storage Explorer

To upload a file to storage, you can click the **Upload** button shown in Figure 2-39 and browse to the file to upload. However, you can also drag and drop files and folders directly into Storage Explorer for an easier copy experience. The same technique works whether you are copying files to Blob storage or a share in Azure Files.

Speaking of Azure Files, one possible problem you may encounter is the remote location of files. If your users or applications are using a file share mapped to Azure Files, they might experience longer-than-usual file transfer times because the files are in Azure. To solve that problem, Microsoft introduced Azure File Sync.

Install Azure File Sync on one or more servers in your local network, and it will keep your files in Azure Files synchronized with your on-premises server. When users or applications need to access those files, they can access the local copy quickly. Any changes you make to the centralized Azure Files share are synchronized to servers running Azure File Sync.

Migrating to Azure

So far, we've looked at scenarios involving copying relatively small amounts of data to Azure, but there are plenty of scenarios where you may need to migrate a large amount of data. You may also want to migrate a server or VM to Azure, or you may need to move one or more databases into the cloud.

For these kinds of migrations, Azure offers a service called Azure Migrate. Azure Migrate can migrate your servers, databases, web apps, virtual desktop infrastructure (VDI), and more to Azure. You can migrate from on-premises, or you can migrate from another cloud provider.

To migrate servers, databases, and web apps you perform a three-step process:

1. **Discover** Locate the servers, databases, and web apps in your environment.
2. **Assess** Determine the requirements for migration and the necessary Azure resources required.
3. **Migrate** Perform the migration to Azure.

To begin a migration, you must first create an Azure Migrate project. A project is a simple Azure resource that stores all the metadata from your discovery, assessment, and migration processes. When you create a project, you specify the resource group, project name, and geography where you'd like the project to be created. Figure 2-40 shows an Azure Migrate project being created in the United States geography.

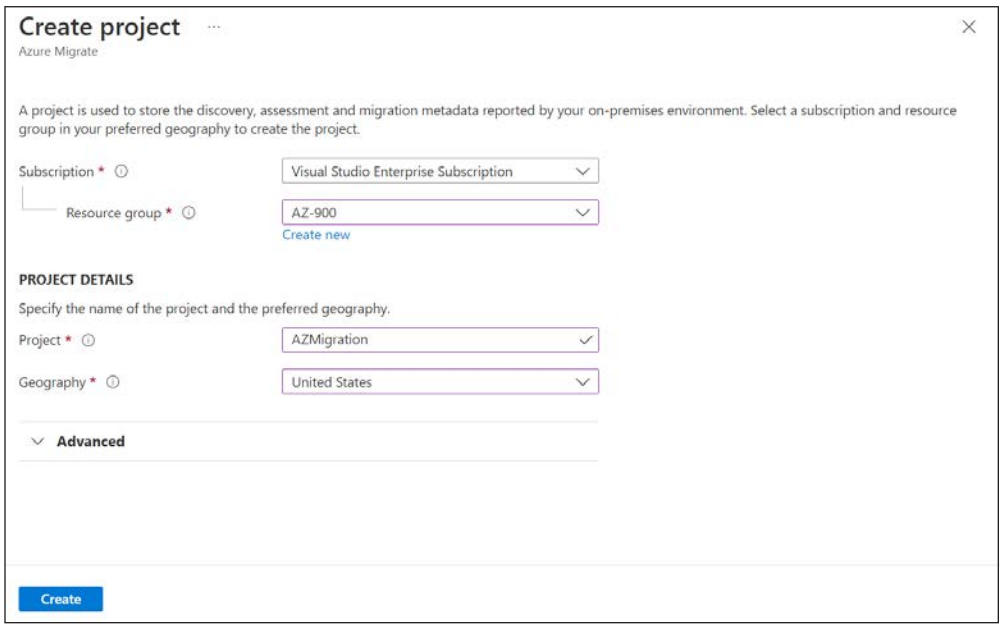


FIGURE 2-40 Creating an Azure Migrate project

Once a project has been created, you can begin the discovery phase by clicking the **Discover** button shown in Figure 2-41.

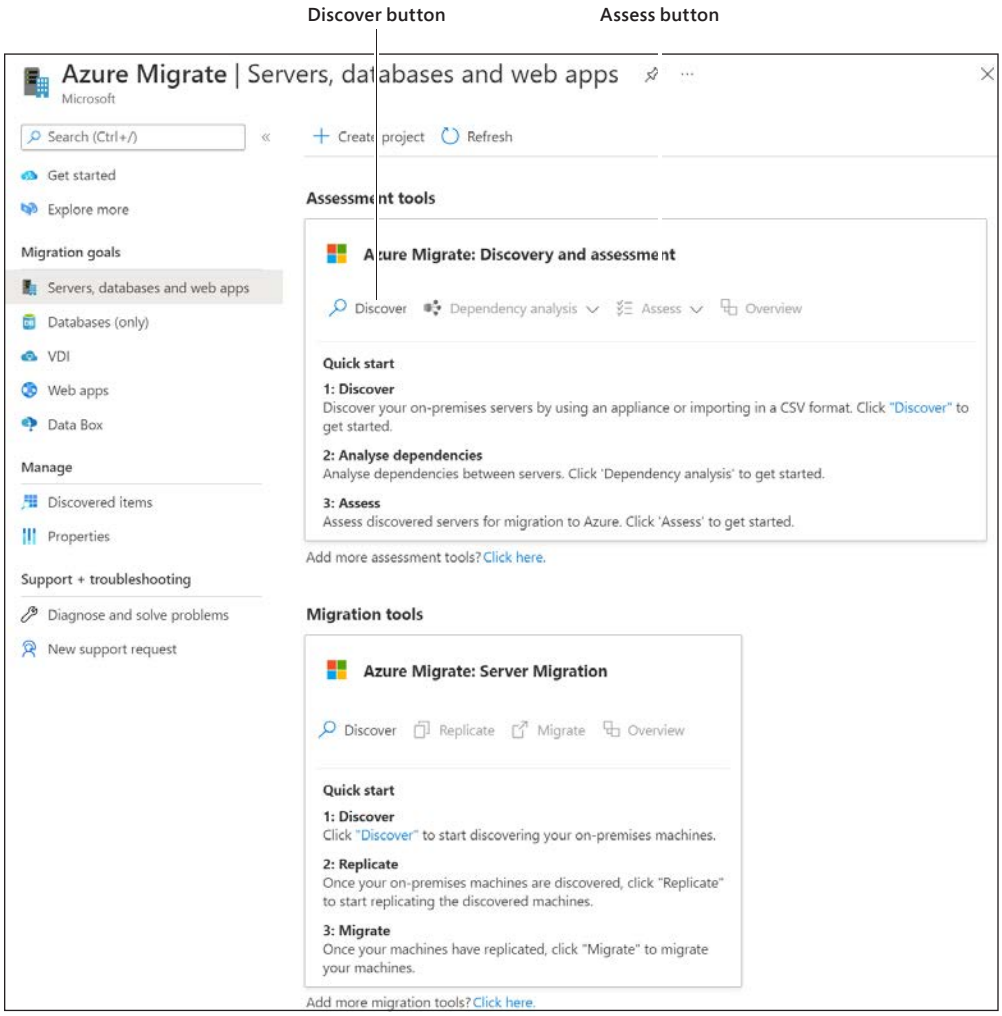


FIGURE 2-41 Azure Migrate tools

Azure Migrate can discover servers, apps, and databases using one of two different methods. One is to document your environment in a comma-separated values (CSV) file using a template provided by Microsoft. The other method is to use the Azure Migrate appliance to automatically discover resources in your environment. The appliance method is preferred. It involves creating a VM in your environment using a Microsoft-provided image. That VM will then go to work collecting information on the resources in your environment.

EXAM TIP

If you want to discover physical servers, servers located in other cloud providers, and so on, you can use third-party assessment tools from companies that partner with Microsoft.

Once discovery is complete, you assess the discovered servers, apps, and databases by clicking the Assess button shown previously in Figure 2-41. The assessment process will determine the requirements of your resources and the best match of resources in Azure for the migration. Assessment can also perform a performance analysis so that Azure Migrate can recommend the best configuration for your situation.

NOTE **PRICING ESTIMATE**

The assessment phase can also provide you with a pricing estimate for your Azure resources after migration.

If you need to move VMWare or Hyper-V VMs to Azure and you don't need to perform an assessment, you can use the **Discover** button in the migration tools shown previously in Figure 2-41. This option allows you to replicate on-premises VMs and then migrate them to Azure without performing an assessment.

Azure Migrate also makes it easy to transfer large amounts of data to Azure. The options we've looked at for copying files to and from Azure Storage are fine for moving smaller numbers of files, but you may need to migrate a lot of data at one time. In fact, enterprises moving to the cloud may need to move many terabytes of data.

For these scenarios, Azure Migrate provides Data Box, a service for transferring large amounts of data to and from Azure. Data Box comes in three different varieties. Data Box Disk, Data Box, and Data Box Heavy.

NOTE **STORAGE SUPPORTED BY DATA BOX**

Data Box supports Azure Blob storage, Azure Files, managed disks used for VMs, and Azure Data Lake Store (ADLS) Gen2 accounts.

Data Box Disk is the least expensive Data Box option, and it provides up to five solid-state disks (SSD) shipped to you from Microsoft. Each disk has about 7 terabytes (TB) of usable space, so the total space provided by a Data Box Disk order is approximately 35 TB. The disks use a USB interface with a maximum transfer rate of 430 megabytes per second.

EXAM TIP

Data Box Disk can only be used to copy data to a single storage account. Data Box and Data Box Heavy can be used to copy data across 10 separate storage accounts.

Data Box consists of a rugged device containing 80 TB of usable disk space in a RAID 5 configuration. The device also contains two gigabit Ethernet (GbE) network interfaces and two additional GbE interfaces for data transfer. For security, the device uses tamper-resistant screws and stickers to make it obvious if someone opens the case. All data on the device is encrypted using AES 256-bit encryption, and the device can only be unlocked using a password obtained in the Azure portal.

> ***NOTE*** **DATA BOX NETWORK INTERFACES**
>
> **One of the interfaces in Data Box is used for the initial setup and management of the device. It is not used for data transfer.**

If your data migration requirements extend beyond Data Box, you can order Data Box Heavy to increase the total storage capacity to about 770 TB. Data Box Heavy consists of a large appliance on a rolling cart that is shipped directly to your datacenter by a freight carrier. Like Data Box, Data Box Heavy contains four network interfaces (one used for management), but the two interfaces used for data transfer are both 40 GbE interfaces. Data Box Heavy is encrypted using AES 256-bit encryption and is protected by security features provided in Azure.

Once you've copied your data to Data Box (whether Data Box Disk, Data Box, or Data Box Heavy), you ship the disks or appliance back to Microsoft. Your data is then copied to Azure Storage, and the disks are wiped as per National Institute of Standards and Technology (NIST) 800-88r1 standards.

Skill 2.4: Describe Azure identity, access, and security

As you might expect, security is of paramount importance in Azure. If you're going to offer a secure environment, you need to know how to identify a user or service that is accessing your resources, and you need to have control over what that user or service can do once they gain access. It's also important to consider that your Azure resources are typically connected to resources outside of Azure, so maintaining a secure environment end-to-end is critical to your overall security strategy.

Azure offers many features designed to cover all facets of security. These include services to identify users and services, authentication services, security options for authentication and access, and much more.

This section covers:

- Directory services in Azure
- Authentication methods in Azure
- External identities and guest access
- Azure AD Conditional Access
- Role-based access control (RBAC)
- Defense in-depth and Zero-trust
- Microsoft Defender for Cloud

Directory services in Azure

In most business applications, not all users have the same privileges. For example, a website might allow a small number of users to add and revise content. Another smaller group of users might be able to decide who can add content. Most users, however, are just consumers of the content. They can't modify the content in any way, and they also can't grant other people the ability to access the content.

To implement this kind of control, you need to know who is using the application so you can determine what their level of privileges should be. To determine who is using the application, you would require that users log in, often with a username and password. Assuming the user provides the right credentials, that user is *authenticated* to the application.

Once a user is authenticated and begins interacting with an application, additional checks might take place to confirm which actions the user is and isn't allowed to perform. That process is called *authorization*, and authorization checks are performed against a user who is already authenticated.

This kind of authentication and authorization scenario isn't limited to a website scenario. When you log into the Azure portal, you are being authenticated. As you interact with Azure resources, you are also authorized to perform the actions you're taking. Based on your privilege level, you are only allowed to do certain things. For example, you might be authorized to create Azure resources but not authorized to give other people access to the Azure subscription you're using.

Azure uses a service called Azure Active Directory to enforce authentication and authorization in Azure, but it has many capabilities beyond simply authentication and authorization.

Azure Active Directory

Azure Active Directory (Azure AD) is an identity service that can control access to resources based on an identity. Azure AD helps you authenticate and authorize users. You can use Azure AD to give users access to Azure resources. You can also give users access to third-party resources used by your company and to on-premises resources, all using the same username and password.

MORE INFO **GRANTING ACCESS TO AZURE RESOURCES**

You'll learn about how you can authorize users to access your Azure resources when we cover role-based access control later in this skill.

The core of Azure AD is a directory of users. Each user has an *identity* that's composed of a user ID, a password, and other properties. Users also have one or more *directory roles* assigned to them. The user ID and password are used to authenticate the user, and the roles are used for authorization to perform certain activities in Azure AD.

MORE INFO **SERVICE PRINCIPALS AND MANAGED IDENTITIES**

Two other entities available in Azure AD are service principals and managed identities. Service principals represent an application in Azure AD, and we'll discuss them later in this section. A managed identity is a special kind of service principal that can only be used with Azure resources. You can read more about them at *https://bit.ly/az900-managedidentities*.

When you sign up for an Azure subscription, an Azure AD resource is automatically created for you, and it's used to control access to Azure resources you create under your subscription. Figure 2-42 shows Azure AD in the Azure portal.

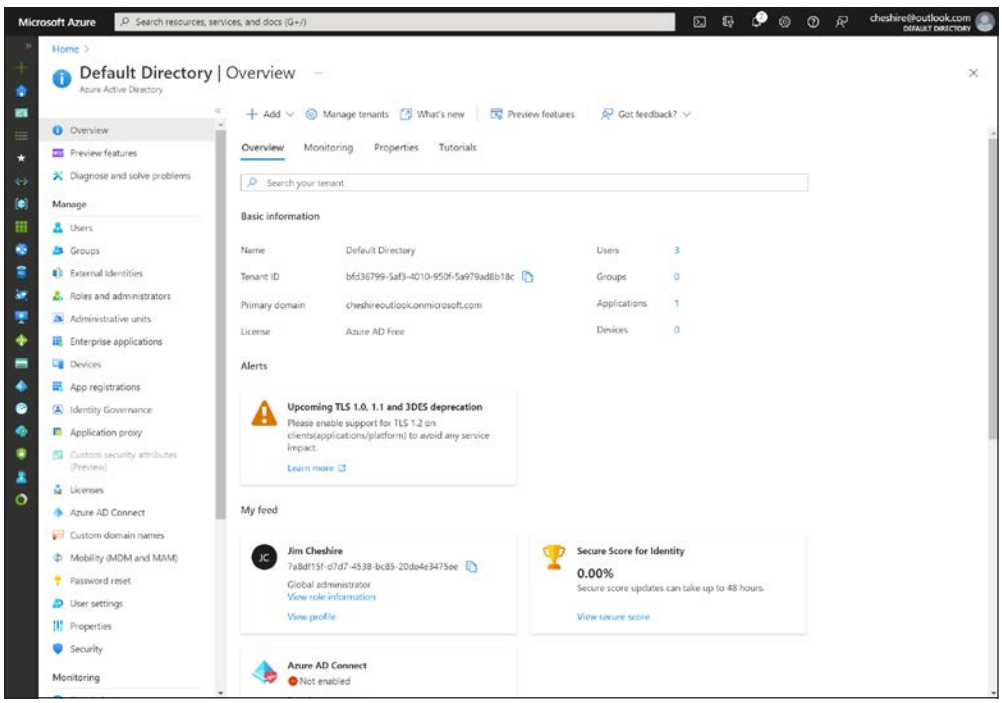


FIGURE 2-42 Azure AD in the Azure portal

MORE INFO **USING AZURE AD**

You'll learn more about using Azure AD as you progress through this skill.

Azure Active Directory Domain Services

For many years, Active Directory Domain Services (AD DS) has been an integral component of managing access to information and resources located on Windows networks. Information, users, and resources are all contained within a *directory*, and Active Directory allows that directory to be segmented into smaller units called ***domains***, and specialized servers called ***domain controllers*** (DCs) are used to manage each domain. Domains can be combined into ***forests***, allowing access to resources across domains.

NOTE **ACTIVE DIRECTORY DOMAIN SERVICES**

A full discussion of AD DS is outside the scope of this book. You don't need to know details about AD DS for the Microsoft Azure Fundamentals exam.

Because so many Windows users are familiar with managing users and access to resources using AD DS, Azure provides a cloud-based implementation of AD DS in Azure AD. This offering is called Azure Active Directory Domain Services, or Azure AD DS. Azure AD DS provides the same services that users of AD DS are familiar with, including the ability to join domains, group policies, and Kerberos and Windows New Technology LAN Manager (NTLM) authentication.

EXAM TIP

There are three primary reasons for using Azure AD DS. One is the use of older applications that don't support the authentication mechanisms in use today. Another is a scenario where you want to integrate Azure resources into your on-premises Windows Active Directory. Finally, if you want to lift-and-shift on-premises applications to the cloud that rely on Windows AD DS, Azure AD DS enables you to do so.

To use Azure AD DS, you first create an Azure AD DS managed domain. This managed domain is the equivalent of an on-premises Windows Active Directory domain, but in Azure AD DS, Microsoft manages the DCs for you. This includes ensuring they are encrypted and backed up regularly.

When you create your managed domain, Azure adds two DCs to your domain, and Azure calls these DCs a *replica set*. You can't manage these DCs directly. As I said earlier, they are completely managed by Azure. Once your domain has been created, you can join your VMs in Azure to the domain just as you would with an on-premises server.

If you are operating in a hybrid cloud scenario, you may want to connect your on-premises Windows AD with Azure AD DS. Azure AD Connect allows you to connect your on-premises AD domains with Azure AD DS domains. It can even ensure that your on-premises and cloud domains are synchronized.

To use Azure AD Connect, you download and install it on your on-premises server. The initial setup will ask you for your Azure AD DS credentials as well as your Windows AD DS credentials. Azure AD Connect takes care of the rest for you. You can, of course, perform more complex configuration depending on your needs.

MORE INFO **AZURE AD CONNECT**

For more information on getting started with Azure AD Connect, see *http://bit.ly/az900-adconnect*.

Azure AD DS offers three SKUs. The Standard SKU is the least expensive and is designed for a maximum of 25,000 objects in the directory and 3,000 authentications per hour. (While not an explicit limitation, exceeding these values can cause performance degradation.) The Standard SKU is designed for use with a single forest.

The Enterprise SKU increases performance to the point where you can support up to 10,000 hourly authentications and 100,000 objects. It also allows for up to five resource forest trusts, allowing access to resources across multiple forests.

The highest SKU is the Premium SKU, and this provides high performance with up to 70,000 hourly authentications and up to 500,000 objects. It also adds an additional five resource forest trusts over the five available in the Enterprise SKU.

Authentication methods in Azure

In the simplest sense, administrators of Azure AD can decide if a user has access to a particular resource by requiring that the user be authenticated with a username and password and has the authorization to access that resource. However, most administrators want much more control than that to keep resources secure.

Suppose you have given a user named Christine access to your Azure resources, and you need to have confidence that someone can't hack Christine's password and gain access to your data. Azure Conditional Access and multifactor authentication (MFA) can help make Christine's account much more secure, and single sign-on can help Christine access her corporate resources without having to enter her username and password. Let's start by looking at single sign-on.

Single sign-on (SSO)

Single sign-on (SSO) is a simple concept with a huge impact for users. When using SSO, users can access corporate resources at their companies without having to enter a username and password. Instead, they are authenticated using the log-in credentials they supplied when logging into the computer they're using. SSO not only provides convenience, but it also provides a more secure environment because passwords aren't being entered for every resource.

For a device to work with SSO, it must be joined to Azure AD. Once the device has been added to AD, the user can access Azure resources and other company resources such as SharePoint and Microsoft 365 resources using SSO. Users can also access on-premises resources using SSO.

EXAM TIP

SSO to on-premises resources is implemented using Azure AD Connect.

SSO supports two sign-in methods: password hash synchronization and pass-through authentication. Password hash synchronization copies a user's password in a hashed format to Azure AD. Because the password is hashed, the actual password can't be retrieved. When the user enters the password, an algorithm is used to generate a hash, and that hash is compared against the hash stored in Azure AD. If the two are the same, the user is authenticated.

Pass-through authentication passes a user's login on Azure AD into an on-premises pass-through authentication agent, and that agent sends the authentication to the on-premises Windows Active Directory instance. Once authenticated, Azure AD Connect is used to pass that authentication through to Azure AD and the user's resources.

Multifactor authentication (MFA)

By default, users can log in to your Azure AD using only a username and password. Even if you require your users to use strong passwords, allowing access to your resources with only a username and password is risky. If a hacker obtains the password by using software that guesses passwords or by stealing it through phishing or some other means, your resources are no longer secure.

Multifactor authentication solves this problem. The concept behind multifactor authentication is that you must authenticate using a combination of:

- Something you know, such as a username and password
- Something you have, such as a phone or mobile device
- Something you are, such as facial recognition or a fingerprint

If multifactor authentication requires all three of these, it's referred to as three-factor authentication, or sometimes 3FA. If only the first two are required, it's referred to as two-factor authentication, or sometimes 2FA. (Microsoft calls this *two-step verification*.) Azure multifactor authentication is two-factor authentication.

NOTE **BIOMETRICS IN MOBILE DEVICES**

Even though Azure multifactor authentication is two-factor, if you are using a device that includes biometric features, you might be authenticating using three-factor authentication. However, the third factor is enforced by your device and not by Azure. Azure multifactor authentication doesn't require three-factor authentication.

To enable multifactor authentication for one or more users of your Azure AD, open the **All Users** blade and click **Per-User MFA**, as shown in Figure 2-43.

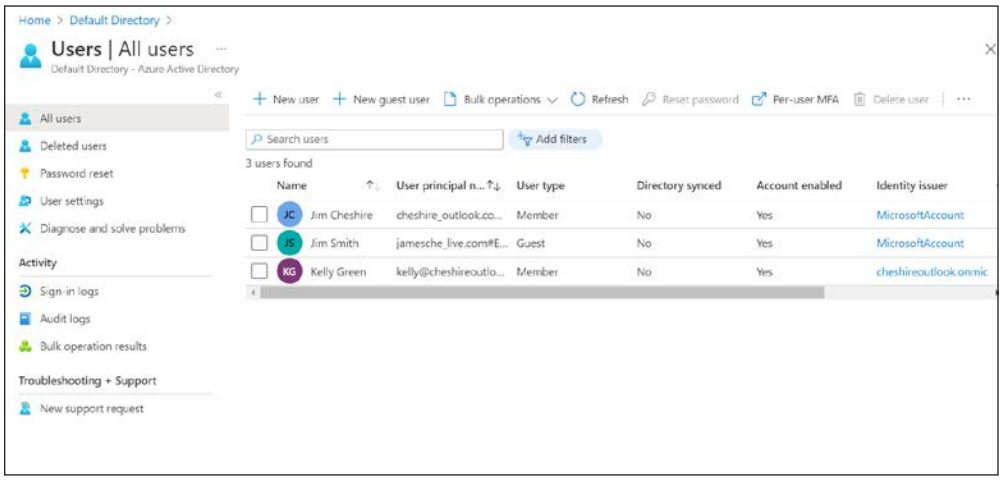


FIGURE 2-43 Enabling multifactor authentication

MORE INFO **COMBINED SECURITY INFORMATION REGISTRATION**

For a better user experience, Microsoft recommends you use combined registration to allow users to register for MFA along with self-service password reset in one operation. You can read more about this at *https://bit.ly/az900-combinedregistration*.

When you click **Per-User MFA**, a new browser window opens and displays the Azure AD user management site. Select one or more users for whom you want to enable multifactor authentication and click **Enable**, as shown in Figure 2-44.

Once a user is required to use MFA, they need to take a second step when logging in to the Azure portal. This can be a prompt from the Microsoft Authenticator app (available for iOS and Android), authentication via Windows Hello for Business, a Fast IDentity Online (FIDO2) security key, an Open Authorization (OAUTH) token, an SMS message with an access number, or a phone call requiring you to enter an access code.

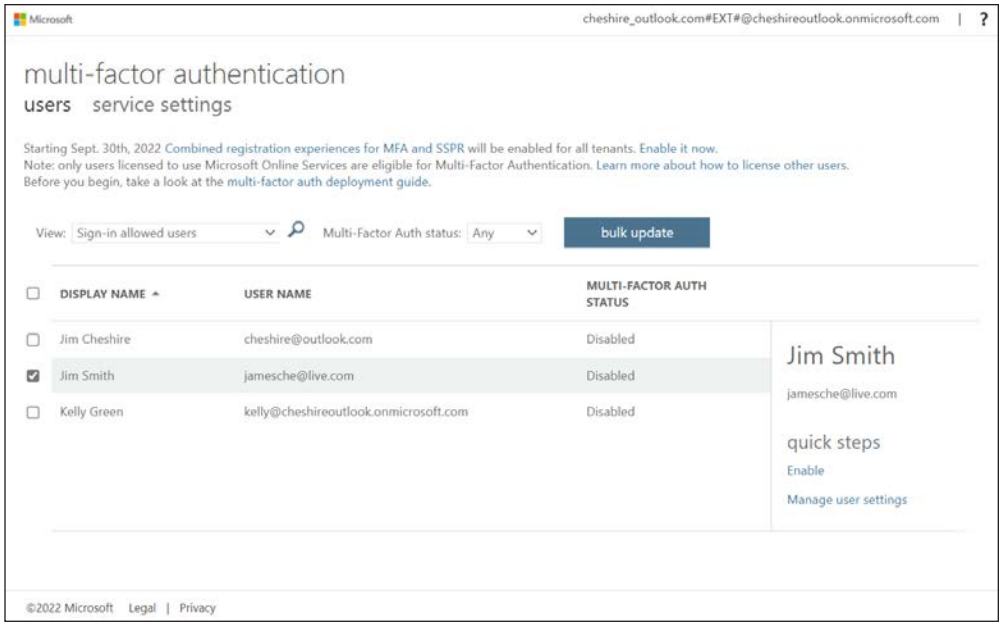


FIGURE 2-44 Enabling multifactor authentication

Passwordless authentication

Multifactor authentication is definitely more secure than using only a password, but the truth is that most people don't use multifactor authentication even when it's available to them. Many people find multifactor authentication to be a hassle. Passwordless authentication provides the same level of security while removing the hassle of multifactor authentication.

> ***NOTE*** **PASSWORDLESS AND MULTIFACTOR AUTHENTICATION**
>
> **Passwordless authentication still uses multifactor authentication. However, instead of authenticating users with something they know (a password) and something they have, passwordless can authenticate users with something they have (a device or security key) and something they are via biometrics.**
>
> **Having said that, passwordless can also authenticate using something a user knows when it allows for authentication using a PIN.**

To enable passwordless authentication, open your Azure AD instance in the portal, click the **Security** menu, and then click **Authentication Methods**. Click on the passwordless method you want to enable, as shown in Figure 2-45.

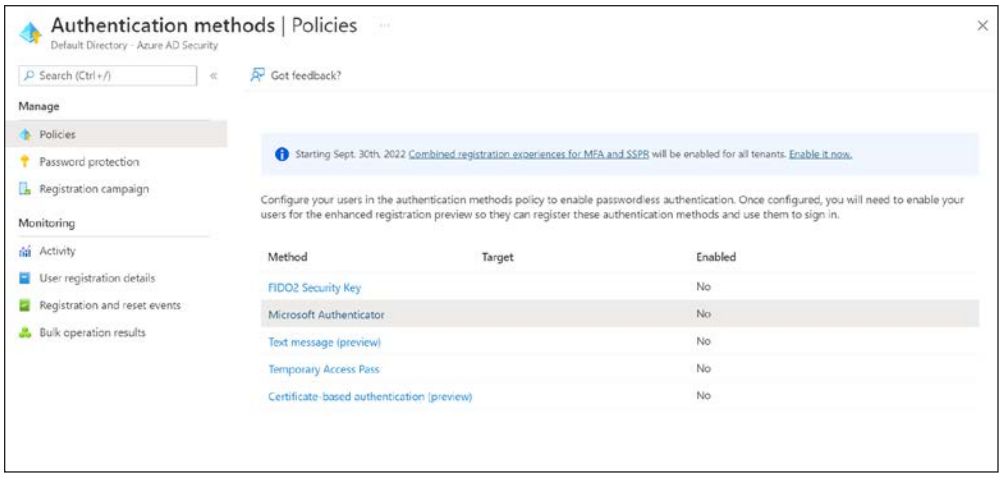


FIGURE 2-45 Passwordless authentication methods

Once you've clicked the desired authentication method, click **Yes** to enable it, as shown in Figure 2-46. You also have the option of applying the change to all users or to selected users. Once you've made your selections, click **Save** to apply them.

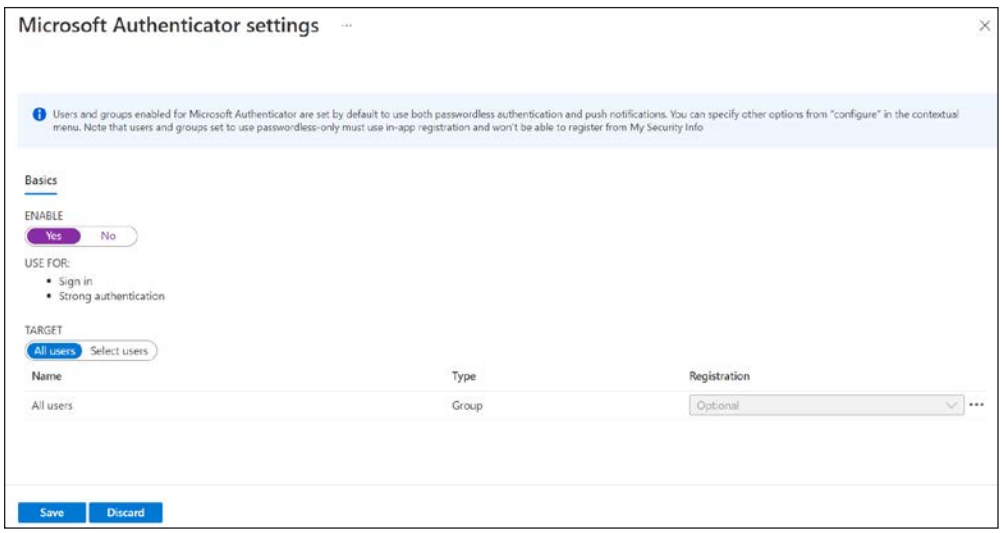


FIGURE 2-46 Enabling an authentication method

MORE INFO **PASSWORDLESS WIZARD**

Enabling some methods (such as Windows Hello for Business) requires planning and more detailed steps. Microsoft has a wizard that can help you with enabling those options, and you can access that wizard at *https://aka.ms/passwordlesswizard*. Note that you must have an admin role for your Microsoft 365 tenant to access the wizard.

External identities and guest access

Earlier in this chapter, I said that Azure AD contains a directory of users. If you're using an Azure subscription as an individual, it's likely you are the only user in your Azure AD. However, if you are using a subscription provided to you by your company, your Azure AD directory of users likely contains all the people in your organization.

Users who are in your Azure AD and part of your organization are called *members* of your Azure AD. You can also add people from outside of your organization to your Azure AD, and those users are considered *guests*. Guests are added to your Azure AD when you want to allow them access to your resources or apps using their own credentials.

To view or manage users in Azure AD, click **Users** in the menu on the left side of the page. This opens the **All Users** blade shown in Figure 2-47.

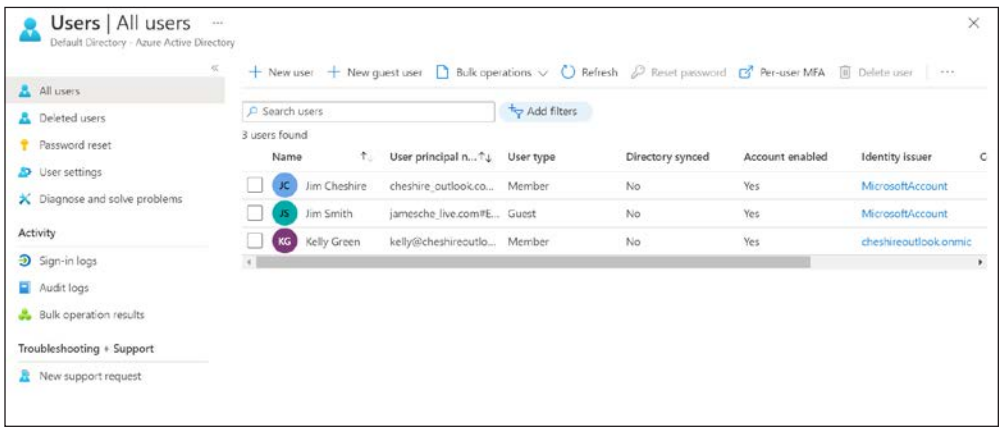


FIGURE 2-47 The All Users blade in the Azure portal

The Azure AD shown in Figure 2-47 contains three users. The first two users are using a Microsoft Account to log into Azure AD, and the second user is identified as a guest. The third user is a member who has been added to Azure AD, and that user must authenticate using a username and password provided to them when they were added to Azure AD.

To add a new user from your company to your Azure AD, click **New User** to display the blade shown in Figure 2-48.

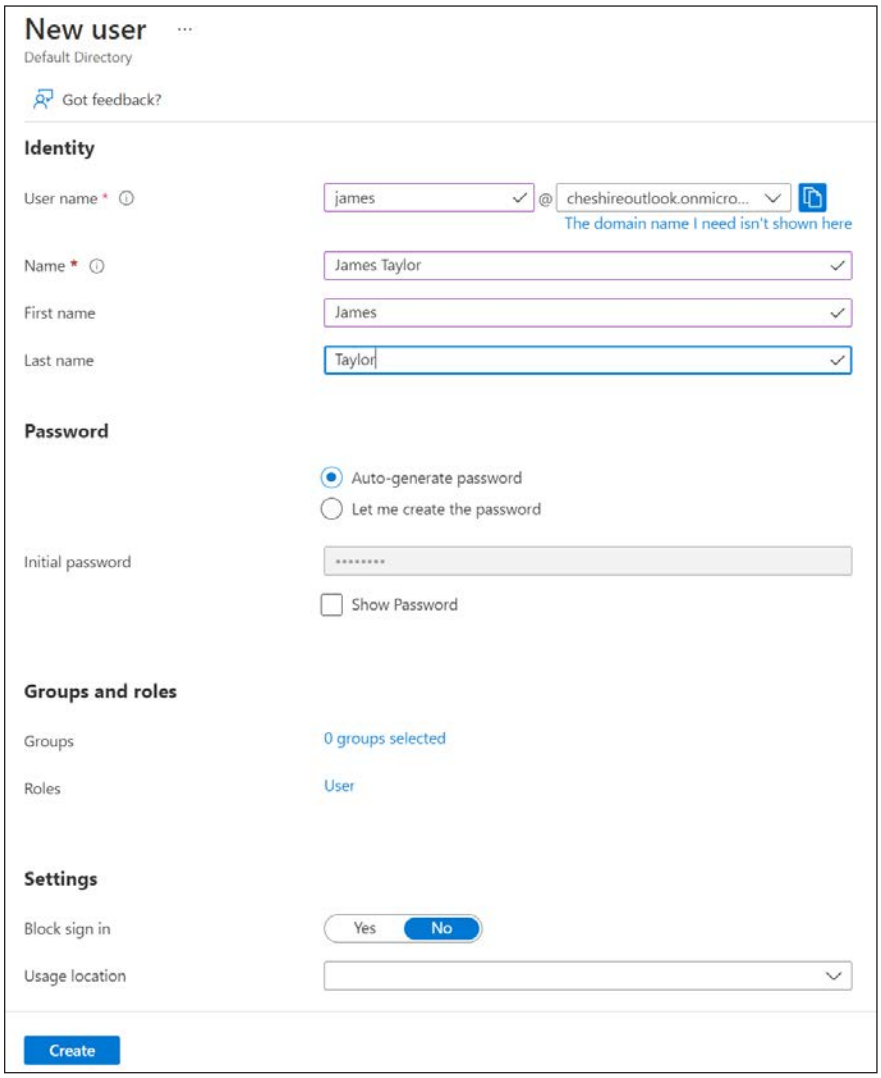


FIGURE 2-48 Adding a new Azure AD user

The specified username is used to log in to Azure AD. The domain name you use must be one that you own and that is associated with your Azure AD. You can also assign the new user to a group or a role. Groups make it easier to manage a larger number of similar users.

As I mentioned previously, you can invite guest users from outside of your company to be members of your Azure AD. Those users can then be given access to your resources. This method of collaboration uses an Azure AD feature called Azure AD B2B (business-to-business) collaboration. To add a guest user, click **New Guest User**, as shown previously in Figure 2-47. This will open the **New Guest User** blade, as shown in Figure 2-49.

When you invite a guest user, an invitation to join your Azure AD is sent to the email address you specify. To accept the invite, the user's email address must be associated with an identity provider you've configured. If the user doesn't have an account with a configured identity provider (for example, a Microsoft Account or a Google account), the user will be given the option to create an account to join your Azure AD.

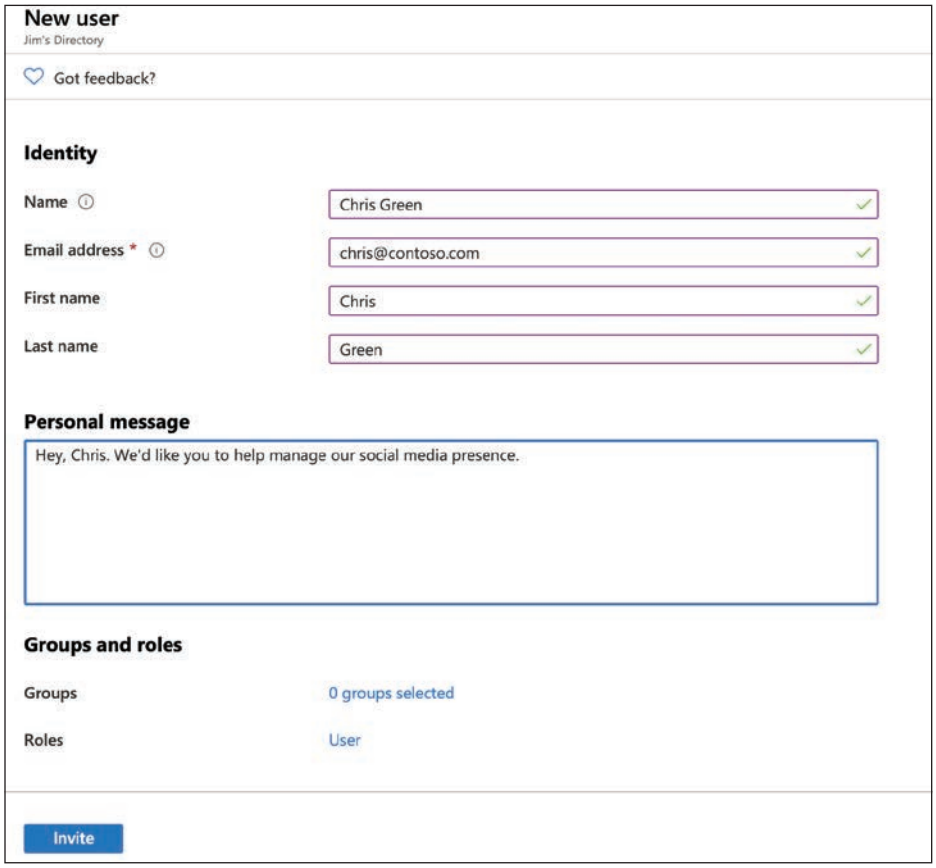


FIGURE 2-49 Adding a new guest user

The user in Figure 2-49 can be given access to the corporate social media accounts by adding those applications to Azure AD. Thousands of applications can be added, including social media apps such as Facebook and Twitter. To add an application, open Azure AD in the Azure portal, click **Enterprise Applications** (shown previously in Figure 2-42) and click **New Application**, as shown in Figure 2-50.

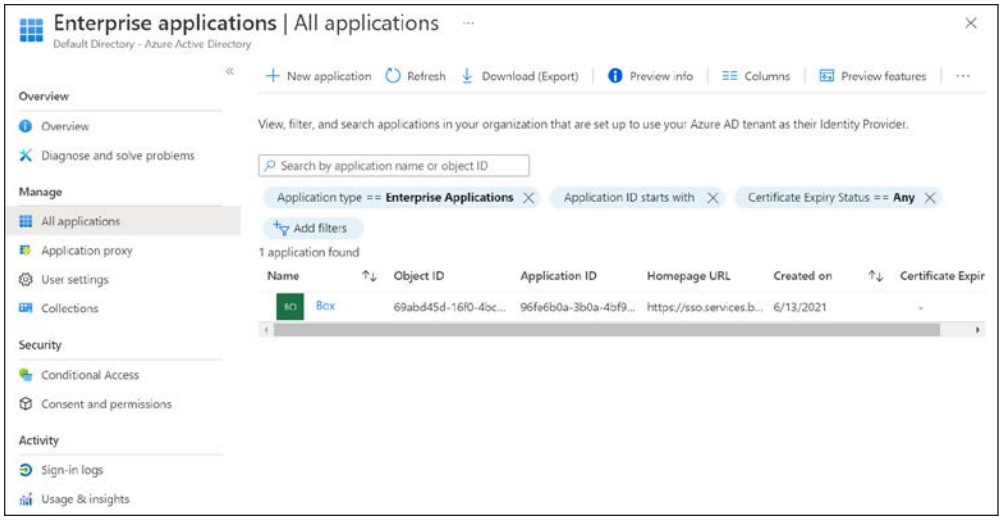


FIGURE 2-50 Enterprise applications in Azure AD

After you click **New Application**, you can choose from popular cloud providers, as shown in Figure 2-51. You can search for an application from here by entering an application name in the search box. You can also filter the view using the filtering buttons to the right of the search box.

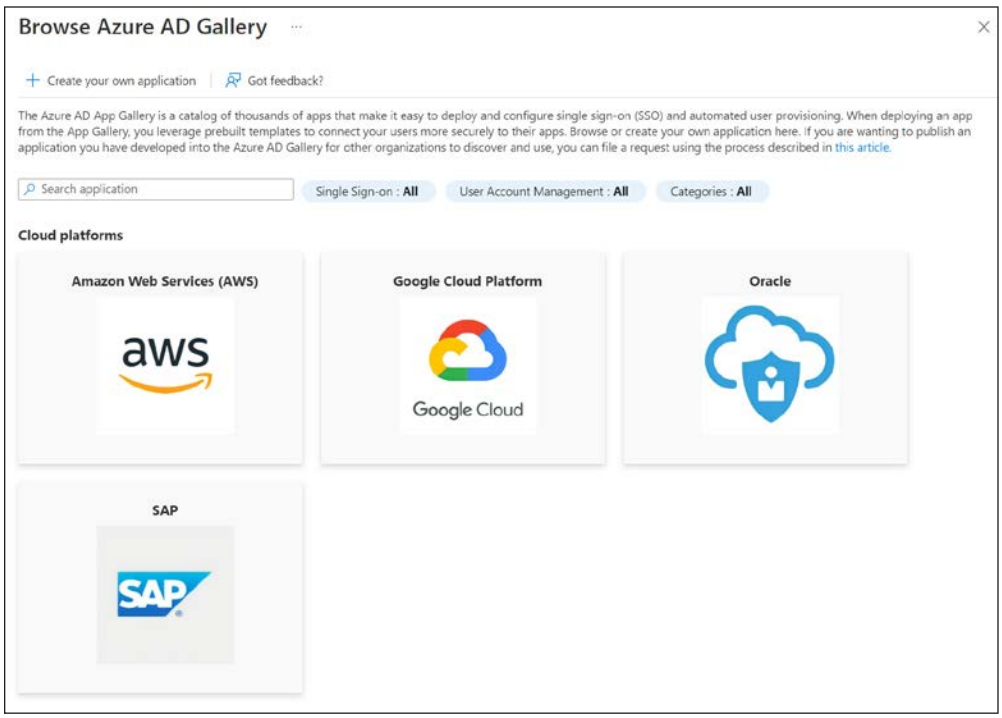


FIGURE 2-51 Cloud platforms in the enterprise application gallery

If you scroll down, you'll see a list of many other applications you can add, as shown in Figure 2-52.

You can also add your own application, add an application that exists in your on-premises environment, or integrate any other application. The application that you add needs to expose a sign-in page to which you can point Azure AD to integrate it.

EXAM TIP

You can configure which resources an application can access using a *service principal*. The service principal is created when you give an application access to Azure resources using role-based access control, which is a concept you'll learn about in the next section.

After you add an application, you can configure Azure AD so that users with access to that application can authenticate to it using the same credentials they use to log in to Azure AD. This kind of authentication is known as *single sign-on* (or SSO), and it's one of the key benefits to using Azure AD.

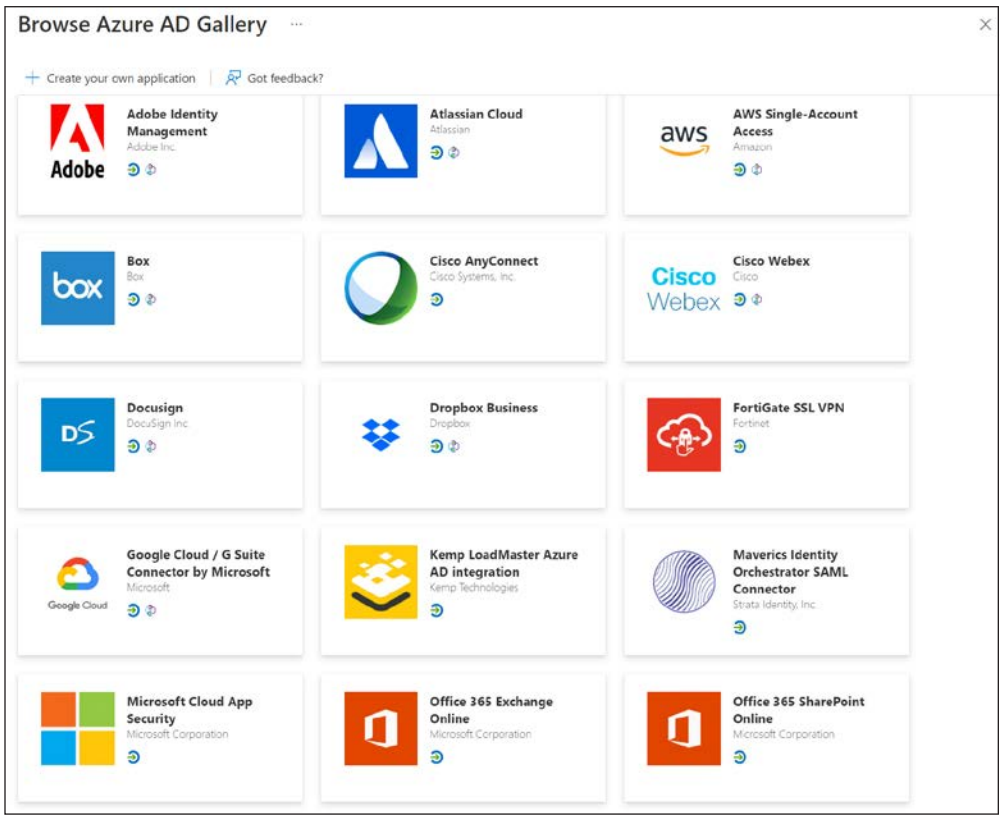


FIGURE 2-52 Enterprise application gallery apps

Azure AD Conditional Access

Azure AD Conditional Access allows you to create policies that are applied against users. These policies use *assignments* and *access controls* to configure access to your resources.

Assignments define who a policy applies to. It can apply to users, groups of users, roles in your Azure AD, or to guest users. You can also specify that a policy only applies to specific applications.

Assignments can also define conditions that must be met (such as requiring a certain platform such as iOS, Android, Windows, and so on), specific locations by IP address, and more.

Access controls determine how a Conditional Access policy is enforced. The most restrictive access control is block access, but you can also use access controls to require that a user use a device that meets certain conditions, that they're using an approved application to access your resources, that they are using MFA, and so on.

To create a Conditional Access policy, search for **Azure AD Conditional Access** in the Azure portal. You can then click the **New Policy** button to create a new policy, as shown in Figure 2-53.

EXAM TIP

Conditional Access is only available in the Premium tiers of Azure AD. Because the free version of Azure AD is being used in these examples, the New Policy button is disabled in Figure 2-53.

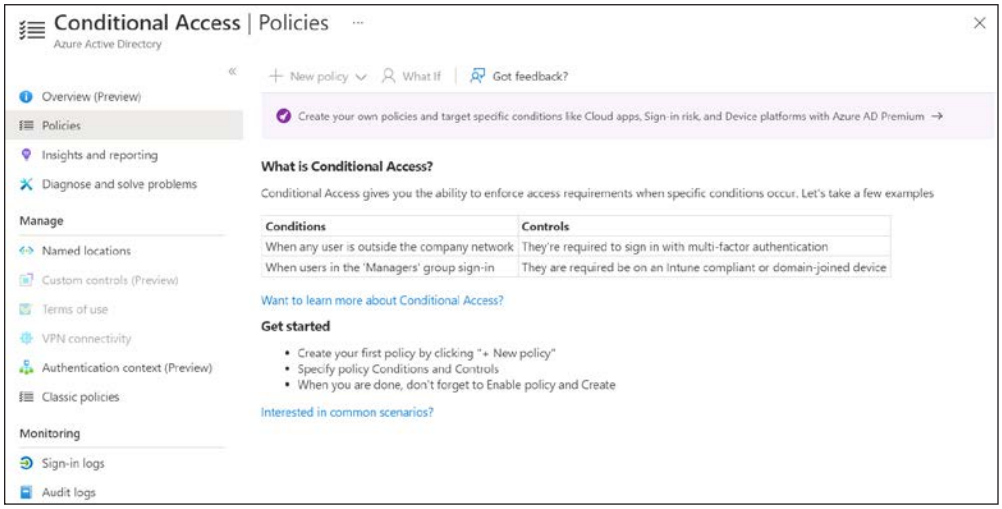


FIGURE 2-53 The Conditional Access Policies blade in the Azure portal

Role-based access control (RBAC)

Role-based access control (RBAC) is a generic term that refers to the concept of authorizing users to a system that is based on defined roles to which the user belongs. Azure implements RBAC across all Azure resources, so you can control how users and applications can interact with your Azure resources.

You might want to allow users who administer your databases to have access to databases in a particular resource group, but you don't want to allow those people to create new databases or delete existing databases. You might also want some web developers to be able to deploy new code to your web applications, but you don't want them to be able to scale the app to a higher-priced plan. These are just two examples of what you can do with RBAC in Azure.

There are four elements to RBAC:

- **Security principal** The security principal represents an identity. It can be a user, a group, an application (which is called a service principal), or a special AAD entity called a *managed identity*. A managed identity is how you authorize another Azure service to access your Azure resource.
- **Role** A role (sometimes called a role definition) is what defines how the security principal can interact with an Azure resource. For example, a role might define that a security principal can read the properties of a resource but cannot create new resources or delete existing resources.
- **Scope** The scope defines the level at which the role is applied, and it specifies how much control the security principal has. For example, if the scope is a resource group, the role defines activities that can be performed on all resources in the resource group.
- **Role assignments** Roles are assigned to a security principal at a particular scope, and that's what ultimately defines the level of access for the security principal.

RBAC includes many built-in roles. Three of these built-in roles apply to all Azure resources:

- **Owner** Members of this role have full access to the resources.
- **Contributor** Members of this role can create resources and manage resources, but they cannot delegate that right to anyone else.
- **Reader** Members of this role can see Azure resources, but they cannot create, delete, or manage those resources.

All the other built-in roles are specific to certain types of Azure resources.

To give someone access to a resource using RBAC, open the resource to which you want to give access in the Azure portal. Click **Access Control (IAM)** in the portal to configure RBAC. In Figure 2-54, RBAC is being configured for a web app hosted in Azure App Service. Clicking **Add Role Assignment** in the **Grant Access To This Resource** box allows you to add a role.

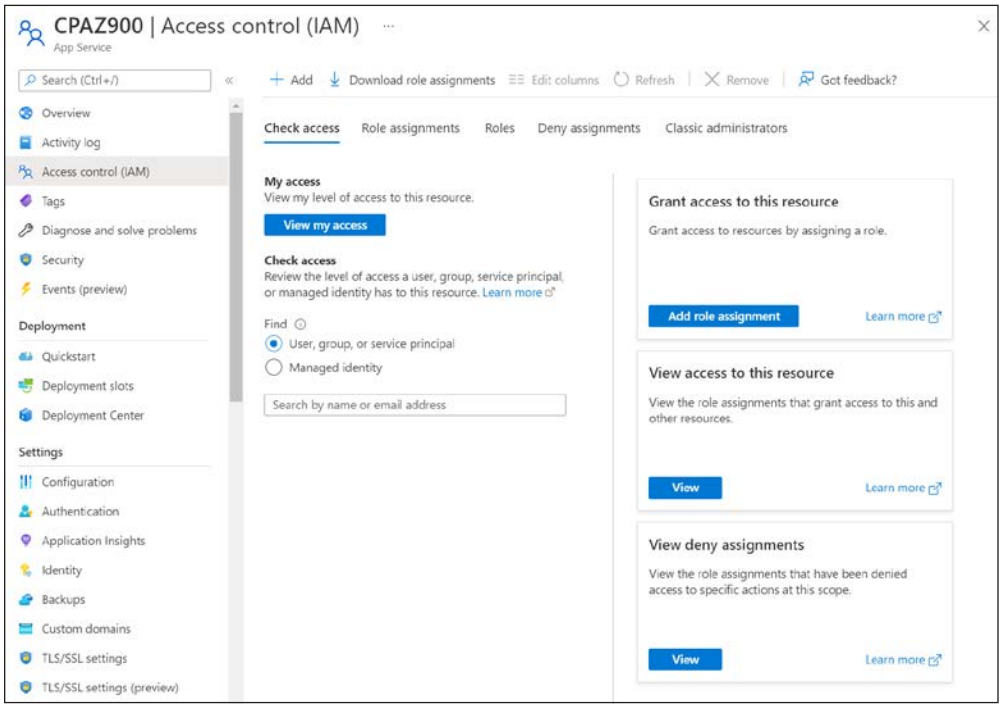


FIGURE 2-54 Configuring RBAC for a web app

EXAM TIP

The scope of RBAC is defined by where the RBAC role is assigned. For example, if you open a resource group in the portal and assign an RBAC role to a user, the scope is at the resource group level. On the other hand, if you open a web app within that resource group and assign the role, the scope is to that web app only.

RBAC roles can be scoped to the management group, subscription, resource group, or resource level.

After clicking **Add**, choose the role you want to assign. The list of roles will differ depending on what type of resource this is. Click the role you want to assign and then click the **Next** button as shown in Figure 2-55 to apply that role to one or more users.

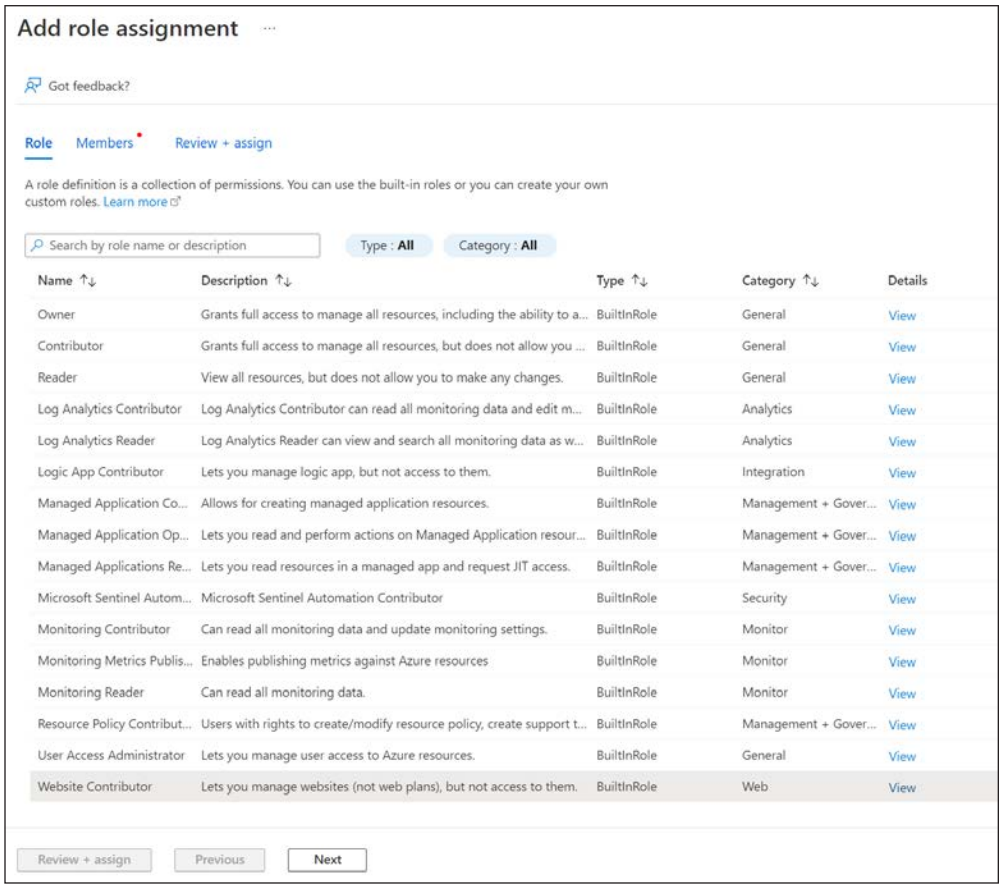


FIGURE 2-55 Adding a role assignment

To assign the selected role to one or more users, click **Select Members**, select the user(s), and then click the **Select** button, as shown in Figure 2-56.

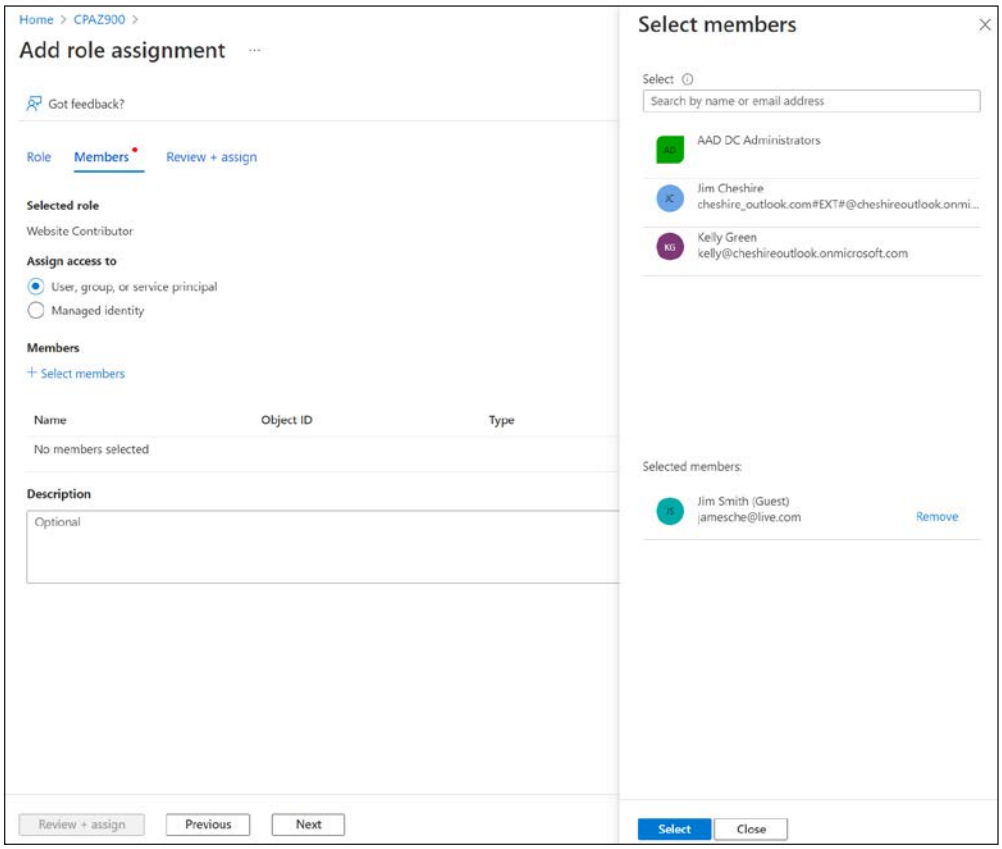


FIGURE 2-56 Selecting a member for role assignment

Figure 2-56 shows a list of users in the AAD because the **Assign Access To** radio button is set to **User, Group, Or Service Principal**. You can see a list of other types of objects by selecting **Managed Identity** as the type. For example, in Figure 2-57, we are selecting a system-assigned managed identity type called **Virtual Machine**, and this will allow us to select from a list of VMs to assign to the role.

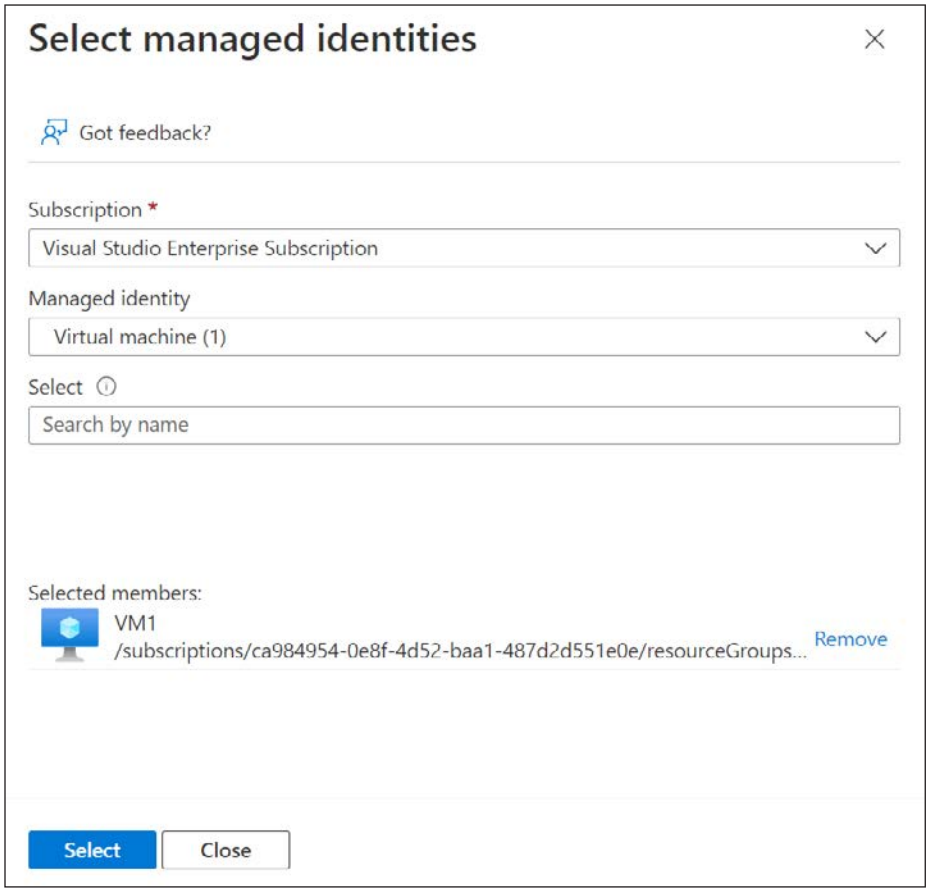


FIGURE 2-57 Using a managed identity to assign a role

EXAM TIP

It's important to understand that role assignments are additive. Your RBAC abilities at any scope are the result of all role assignments up to that level. In other words, if I have the Owner role on a resource group and you assign me the Website Contributor role on a web app within that resource group, the Website Contributor assignment will have no effect because I already have the Owner role on the entire resource group.

RBAC is enforced by Azure Resource Manager (ARM). When you attempt to interact with an Azure resource, whether in the Azure portal or by using a command line tool, you are authenticated by ARM and a token is generated for you. That token is a representation of your identity and all your role assignments, and it's included with all operations you perform on the resource. ARM can determine if the action you are performing is allowed by the roles to which you are assigned. If it is, the call succeeds; if not, you are denied access.

You can ensure that someone has the rights you desire by checking access in the Azure portal. After opening the resource and clicking **Access Control (IAM)**, use the radio buttons and search box shown previously in Figure 2-54 to search for a user or object and then click the user or object to see the access level, as shown in Figure 2-58.

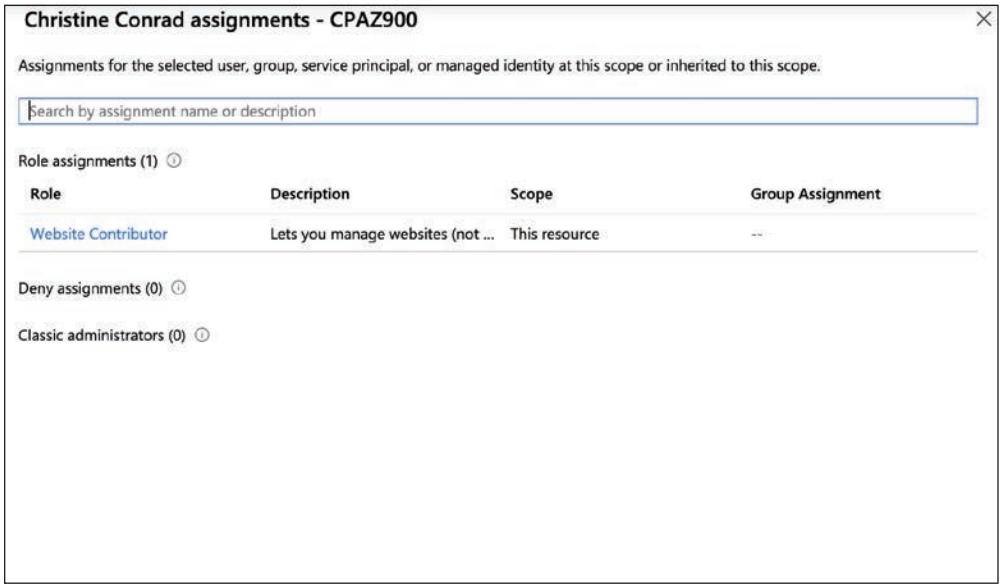


FIGURE 2-58 Showing RBAC assignments for a user

For a greater level of detail on what exact operations are and aren't allowed, click the role that's displayed. This will allow you to see a detailed list of operations and the combination of read, write, delete, and other actions that a security principal can perform.

Defense in depth and Zero-trust

Just for a moment, transport yourself back to medieval times and think about what it was like living in a castle. These weren't friendly times in many ways, and there was always a hostile force trying to enter the fortress. To prevent invasion, moats were built around castles. The purpose of the moat was to prevent an opposing force from tunneling under the wall and gaining entry.

Even before an enemy reached the moat, archers along the high wall of the castle would pose a formidable risk to attackers approaching the castle. Assuming an opposing force made it past the archers and traversed the moat, they were met with a high wall and a sturdy gate. If they were able to make it past the gate, they were met with an army of warriors with swords and other nasty weapons.

Medieval folks had a pretty good idea when it came to security. They realized that a single opposing force wouldn't be enough to keep them secure. They needed layered opposition so that anyone defeating one method of security would be met with several more down the line.

This is a perfect example of defense in depth, and it's why defense in depth is often referred to as the "castle approach." When it comes to network security, this multi-layered approach is also the best way to keep your network safe. Azure Firewall can help prevent a malicious user from making it into your network, Network Security Groups can help you control network traffic inside your network, and Azure DDoS Protection can help to identify and mitigate malicious traffic that might otherwise seem normal.

The concept of defense in depth is only a starting point when it comes to securing your resources. To implement and maintain a solid security approach, you must have a strategy in place. While that sounds reasonable, many of us are not security experts. Thankfully, Microsoft has developed a strategy that you can adopt called Zero-trust.

A traditional approach to security was to consider everything happening inside of an internal network as trusted. While operations on resources in that network relied on proper authorization, traffic within the network was not considered a breach of security.

This model worked well for a long time, but as the IT world has evolved, it has become too restrictive. This has been made especially clear due to the enormous increase of the hybrid workforce in the era of Covid-19.

MORE INFO **IMPLEMENTING ZERO-TRUST**

Microsoft has a documented plan for implementing Zero-trust that can help you transition. You can find it at *https://bit.ly/az900-zerotrust*.

The Zero-trust framework relies on identifying users, but instead of simply locking down a network, Zero-trust uses Conditional Access policies to control access to resources. This model extends to network endpoints, data, apps, infrastructure, and the network. Multifactor authentication is also a cornerstone of this approach.

Zero-trust doesn't end with authentication. Your apps and your networks must also adopt Zero-trust principles. Apps should be designed to allow the lowest level of access necessary, and every access to an app and its data should be assumed to be a breach until determined otherwise.

Networks should take the same approach by using features such as Azure Firewall, Network Security Groups, DDoS protections, Azure VPN Gateway, and so forth.

Microsoft Defender for Cloud

Security features in Azure have evolved over the past few years. What was once a collection of various security features and services has now coalesced into a single solution called Microsoft Defender for Cloud.

Defender for Cloud can help you protect your cloud resources, not only in Azure, but also in Amazon Web Services (AWS) and the Google Cloud Platform (GCP). It can also aid in ensuring regulatory compliance.

Defender for Cloud provides insight into four different areas:

- **Security posture** Provides a Secure Score, which shows the security of your resources. This area also shows a breakdown of management groups, subscriptions, unhealthy resources, and any recommendations.
- **Regulatory Compliance** Provides a high-level overview of the regulatory compliance of your Azure resources.
- **Workload protections** Shows you the percentage of protection for each of your service types.
- **Firewall Manager** Provides insights into the security of your networks in Azure.

When you first open Defender for Cloud in the portal, you are taken to the Overview page, as shown in Figure 2-59. This provides a high-level overview of the resources protected by Defender for Cloud across the four areas highlighted above.

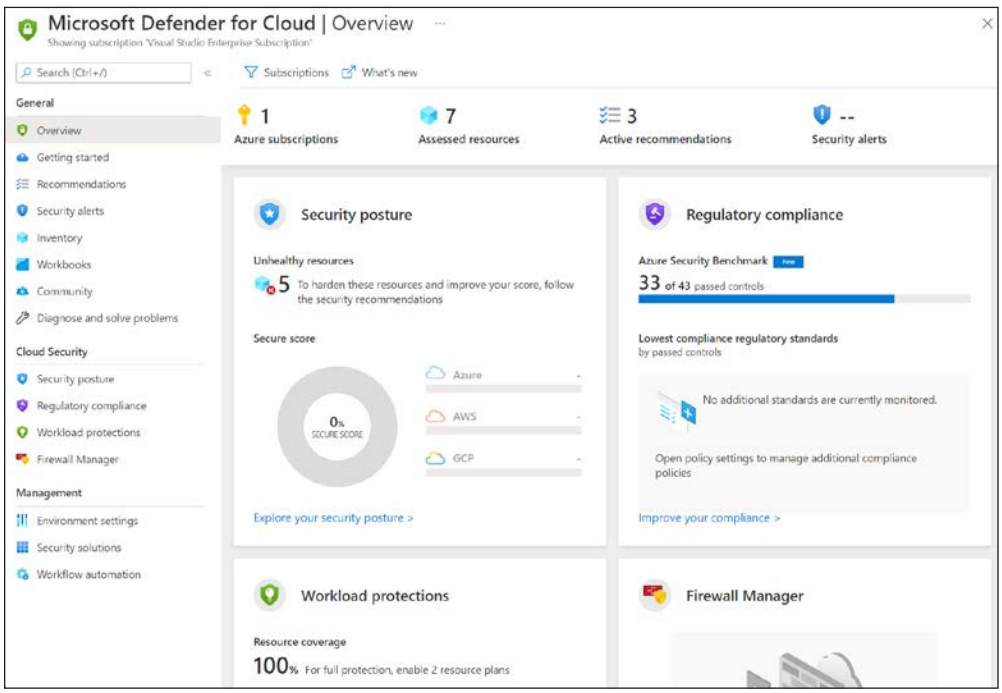


FIGURE 2-59 Defender for Cloud

By clicking on **Regulatory Compliance** in the menu on the left, you can see how your resources fare related to various regulation controls, as shown in Figure 2-60. By clicking on the controls, you can see details of compliance, and clicking on a specific result will take you to further details about how to remediate the issue.

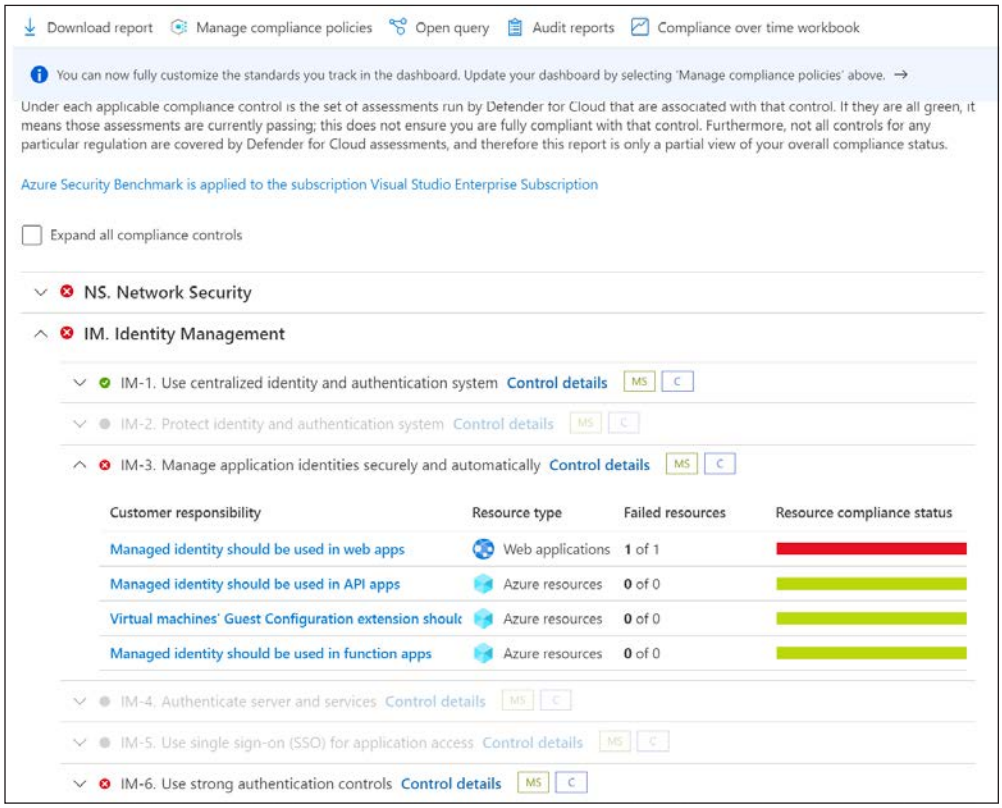


FIGURE 2-60 Regulatory compliance in Defender for Cloud

The **Workload Protections** screen shows you the percentage of your resources covered by Defender for Cloud, as shown in Figure 2-61. You'll also see a timeline of any security alerts and the status of advanced protection for all your Azure resources.

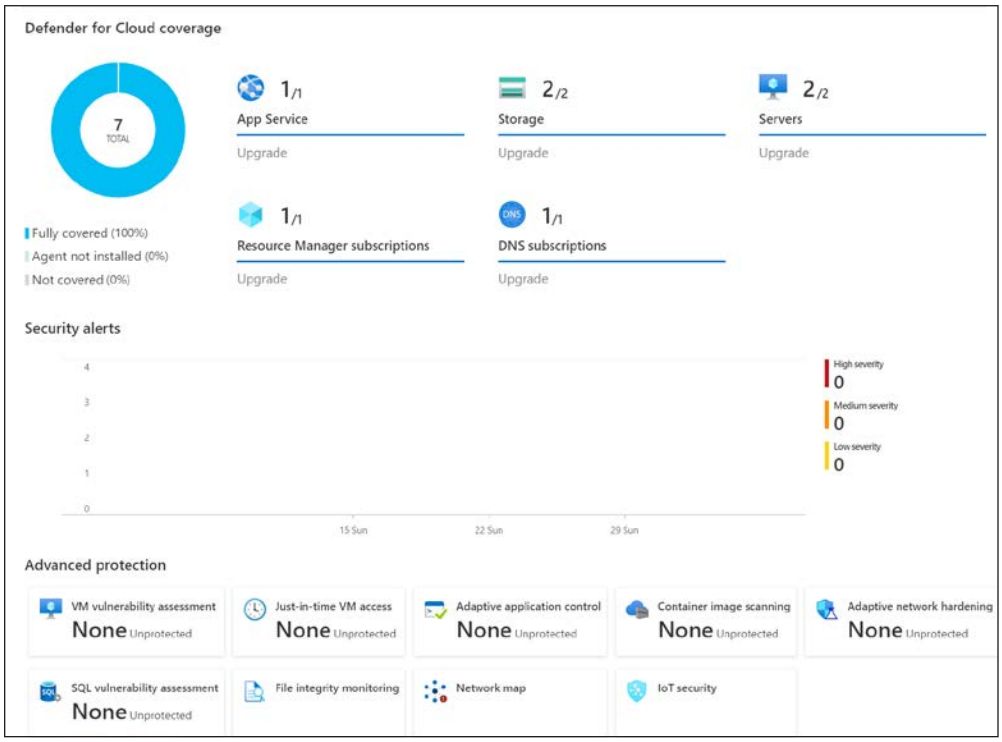


FIGURE 2-61 Workload protection in Defender for Cloud

Finally, clicking on **Firewall Manager** will open Firewall Manager and display the status of your networks, as shown in Figure 2-62. This shows you your virtual hubs and your virtual networks.

> ***NOTE*** **VIRTUAL HUB**
>
> A *virtual hub* refers to a Microsoft-managed virtual network that is a component of another service called Azure Virtual WAN. This is out of scope for this book.

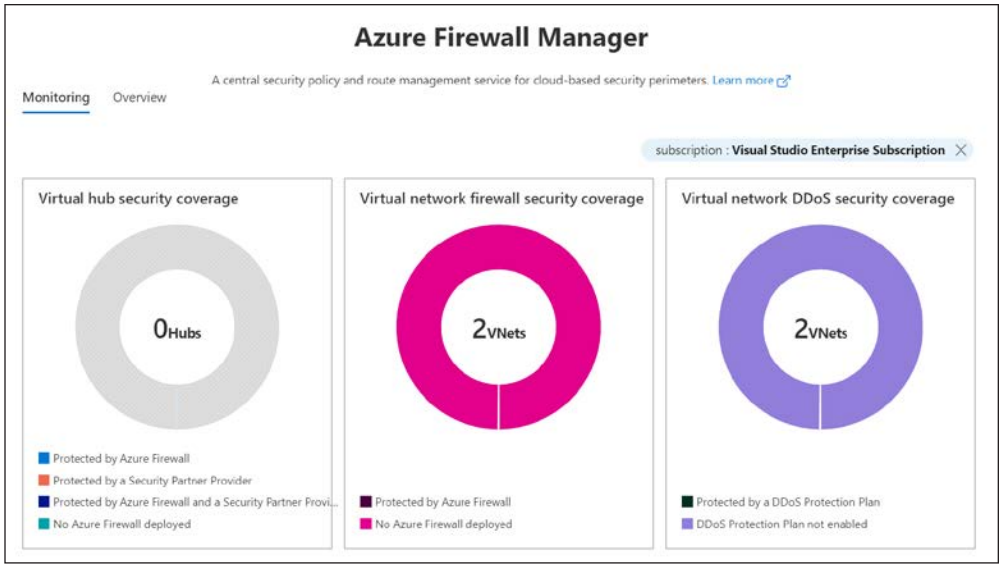


FIGURE 2-62 Azure Firewall Manager

As I said earlier, Defender for Cloud can also monitor resources on other cloud providers. In addition to that, you can add non-Azure servers from your on-premises environment.

To add another cloud provider, click on **Environment Settings** in Defender for Cloud and then click **Add Environment**, as shown in Figure 2-63. Select your cloud provider from the list to add your resources from that platform.

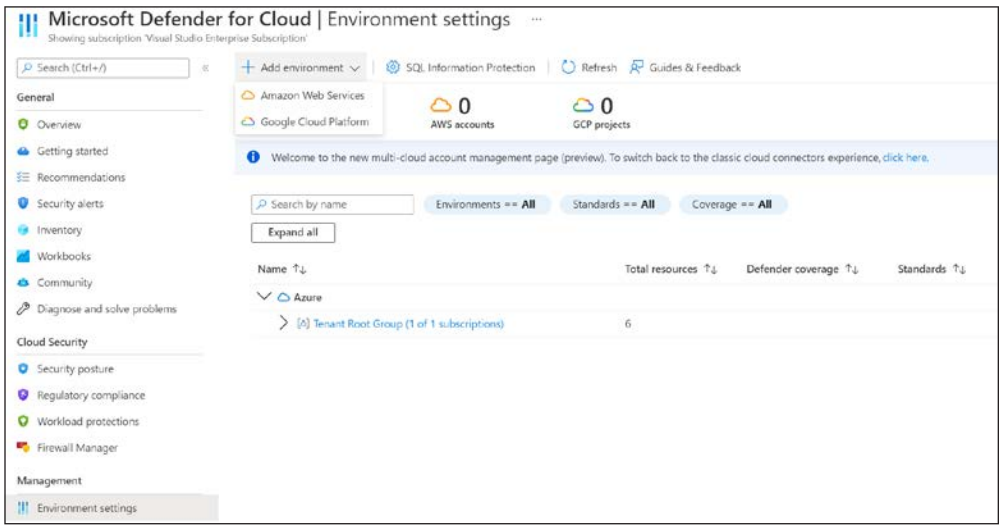


FIGURE 2-63 Adding a new cloud environment

To add an on-premises server, click **Inventory**, and then click **Add Non-Azure Servers**, as shown in Figure 2-64. This will give you directions on how to add your server to Defender for Cloud.

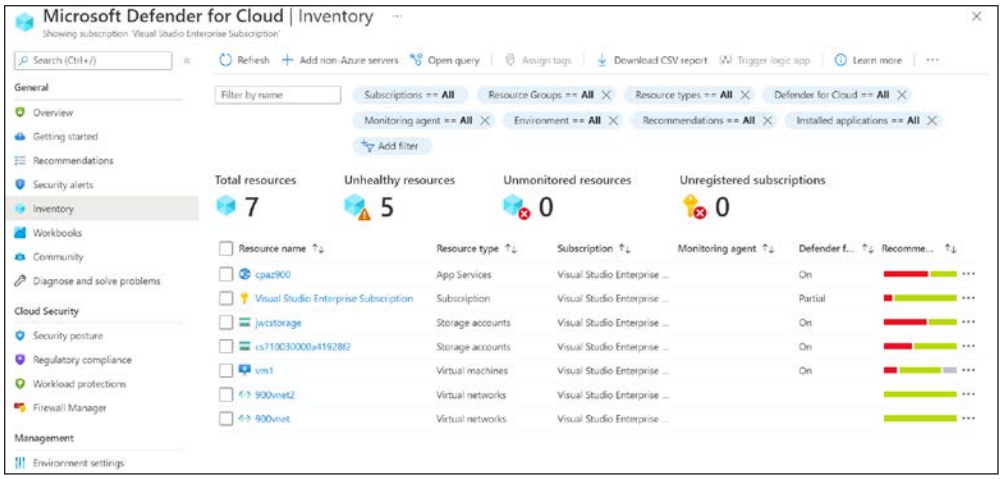


FIGURE 2-64 Inventory view and adding a non-Azure server

Thought experiment

We've covered a lot in this chapter. Now it's time to put your knowledge to the test in a thought experiment. The answers to this thought experiment are in the section that follows.

Governing VMs and keeping them available

Contoso Medical Group (CMG) has taken the advice you gave earlier and has moved its apps to the cloud. They're preparing to deploy a new app, and that app will use several VMs. The director responsible for that app needs assurance that the VMs are highly available and fault tolerant. The director's requirement is that they obtain an SLA from Microsoft of 99.99 percent.

It's also important to this director that the VMs are segmented from other parts of the organization when it comes to management and billing. For this reason, the director has acquired two Azure subscriptions, one for the application team and another for the IT team managing the application. It's important to be able to manage both of those subscriptions as a single unit because governance will need to be applied to both.

How can you ensure the director that the 99.99 percent SLA is met on the VMs for this app? What recommendation would you make to ensure the requirement for subscription management is met?

Minimizing expenses for small workloads

One of the developers of the application has also asked for your help. The developer has some quick jobs that the application will need to perform regularly, and their team has built a Docker image that can run those workloads. The director's budget for the application is very tight, and they're worried that they'll go way over budget by allocating an additional VM to run these workloads.

What would you recommend to the developer to minimize the expense of running their workloads?

Fault-tolerant VMs

Another component of the application will expose a website that operates a critical component to the field. It's important that users in the field always have access to this component. The director is concerned that if the VM ever needs to reboot for an update, it will be unavailable for a period. Also, the director is adamant that this component be fault-tolerant as well.

What recommendation would you make to ensure the website is protected from a reboot and remains fault tolerant?

Cost-effective app usage

In another department at CMG, a manager is rolling out a reporting app that uses Microsoft Excel to analyze data. The manager has about 200 employees who will be using this app, and the company doesn't have the budget to cover 200 licenses for Windows and Excel. The manager's employees in the field already have company-issued Android tablets, and they'd prefer to use what they already have. It's also important to the manager that employees do not have copies of spreadsheets stored on their tablets in case an unauthorized person were to gain access.

What service would you recommend to the manager, and why?

Fast and easy web app management

The sales manager at CMG is in the process of launching a new product, and they want to promote the product using a website. The sales manager has already gotten the domain name purchased for the site, and the development team has written a web app in PHP that they'll be using. Parts of the web app will require authentication, and the sales manager would like people to be able to use a social media account to authenticate for the sake of convenience. The sales manager needs to get the site up as soon as possible, and they also need to minimize management costs.

What service would you recommend to the sales manager, and why?

Connecting VNets and DNS management

The IT director is working with the developers on the sales manager's application. One of the back-end components has a complex network architecture involving two different virtual networks. The IT director needs to figure out a way to allow communication between VMs on these two networks, and the problem is further complicated by the fact that these networks are in different Azure regions. Additionally, because of regulations, the data must be encrypted when on the wire.

The IT director also needs to find a way to create and manage DNS records for the VMs in these networks. It's important to the director that they be able to use friendly DNS names for these VMs.

What would you recommend to the IT director, and why?

Storage migration, security, and governance

One of the application developers has asked for your help in data storage for the web app. They are going to have some video files and documents that need to be stored securely. They also need to be certain that the data always remain within the country of origin. Finally, the developer wants to ensure that if there's a problem in an Azure datacenter where the data is stored, users can still access the data.

What recommendation would you make to the developer, and why?

When CMG deploys the application for the first time, they'll need to move the records the app will use from their on-premises server to Azure Storage. They have 70 TB of data that they'll need to move, and they need to do it quickly and securely.

How would you recommend they move the data to Azure?

Effective and secure collaboration with resources

The CTO of CMG has made an agreement with a partner company to collaborate on some of their ongoing work. This collaboration will involve giving the employees of the partner application access to some of CMG's Azure resources. The CTO is security-focused and wants to make sure they are following the best security guidance, but he also doesn't want to make collaboration complicated for their partners.

What advice would you offer to the CTO?

Thought experiment answers

This section covers answers to the thought experiment.

Governing VMs and keeping them available

The director at CMG needs a 99.99% SLA and a fault tolerant solution for VMs. Also, they need to have a way to manage the two subscriptions the app is using as one unit and to apply governance to both easily.

You recommend to the director that they use at least two VMs deployed to availability zones. This will ensure that they obtain an SLA of 99.99%. To properly manage the subscriptions and apply governance across both, you recommend that the director create a new management group and place both subscriptions into it. Then, the director can apply governance to the management group, and it will apply to both subscriptions.

Minimizing expenses for small workloads

A developer has asked you about running some quick workloads using a Docker image, and they need a cost-effective way to do that. You recommend that the developer run those workloads in Azure Container Instances (ACI). Because ACI is a serverless option, the company won't have to pay for a VM that's allocated to it. The developer can quickly spin up the Docker container and run its workloads, and they'll only pay for the compute resources used.

Fault-tolerant VMs

The CMG director has asked you for advice on a website that will be hosted for the app. The company wants to ensure that the VM hosting the site is fault-tolerant and protected from availability issues caused by a reboot. You recommend to the director that the company deploy multiple VMs into an Availability Set. This will provide multiple fault domains for fault tolerance, and multiple update domains will protect the VMs from availability issues caused by reboots.

Cost-effective app usage

A manager has asked for help on the best way to implement access to Excel for a reporting app. The manager needs users to have access using existing Android tablets, and they cannot have copies of spreadsheets on those tablets. You recommend that the manager use Azure Virtual Desktop (AVD). By installing Excel and the spreadsheets in an AVD tenant, the manager can give their users access using the AVD client for Android. The Excel spreadsheets would then remain on the AVD tenant and would not be downloaded to the Android tablets.

Fast and easy web app management

A sales manager is launching a new product and website written in PHP. They need to use a custom domain for that website, and they want people to authenticate using a social media account. You recommend that the manager host the app in Azure App Service as a web app. This will make it fast and easy to get the site running, and they can add social media authentication quickly and easily while limiting management costs.

Connecting VNets and DNS management

An IT director needs to connect two networks to allow for VMs to communicate. Traffic must be encrypted, and the director needs to use friendly DNS names for the VMs. You recommend that the director use a VNet-to-VNet connection using Azure VPN Gateway. Unlike using VNet peering, VPN Gateway satisfies the requirement that the traffic be encrypted. To allow for friendly DNS names, you recommend that they set up a private DNS zone using Azure DNS.

Storage migration, security, and governance

A developer needs to store some binary file that must be always secure and within the country of origin. The developer wants to ensure that a datacenter problem won't affect availability. You recommend they use Azure Blob storage and enable zone-redundant storage (ZRS). This will ensure that availability zones are used for redundancy, and unlike GZRS, it will ensure the data remains in the country of origin.

CMG needs to migrate about 70 TB of data from their on-premises datacenter to Azure. You recommend that they use Data Box for this. Microsoft will ship them a Data Box device with 80 TB of capacity. The device is secure, tamper-resistant, and meets the needs of migrating data quickly and securely.

Effective and secure collaboration with resources

The CTO wants to enable collaboration with a partner and give them access to some Azure resources, and he wants to do it in a secure and convenient way. You discuss the Zero-trust framework with the CTO, and you recommend that they use the B2B features in Azure AD to enable this collaboration. To ensure security and convenience, you recommend enabling multi-factor authentication and passwordless authentication for users.

Chapter summary

The scope of this chapter was enormous! We started with the basics of Azure architecture and ended up discussing a wide array of Azure services. We covered concepts related to identity, security, and more.

Here's a summary of everything this chapter covered:

- Azure is organized by geographies and regions.
- Regions have a region pair for high availability.
- Sovereign regions account for the requirements of governments and countries.
- Availability zones ensure high availability if a problem is encountered in a datacenter.
- An enabled region has at least three availability zones.
- An Azure datacenter is a physical building in a region with its own power, water, cooling, and network system.
- A resource group is a logical entity for organizing Azure resources.
- An Azure subscription is used to give you access to create and manage Azure resources.
- A management group is a logical entity for managing subscriptions. A management group can only contain subscriptions and other management groups.
- Azure Container Instances (ACI) is a serverless option for easily running containerized workloads.
- Azure Functions is a service for hosting microservices in the cloud.
- Azure offers the ability to create VMs easily with a choice of multiple operating systems.
- Availability sets provide a virtual representation of a network rack (a fault domain) and an update domain. The fault domain protects against a fault in the rack, and the update domain ensures availability during reboots from updates.
- Virtual machine scale sets (VMSS) allow you to scale VMs easily.
- Azure Virtual Desktop (AVD) provides desktop virtualization for running operating systems and apps using a client for Windows, macOS, iOS, or Android. It can also be accessed via a web browser.
- Creating a VM in Azure creates many additional resources under the hood, including network interfaces, virtual networks, and disks.
- Azure App Service is a PaaS service for easily hosting web apps in the cloud.
- App Service runs web apps in App Service plans.
- Azure Kubernetes Service (AKS) is a PaaS offering for hosting Kubernetes clusters running containerized applications.
- Azure Spring Cloud makes hosting Spring apps in the cloud easy.
- Azure virtual networks (VNets) are used to create network environments in the cloud.

- VNet peering can be used to connect two VNets to each other. Global VNet peering connects VNets in separate regions.
- Azure DNS allows for the management of DNS records in the cloud. Private DNS zones are for resources in a VNet. Public DNS zones are for internet-facing resources.
- Azure VPN Gateway connects your VNet to other networks securely.
- Azure ExpressRoute connects networks to your Azure VNets using a secure connection called a circuit. ExpressRoute can transfer at speeds up to 10 Gbps.
- Azure Blob storage is for storing binary files in Blobs. Blob storage can be organized using containers.
- Azure Disks stores disks used in VMs.
- Azure Files is a cloud-based SMB file share. Azure File Sync can synchronize an Azure Files share with an on-premises server.
- Azure Blob storage offers three tiers. The Hot tier, Cool tier, and Archive tier.
- Locally redundant storage (LRS) is for data that can be easily recreated. It stores three copies of data in the same datacenter.
- Zone redundant storage (ZRS) stores data in multiple datacenters using availability zones.
- Geo-redundant storage (GRS) is the same as LRS, but an additional three copies of your data are stored in a second region.
- Geo-zone-redundant storage (GZRS) is like ZRS, but it creates three additional copies of your data in a second region using LRS.
- A general-purpose v2 storage account (Standard tier) is recommended for most users and supports all storage services.
- Block Blobs, file shares, and page Blobs storage account types (Premium tier) are for the highest level of performance in specific storage scenarios.
- AzCopy is a command-line utility for moving files to and from Azure Storage.
- Azure Storage Explorer is a cross-platform application for moving files to and from Azure Storage.
- Azure Migrate is a service that can discover, assess, and migrate workloads to Azure.
- Azure Data Box is a service for offline data migration to Azure Storage. It's offered as Data Box Disk, Data Box, and Data Box Heavy.
- Azure Active Directory is a cloud-based identity service. You can use Azure AD to give users access to Azure resources.
- Azure Active Directory Domain Services (Azure AD DS) connects on-premises Windows domains to Azure AD.
- Single sign-on (SSO) allows users to use resources such as SharePoint and Microsoft 365 without entering a password.
- Multifactor authentication (MFA) authenticates using a combination of two or more factors using something you know, something you have, and something you are.

- Passwordless authentication allows for authentication using a device or a security key instead of a password.
- Guest users can be added to Azure AD to give them access to your Azure resources.
- External users from other organizations can be given access to your Azure AD so they can collaborate on apps and services.
- Conditional Access allows you to define policies that must be met for people to authenticate to Azure AD.
- Role-based access control (RBAC) allows others a specific level of access to resources based on a role assignment.
- Defense in depth (castle doctrine) is the concept of multiple layers of defense designed to protect from a breach.
- Zero-trust is a framework that applies end-to-end security and the assumption that all access may be a breach of security.
- Microsoft Defender for Cloud is a single solution for managing the security and regulatory compliance of Azure resources, resources in other clouds, and resources on-premises.

CHAPTER 3

Describe Azure management and governance

As you move to the cloud, it's important to plan accordingly. You must consider several factors. Cost is certainly one factor, but how you plan your resource governance is also important, and in most situations, you'll also need to deal with compliance with your corporate policies.

As you deploy your resources, you'll need to understand the options available for managing and deploying your resources. You've had a chance to see the Azure portal already, and now it's time to dig deeper, as well as look at some of the available command-line options.

You'll also need to understand how to monitor your resources, and this chapter will cover the tools that make that possible.

Skills covered in this chapter:

- Describe cost management in Azure
- Describe features and tools in Azure for governance and compliance
- Describe features and tools for managing and deploying Azure resources
- Describe monitoring tools in Azure

Skill 3.1: Describe cost management in Azure

Managing your costs in the cloud requires a different approach than when on-premises. In the cloud, there are factors that can dramatically impact your costs, and you'll need to be aware of those factors. You may also need to accurately track the spending of specific areas of your organization or within specific deployments.

Azure provides information and tools that address all these needs.

This section covers:

- Factors that can affect costs
- Pricing calculator and Total Cost of Ownership (TCO) calculator
- Azure Cost Management and Billing
- Tags

Factors that can affect costs

As you're planning your Azure deployments, you should keep in mind the factors that can affect your costs, such as the resource type, how you purchase the resource, the Azure regions you use, and the billing zone your resources are in.

Azure services are billed according to *meters* associated with a resource. These meters track how much a specific metric has been used by the resource. For example, there is no charge specifically for an Azure virtual network, and you aren't charged for network traffic within a virtual network; however, you are charged per gigabyte for traffic into and out of the virtual network from peered virtual networks.

EXAM TIP

Each Azure service has a pricing page that outlines estimates on pricing for that resource based on typical usage.

As you determine which resources you need to use in your Azure deployment, consider how those resources are going to use the metrics the resources charges for. For example, if you can plan your virtual networks so that you have fewer peered networks, you can save substantially over the long term.

Also, there are ways to save money when purchasing Azure resources. For example, Microsoft will offer you a reduced rate if you agree to pay in advance using an Enterprise Agreement. Longer-term agreements offer even more price breaks. Cloud Solution Providers (CSPs) might also provide you with complete solutions that are more cost-effective than purchasing all the resources yourself.

Microsoft's costs for operating Azure services differ by region, even when those regions are within the same geographic boundary. Therefore, your pricing will differ based on which Azure region you use. For example, a VM deployed to the Central US region will cost more than the same VM deployed to the East US region. Microsoft doesn't provide a breakdown of its costs, but you can assume that electricity and other resources needed for an Azure data center are more expensive in the Central US region than they are in the Eastern US region.

EXAM TIP

Choosing the least-expensive region for each of your Azure resources usually isn't a good way to control costs. You might have to pay for network traffic across regions, which could increase your costs above the amount you're saving. Many Azure resources do not charge for network traffic within the same region, but they will charge for traffic across regions.

It's also important to keep in mind that you're not charged for network traffic into an Azure datacenter, but you are charged for network traffic out of a datacenter. However, your first 100 GB of outbound data is free. After that point, you are charged a set amount on a tiered model.

MORE INFO **PRICING OF NETWORK BANDWIDTH**

For more information on pricing of network bandwidth in Azure, see *https://bit.ly/az900-bandwidthpricing*.

It's also important to know that Azure regions are broken out into four separate groups for billing purposes. These groups are called *billing zones*, or more commonly, *zones*. Microsoft's costs for network traffic out of each zone differ, so your costs will differ, too.

Reducing Azure costs

It's important to understand how your decisions can affect your costs, but it's also important to understand those things you can do to reduce the costs you experience in Azure.

If you regularly find that you use VMs month over month, you can save substantially by using Azure Reservations. By committing to resource usage in advance and taking advantage of reserved instances, you no longer must pay the fees associated with pay-as-you-go pricing. You can also modify your reserved instance reservation to adjust for your usage as you go, so you don't have to sacrifice flexibility.

Similarly, if you have a need for consistent usage of Azure SQL Database, Azure Cosmos DB, or Azure Synapse Analytics, you can save substantially with reserved capacity pricing on those services. Reserved capacity allows you to save by purchasing a one-year or three-year contract, and you are then billed monthly at a reduced amount.

You can save even more on VMs and Azure SQL Database by taking advantage of hybrid use benefit. When you use Azure Hybrid Benefit, you bring your own license for Windows Server or SQL Server to Azure. For example, if you use 10 Windows VMs for, say, a month, you can save more than 40 percent by using your own license with Azure Hybrid Benefit.

Another way to save money is by taking advantage of Microsoft's unused server capacity to run workloads that only require the use of a computer for a temporary (usually short) timeframe. By using Azure Spot VMs, you can take advantage of this unused capacity for huge savings.

As you can see, many factors can affect your costs in Azure, and it can be difficult to estimate costs based on all these factors. Fortunately, Microsoft offers calculators that can help you get a handle on estimating your costs as you move to the cloud.

Pricing calculator and Total Cost of Ownership (TCO) calculator

Azure provides a couple of calculators to help you with cost management. The Azure pricing calculator can help you get an estimate of expenses based on the products you intend to use, as well as where those products will be deployed, and so on. The Total Cost of Ownership (TCO) calculator can help you forecast your savings by moving from on-premises to the cloud.

You can access the pricing calculator by browsing to *https://bit.ly/az900-pricingcalculator*. You can then use the pricing calculator to create an estimate of your Azure expenses.

When calculating an estimate of your Azure expenses, the first step is to select which products you want to use. As shown in Figure 3-1, some of the more common Azure products are displayed by default, and you can add any of those products by clicking its tile.

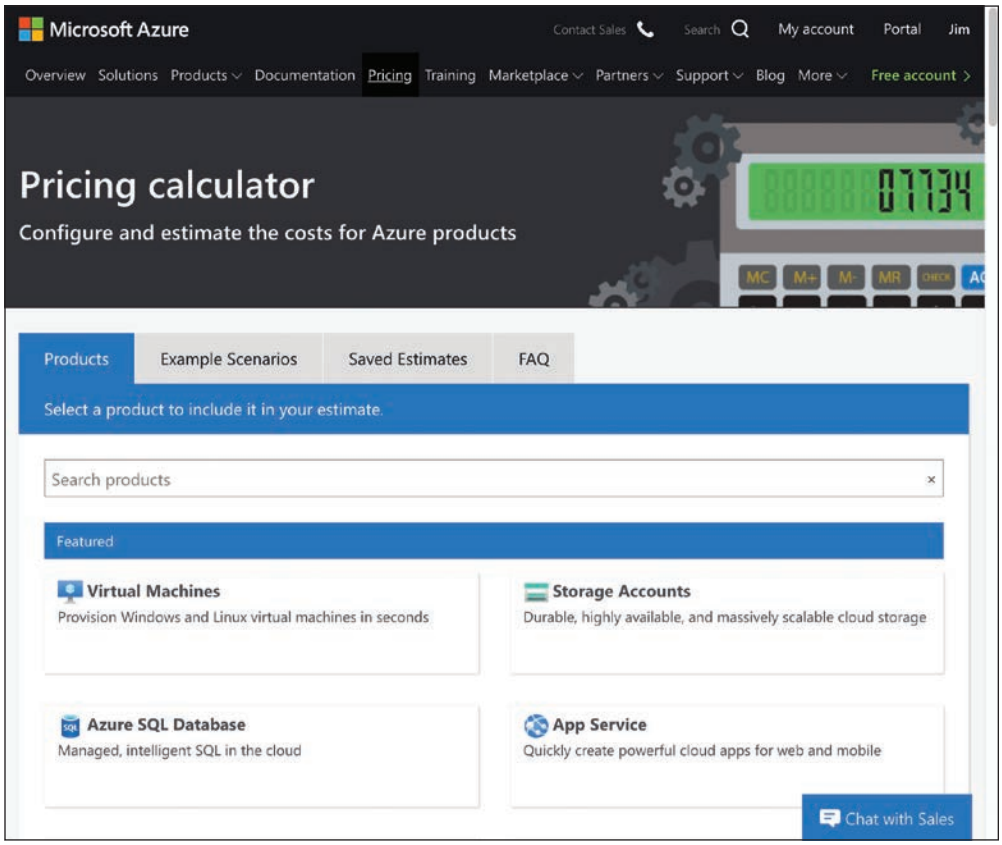


FIGURE 3-1 The pricing calculator

If the product you want is not listed, you can search for your product by entering its name in the Search Products box.

After you add the products you want to use, scroll down to configure the specific details of each service. These details vary based on how Microsoft charges for the product. Figure 3-2 shows the options for Azure SQL Database.

Clicking the informational icon to the right of the product name displays a menu for quick access to the pricing page for the product, additional product details, and documentation to help you make better decisions about the options you select.

Once you've configured a product based on your needs, you can add another instance of that product to your estimate by clicking the **+** button (**Clone** button) in the upper-right portion of the window. For example, suppose you need two Azure SQL Databases for your app, and each of them is going to be using the same service tier, instance size, and so on. The easiest way to add these is to add one Azure SQL Database product to your estimate, configure it with the desired pricing options, and then click the **Clone** button to add the second instance.

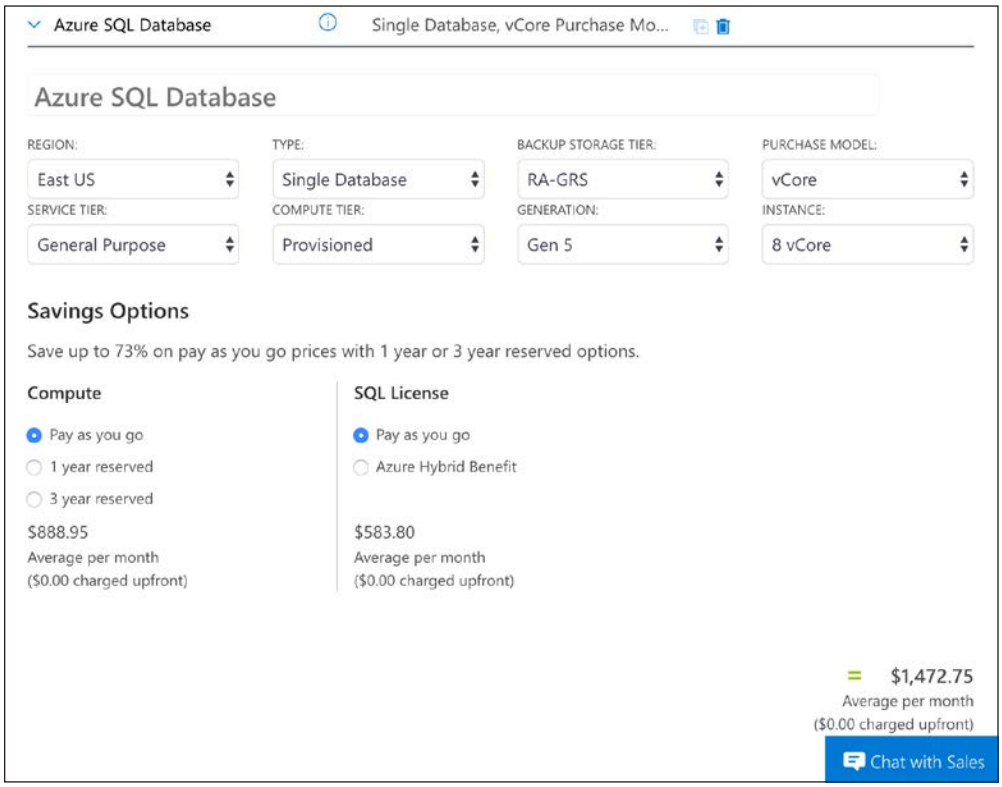


FIGURE 3-2 Estimate of costs for Azure SQL Database

To review your pricing estimate, scroll to the bottom of the page. As shown in Figure 3-3, you can choose a support plan to add to your estimate. If you have a Microsoft Online Services Agreement, an Enterprise Agreement, or a Microsoft Customer Agreement, you can choose to have that pricing applied to your estimate. You can then click **Export** to save your estimate as an Excel file and then select **Save** to save your estimate in the pricing calculator to make changes later. You can also click **Share** to create a sharable link to your estimate so others can view it.

> ***NOTE*** **SAVED ESTIMATES**
>
> **If you save an estimate in the pricing calculator, you can access it later by clicking the Saved Estimates tab at the top of the page.**

FIGURE 3-3 Completing an estimate in the pricing calculator

Total Cost of Ownership calculator

The pricing calculator is helpful for estimating your expenses for new applications in Azure, but if you have on-premises applications you want to migrate to Azure and you want an estimate of how much you can save in Azure, the TCO calculator is a better choice. You can access the TCO calculator by browsing to *https://bit.ly/az900-tcocalculator*.

When using the TCO calculator, the first step is to add details about your on-premises servers, databases, storage, and network usage. In Figure 3-4, an on-premises server has been configured for a web app. You can configure all the details about the server, including the OS, whether it's a VM or a physical server, and more.

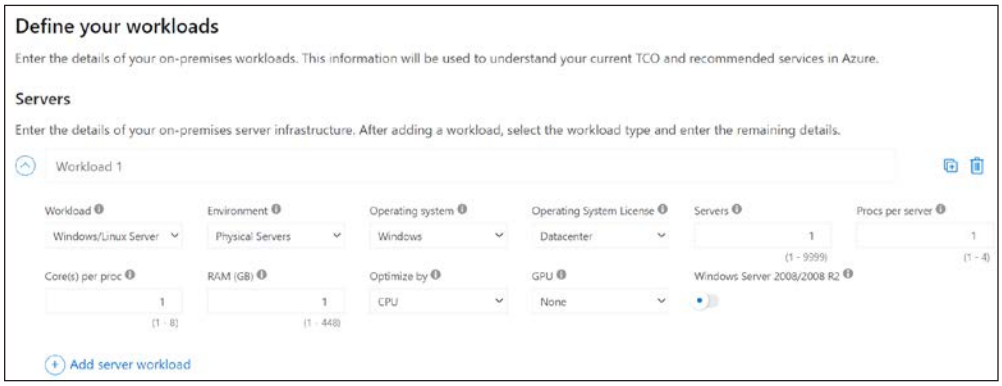


FIGURE 3-4 Configuring an on-premises server in the TCO calculator

On-premises databases and storage systems also should be added, in addition to any network usage for your application. In Figure 3-5, a storage system has been added, and network usage for the app has been specified.

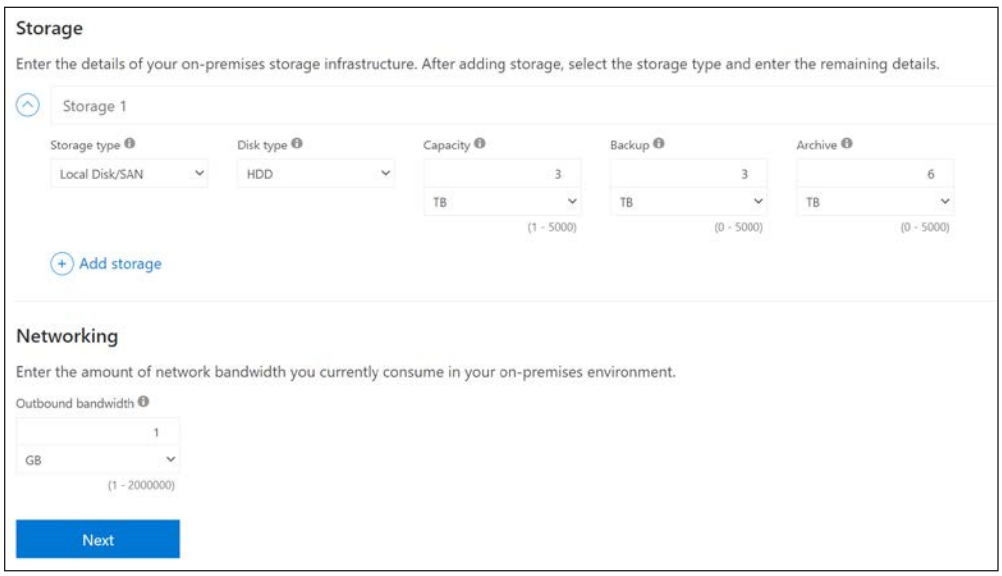


FIGURE 3-5 Configuring storage and networking

After entering all your on-premises workloads, you can view the assumptions the TCO calculator uses by clicking **Next**. The TCO calculator uses a comprehensive list of on-premises expense assumptions that Microsoft has put together based on years of experience, and these assumptions are used to provide you with the best estimate possible of your cost savings. As shown in Figure 3-6, assumptions include items such as whether you've purchased a Software Assurance plan for your on-premises servers, details on your current expenses on-premises, your IT labor costs, and much more. For an accurate TCO estimate, it's best to carefully record your expenses before generating a TCO report.

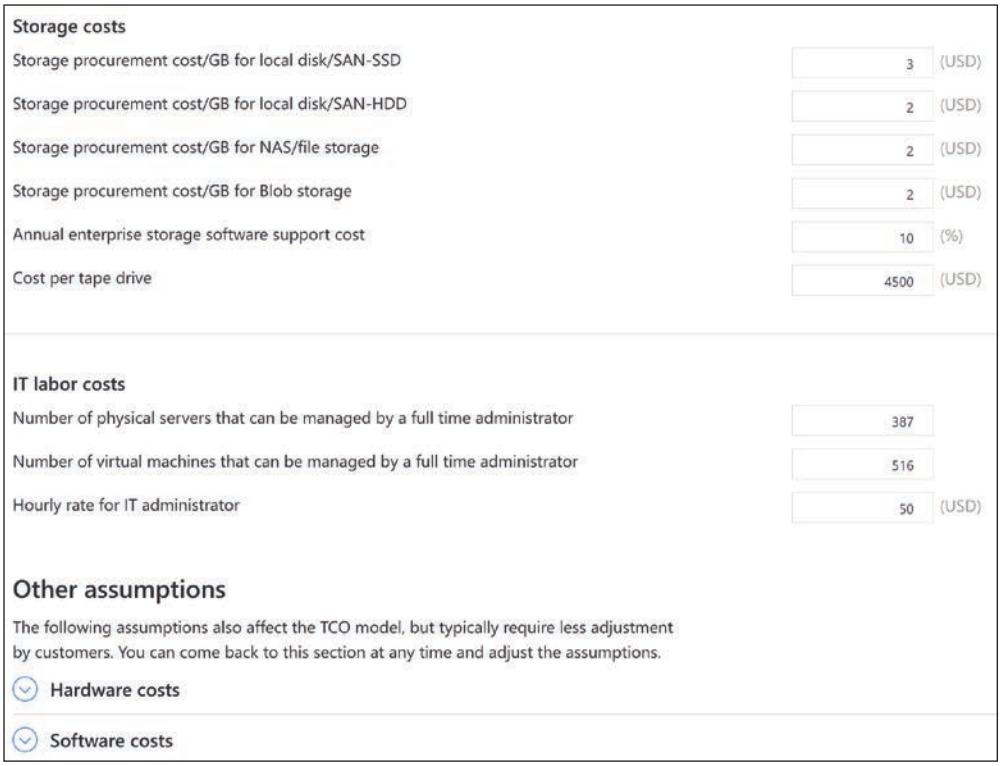


FIGURE 3-6 Adjusting assumptions made by the TCO calculator

After adjusting your assumptions, scroll to the bottom of the screen, and click **Next** to view your TCO report. Your TCO report shows you how much you can save over the next five years by moving your app to Azure, as shown in Figure 3-7.

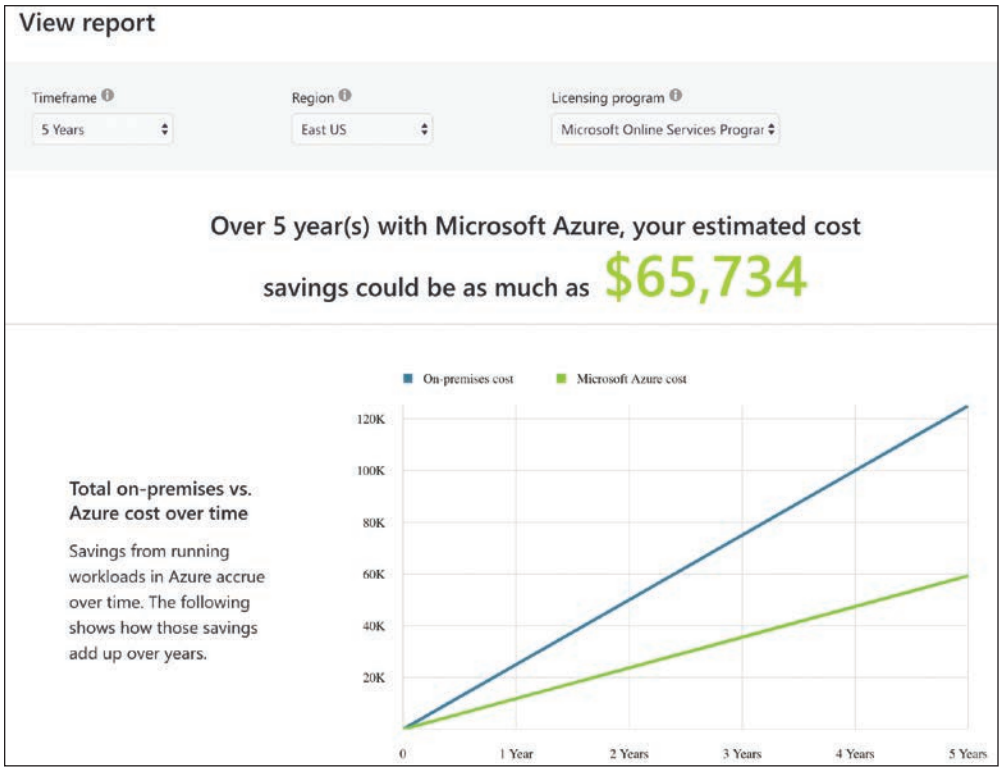


FIGURE 3-7 TCO savings report

A TCO report includes detailed charts of expense savings, and at the bottom of the report, you'll find a breakdown of on-premises costs and Azure costs, so you can easily determine where you'll save money. Just as with the pricing calculator, reports generated by the TCO calculator can be downloaded, saved, and copied by clicking the appropriate button, as shown in Figure 3-8.

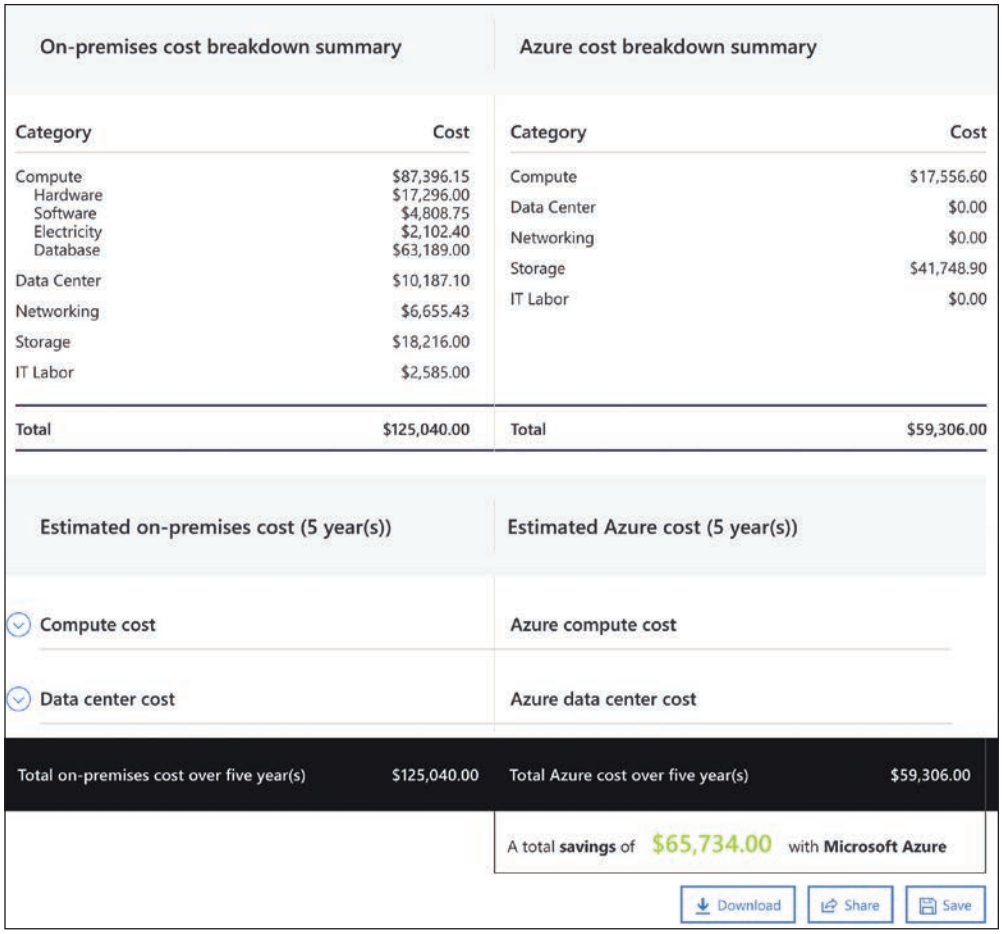


FIGURE 3-8 Summary of costs on-premises and on Azure

Azure Cost Management and Billing

Azure Cost Management and Billing is a tool in Azure that makes it easy to analyze your costs at a granular level. Cost Management and Billing allows you to create a budget for your Azure expenses, set configurable alerts so you'll know if you are approaching a budgeted limit, and analyze your costs in detail.

To get started with Cost Management and Billing, open the Azure portal, search for **Cost Management**, and click **Cost Management + Billing**.

Once Cost Management + Billing loads in the portal, click **Cost Management** in the menu on the left (shown in Figure 3-9) to access Cost Management.

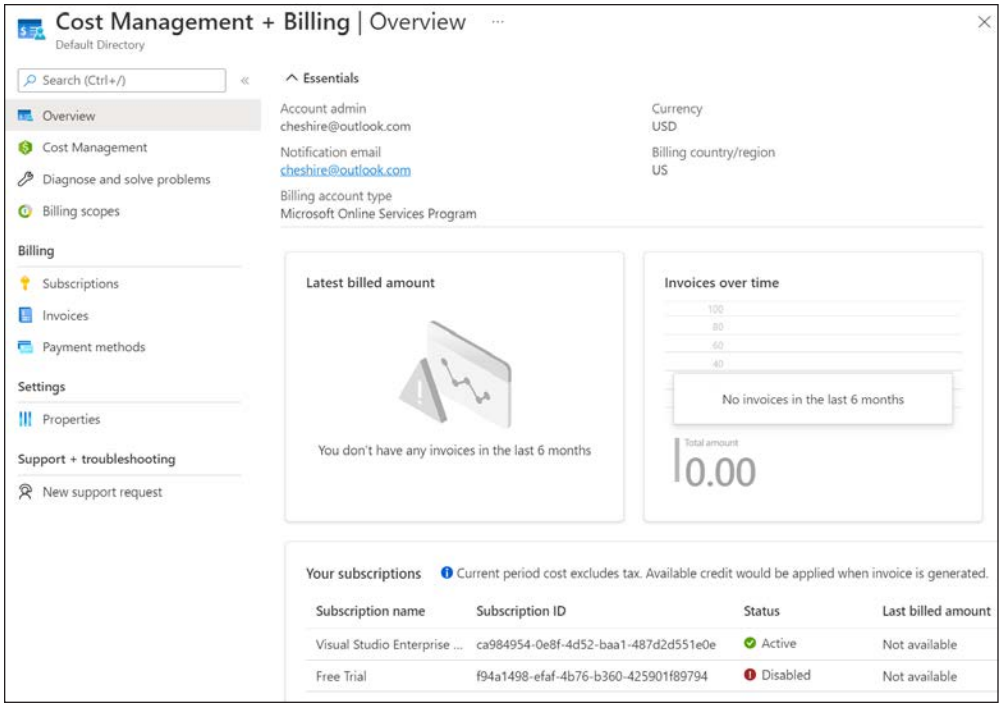


FIGURE 3-9 Cost Management + Billing in the Azure portal

To effectively monitor your costs, you should create a budget in Cost Management. Creating a budget isn't required, but it will allow you to visualize your spending compared to your planned expenses.

1. Click **Budgets** on the left menu, and click **Add**, as shown in Figure 3-10.

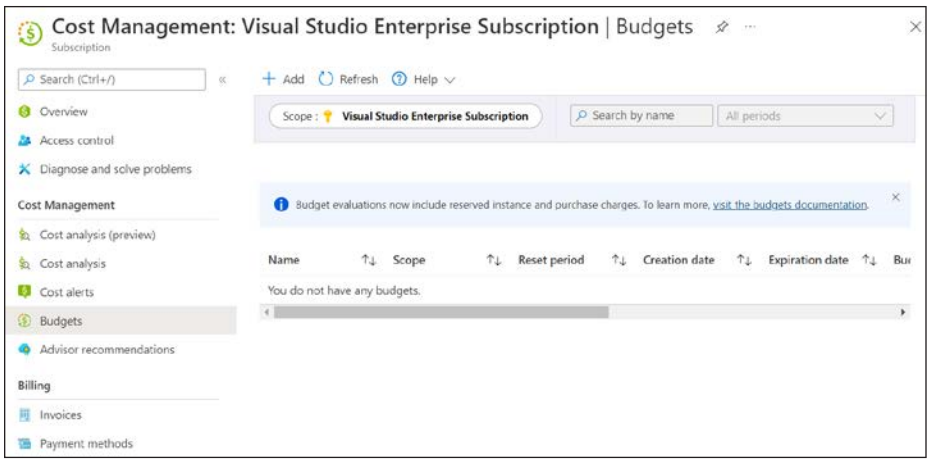


FIGURE 3-10 Adding a new budget

2. Enter a Name for your budget.
3. In the Amount field, enter a spending amount and the period at which your spending resets.
4. Enter a Create Date for your budget.
5. Enter an Expiration Date, as shown in Figure 3-11.
6. Click **Next** to complete your budget.

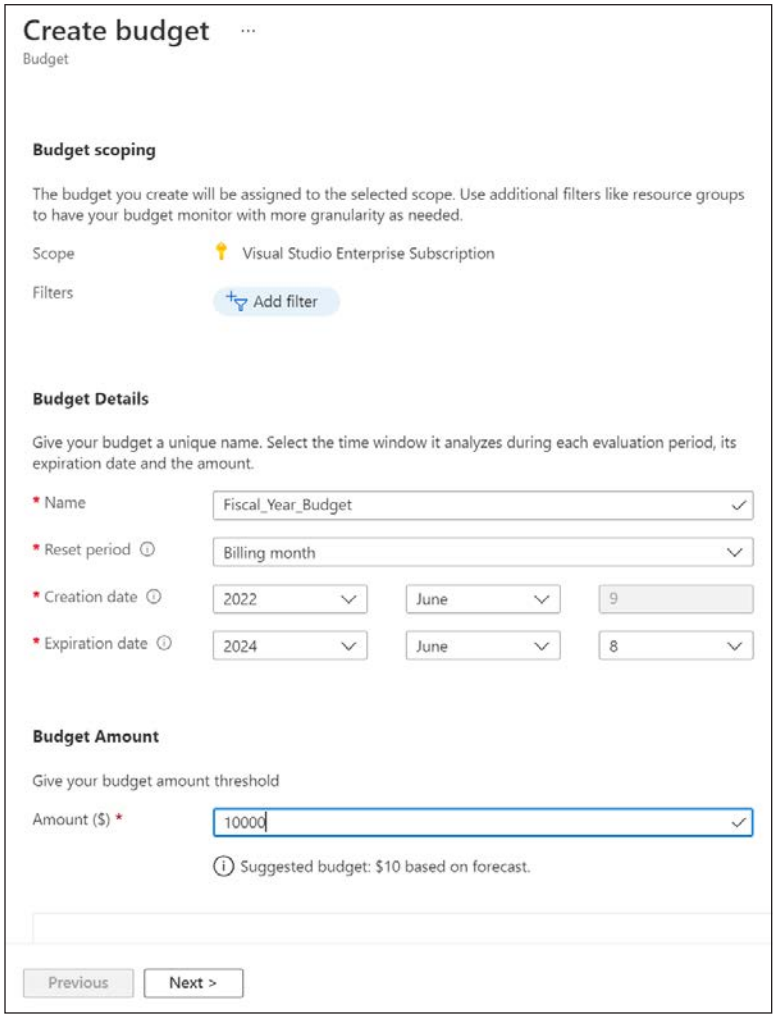


FIGURE 3-11 Creating a budget

7. Configure your Alert Conditions.
8. Under Alert Recipients, add an email address for someone who should receive the alerts, as shown in Figure 3-12.

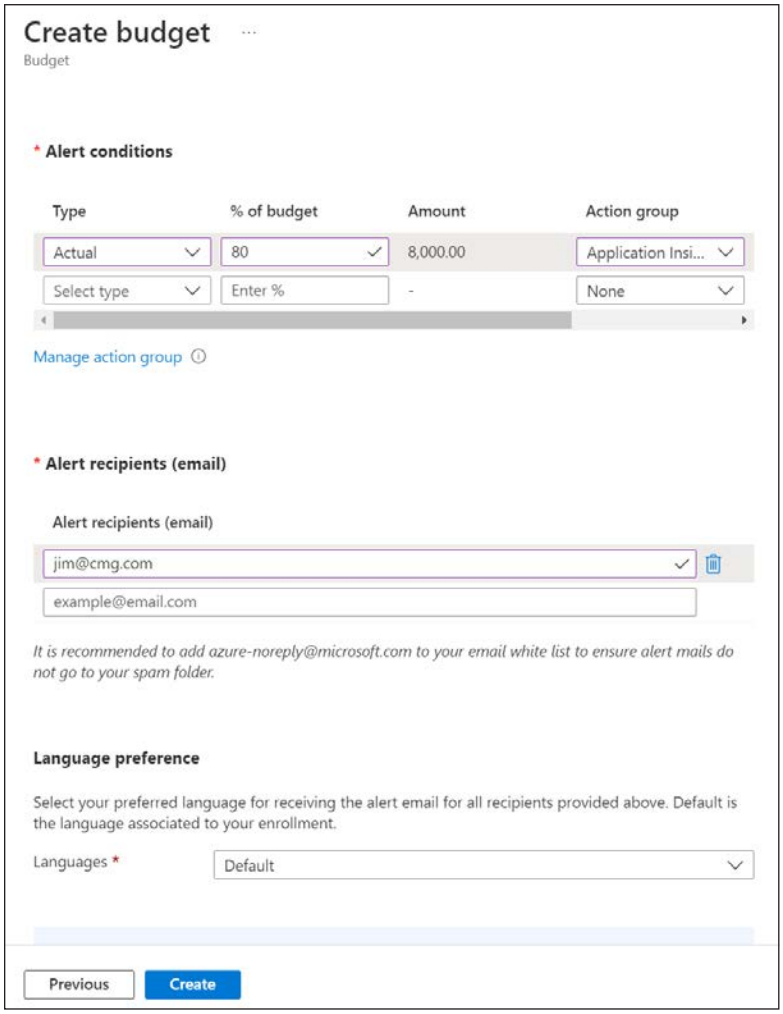


FIGURE 3-12 Configuring alerts for a budget

9. Click **Create** to create your budget.

After you create a budget, click **Cost Analysis** to see how your spending compares to your budget.

Tags

Tags make tracking expenses in Azure easy. A tag consists of a name and a value. For example, suppose a company is participating in two trade events: one in Texas and one in New York. You have also created a lot of Azure resources to support those events. You want to view all the Azure resources for a specific event, but they're spread out across multiple resource groups. By adding a tag to each resource group that identifies the event it's associated with, you can solve this problem.

Once you've added your tags, they will be visible on your Azure invoice, making them an ideal method of categorizing and reporting on costs.

In Figure 3-13, you can see the tags associated with a `VM1` resource. In the Name field, this VM has been assigned a tag named `EventName`, and the Value of that tag is `ContosoTexas`. By clicking the cube icon to the right of the tag, you can view all resources that have that tag.

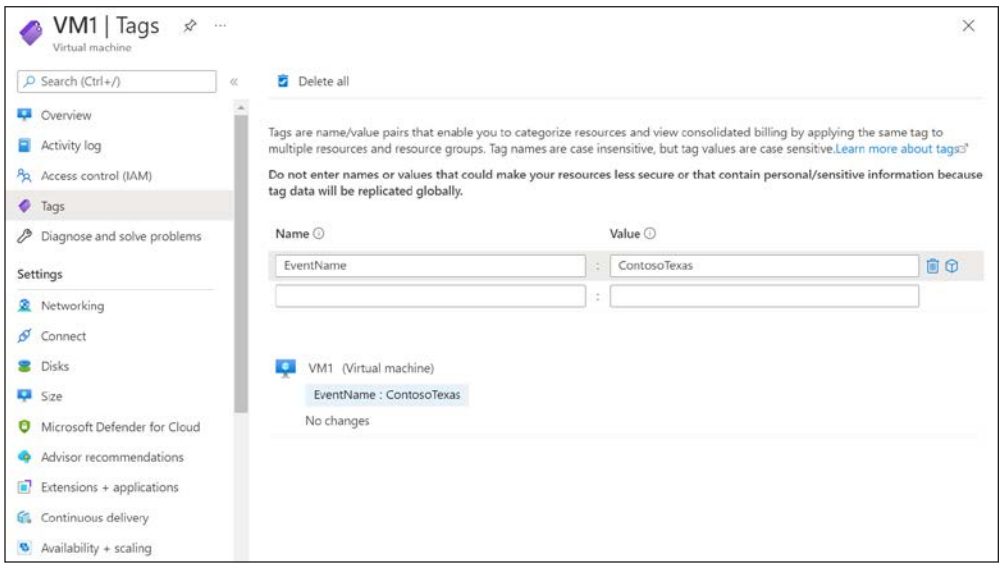


FIGURE 3-13 Tagging a VM

> ***NOTES*** **SHOWING ALL TAGS**
>
> To view all your tags, choose All Services from the main menu in the portal and then search for *Tags* in the list of services.

You can apply a tag to most Azure resources, including resource groups. It's important to understand that by adding a tag to a resource group, you are not adding that tag to the resources within the resource group. Tags add an additional layer of flexibility and power when viewing your Azure resources.

EXAM TIP

When you download your Azure invoice, resource tags will appear in one of the columns. Because Azure invoices can be downloaded as comma-separated values, you can use tools such as Microsoft Excel to filter based on tags.

Skill 3.2: Describe features and tools in Azure for governance and compliance

Unless you have some control over how your resources are created and managed, costs can spiral out of control. In addition to cost control, you might have other restrictions you'd like in place as well, such as which regions certain resources should be created in, how certain resources are tagged, and so on.

The traditional way of handling such governance issues would be to send out a memo to everyone explaining what the requirements are, and then crossing your fingers that people adhere to them. Fortunately, Azure Policy can ensure that your requirements and policies are adhered to.

Ensuring that you can re-create environments is another important aspect of governance, and Azure Blueprints is the perfect way to make sure you can re-create an environment with precision.

Another important facet of the cloud is compliance. Companies moving to the cloud are often concerned about keeping information private and compliant with regulations. Microsoft addresses these concerns using many methods, and in this section, we'll talk about how Azure can help with compliance.

This section covers:

- Azure Blueprints
- Azure Policy
- Resource locks
- Service Trust Portal

Azure Blueprints

When a company decides to create a cloud application, it usually doesn't start by creating a resource in the portal. Instead, a lot of planning happens before a single Azure resource is created. This includes a plan for making sure the entire architecture of the cloud app complies with the necessary standards. It also involves concepts like detailed planning of virtual network topologies and rights assignments for users of the app. Also, best practices are likely a part of this planning.

There's a lot of risk involved if a company fails to plan carefully, and for that reason, many companies will hire someone with deep technical knowledge of the cloud to help in that planning. Hiring that kind of resource can add a lot of additional expense, and it can also add a lot of time to a project.

Azure Blueprints is a service that can make the process of deploying to the cloud easier. Blueprints allow you to configure an environment just as you need it to be, along with all the policies and other governance aspects in place. That configuration can then be saved so it can be duplicated at any time in other deployments.

Items that you add to a blueprint are called *artifacts*. An artifact can be a resource group, an ARM template, a policy assignment, or a role assignment. Once you've created a blueprint, you can either save it in a subscription or management group. A blueprint that's saved in a management group can then be used by any subscription within that management group's hierarchy.

EXAM TIP

Because blueprints are actual Azure resources and not simply files designed to define a deployment, Azure maintains a connection between the blueprint and the resources that use the blueprint. That allows companies to iterate on blueprints and improve them. It also makes it much easier for a blueprint to evolve with a company's needs. Also, blueprints are versioned and can be stored in a source-control system, so tracking blueprints is easy and effective.

With that said, it's important to understand that blueprints aren't a replacement for ARM templates. In fact, most blueprints make extensive use of ARM templates as artifacts. You'll learn about ARM templates in Skill 3.3, "Describe features and tools for managing and deploying Azure resources."

Follow these steps to create a blueprint.

1. Search for **blueprints** in the Azure portal to open the Blueprints | Getting Started page, as shown in Figure 3-14.

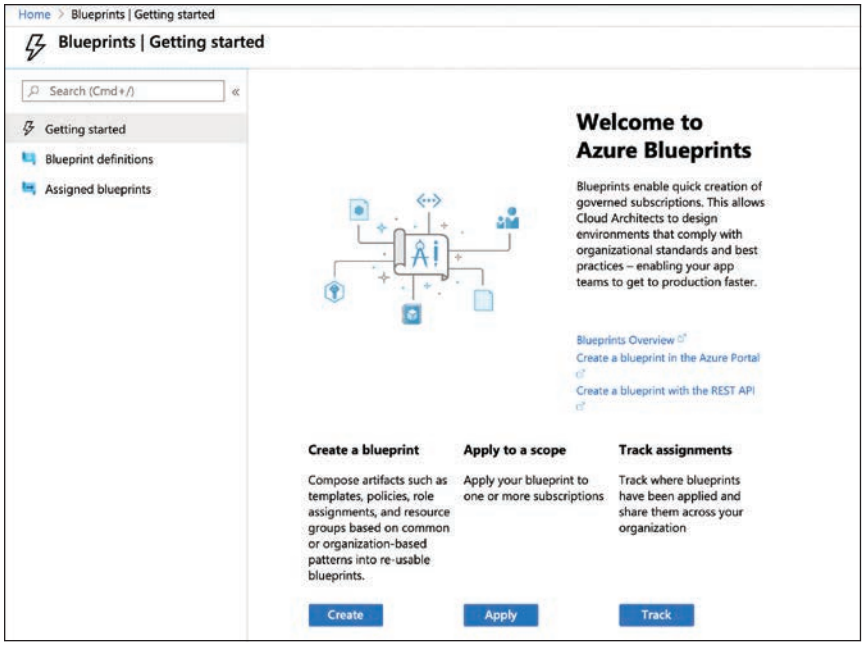


FIGURE 3-14 Azure Blueprints Getting Started page

2. From the **Getting Started** page, click **Create** to start the process of creating a blueprint. As shown in Figure 3-15, Microsoft provides many sample templates that you can use as a foundation for your blueprint, but you can also start with a blank blueprint.

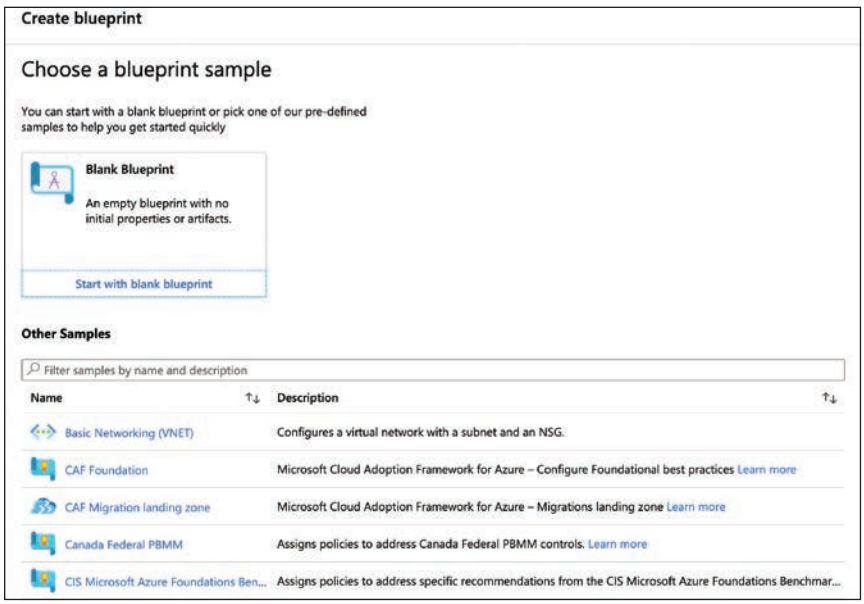


FIGURE 3-15 Creating a blueprint

3. Click the link to start with a blank blueprint. Enter a name for your blueprint, a description, and set the place where your blueprint definition will be saved. This is either a subscription or a management group. In Figure 3-16, a blueprint is being created that will be saved to a subscription.

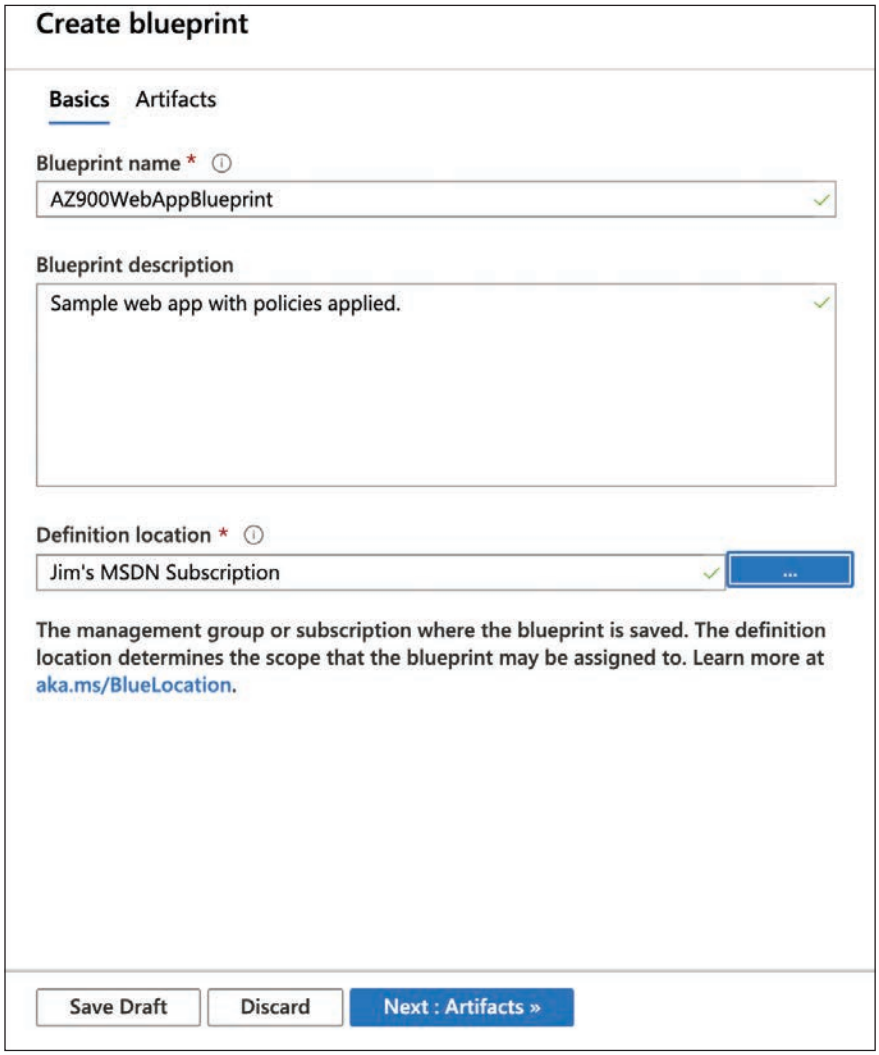


FIGURE 3-16 Specifying a blueprint's basic settings

EXAM TIP

You cannot change the name or the definition location of a blueprint after it's created.

4. To add artifacts to your blueprint, click **Next: Artifacts**, as shown in Figure 3-16. Click **Add Artifact** to add your first artifact. Select the **Artifact Type** and enter the necessary information to add the artifact. In Figure 3-17, a resource group artifact is being added.
5. In order for this blueprint to remain more generic at this point, you can specify that the resource group name and the location are supplied when the blueprint is assigned by clicking the **This Value Should Be Specified When The Blueprint Is Assigned** checkbox next to the **Resource Group Name** and/or **Location**. (We'll cover blueprint assignment later in this section.) Clicking **Add** will add this artifact to the blueprint.

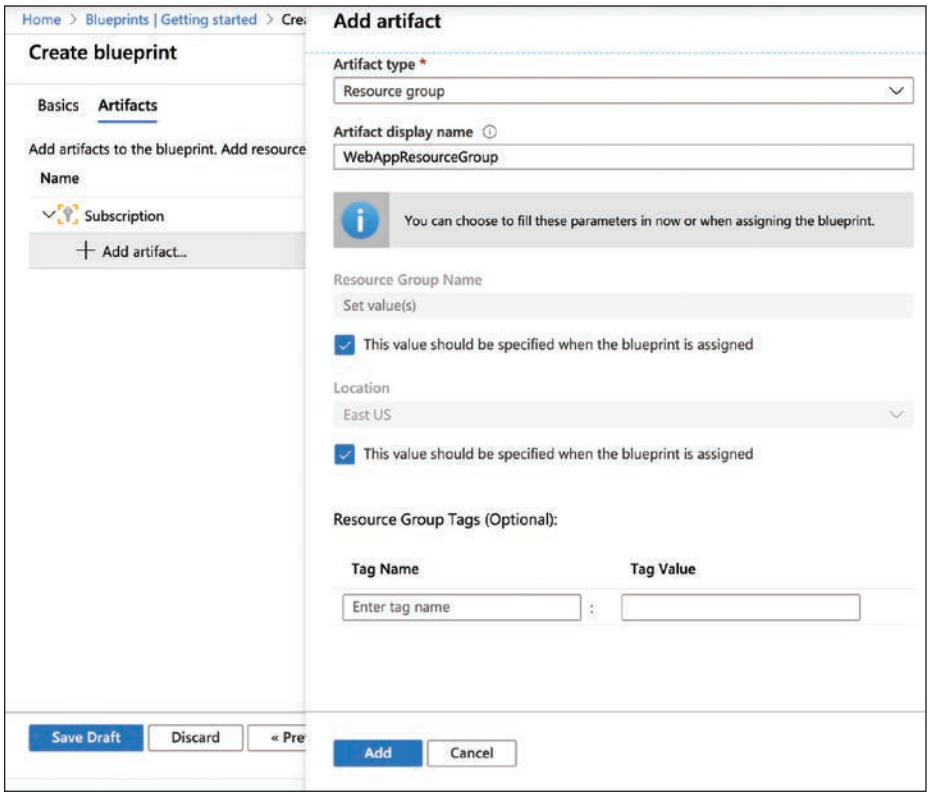


FIGURE 3-17 Adding an artifact

6. When you're finished adding artifacts, click **Save Draft** (shown in Figure 3-17) to save a draft of the blueprint. While in draft mode, the blueprint can be edited and updated. When you're ready to make the blueprint available, you can publish it.

7. To publish a blueprint, click **Blueprint Definitions** and click the blueprint, as shown in Figure 3-18.

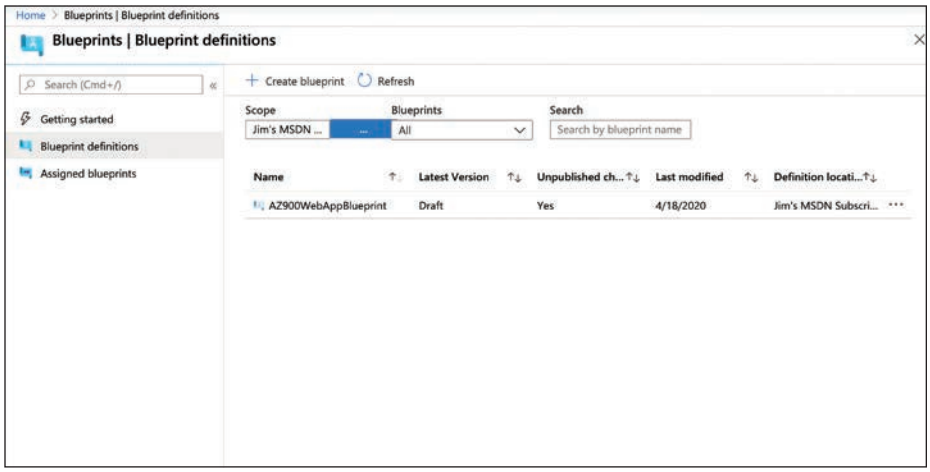


FIGURE 3-18 Viewing blueprint definitions

8. Click the **Publish Blueprint** button, as shown in Figure 3-19, to publish the blueprint.

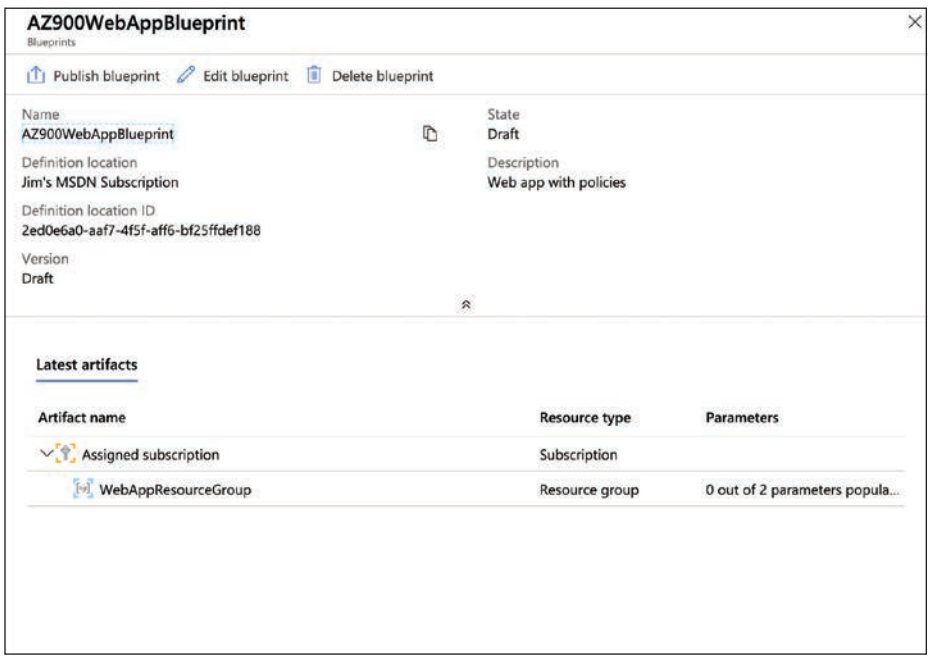


FIGURE 3-19 A blueprint ready to be published

9. When you publish a blueprint, you need to provide the version number. It's advisable to also add change notes. In Figure 3-20, our new blueprint is being published as Version 1.0. Clicking the **Publish** button completes the process.

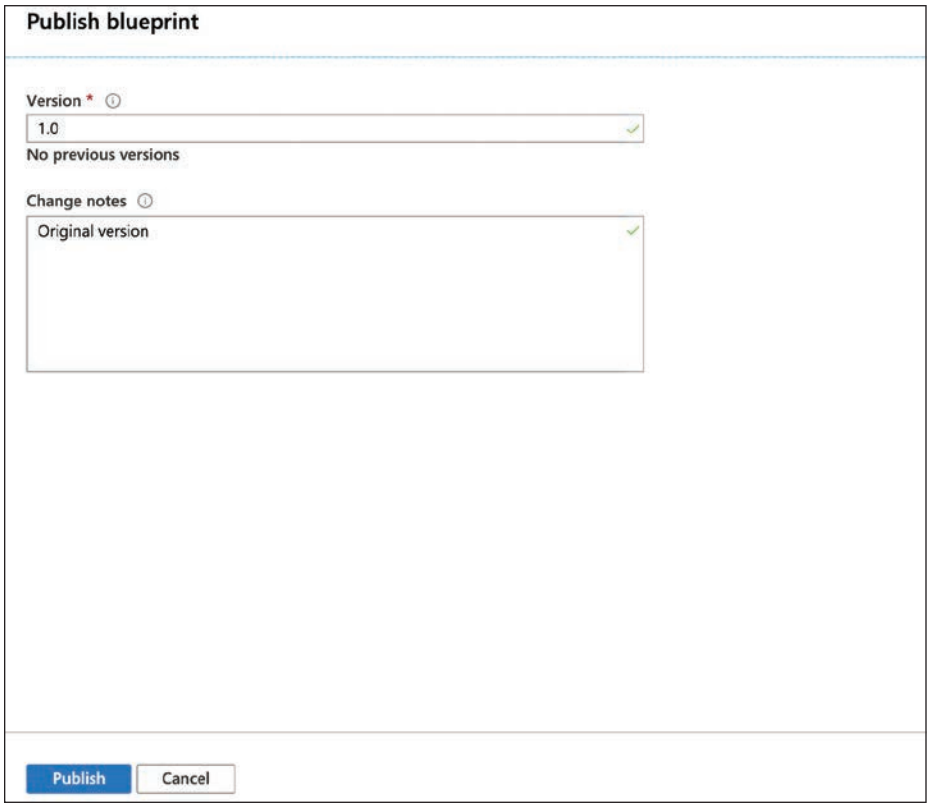


FIGURE 3-20 Publishing a blueprint

10. Once a blueprint has been published, it's available to assign to a subscription. Click the blueprint and click **Assign Blueprint** to assign it, as shown in Figure 3-21.

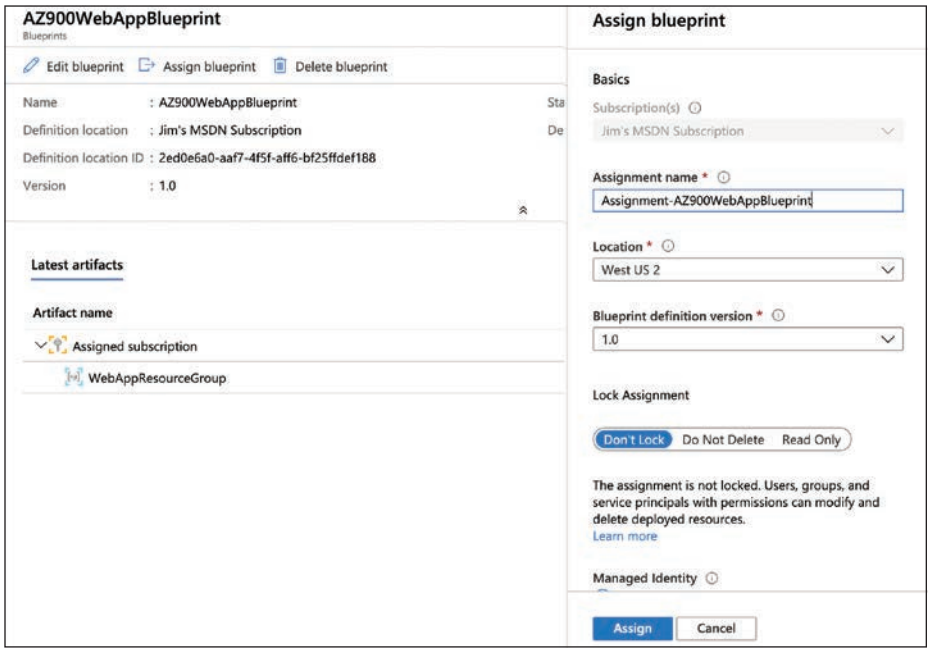


FIGURE 3-21 Assigning a blueprint

11. To assign the blueprint, enter an assignment name, select a location, and select the blueprint definition version. (You can also accept the defaults for all these settings.) You can also specify a lock assignment for the resources the blueprint creates by clicking the desired lock assignment setting.
12. Clicking **Assign** completes the blueprint assignment.

> ***NOTE*** **BLUEPRINT PARAMETERS**
>
> **When this blueprint was created, it was specified that the resource group name and location should be chosen when the blueprint is assigned. Therefore, you will need to enter those values in the spaces provided when you assign the blueprint.**

When the blueprint is assigned to a subscription, resources defined by the blueprint are created in that subscription.

Azure Policy

Azure Policy allows you to define rules that are applied when Azure resources are created and managed. For example, you can create a policy that specifies that only a certain size VM can be created and that the VMs must be created in the South-Central US region. Azure will take care of enforcing this policy so that you remain in accordance with your corporate policies.

To access Azure Policy, type **policy** in the search box in the Azure portal and click **Policy**. Alternatively, you can click **All Services** and search for **policy** in the list. This will display the Policy blade, as shown in Figure 3-22.

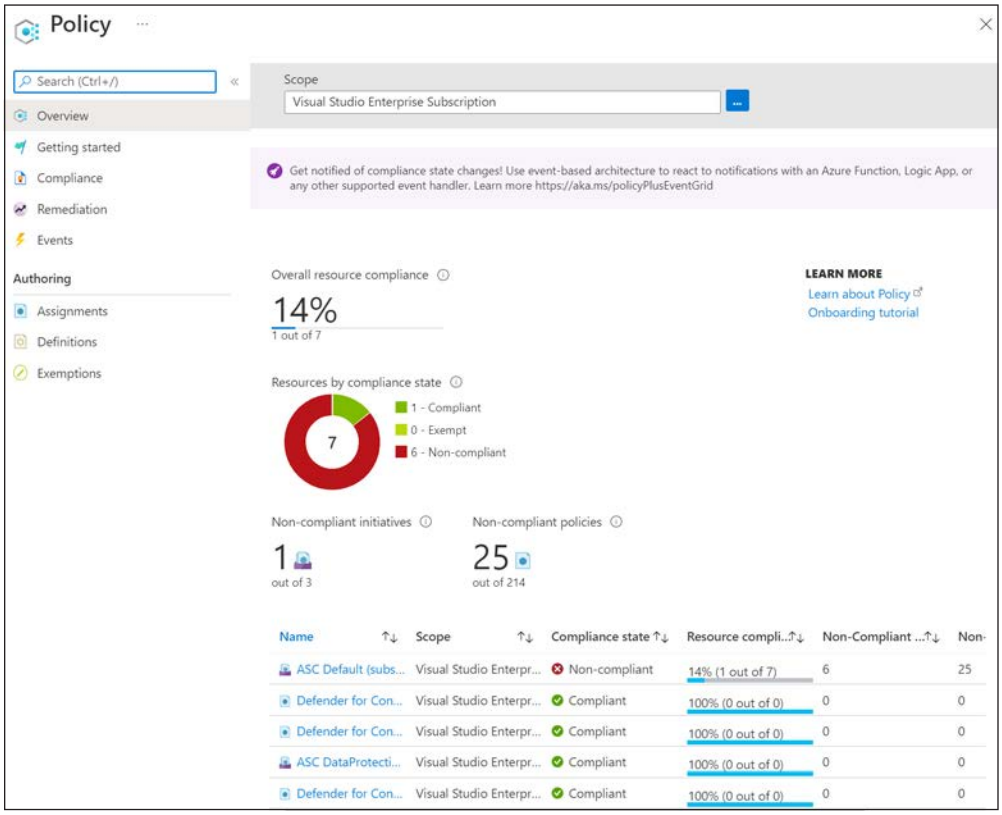


FIGURE 3-22 Azure Policy in the Azure portal

By default, Azure Policy shows your compliance with policies defined on an Azure subscription. If you want to, you can scope this view to a different subscription or to a resource group by clicking the ellipsis (**...**) button next to Scope and selecting the new scope. See Figure 3-23.

The non-compliance shown in Figure 3-22 is based on policies implemented by Azure Security Center. By clicking the non-compliant item, you can see the full details of what is and isn't within policy, as shown in Figure 3-24.

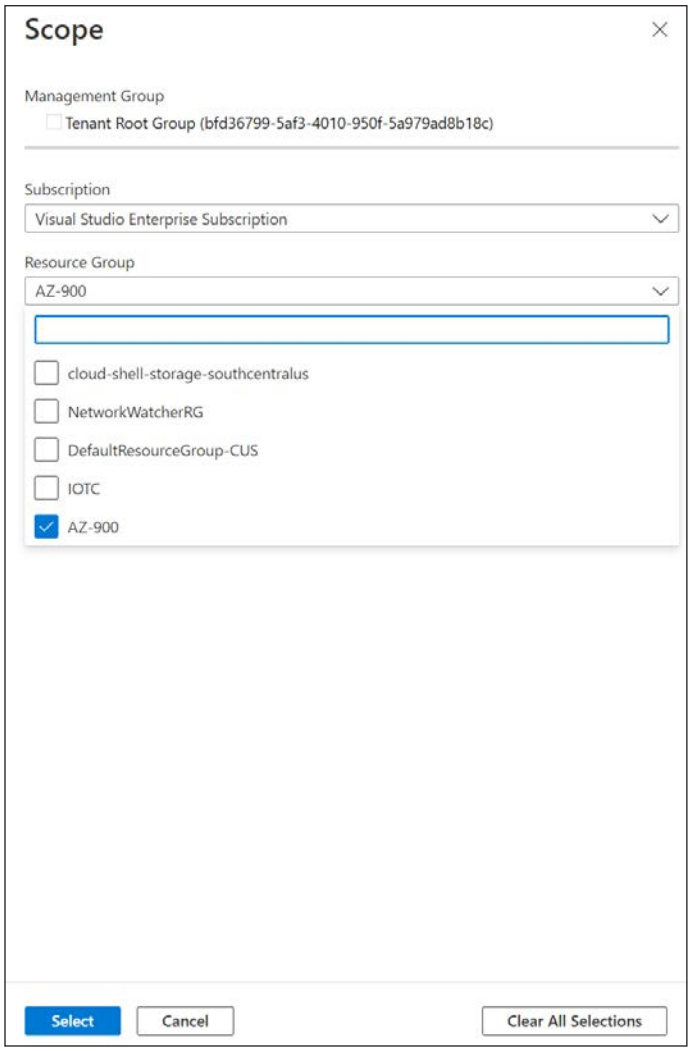


FIGURE 3-23 Changing the scope of the Policy blade in the portal

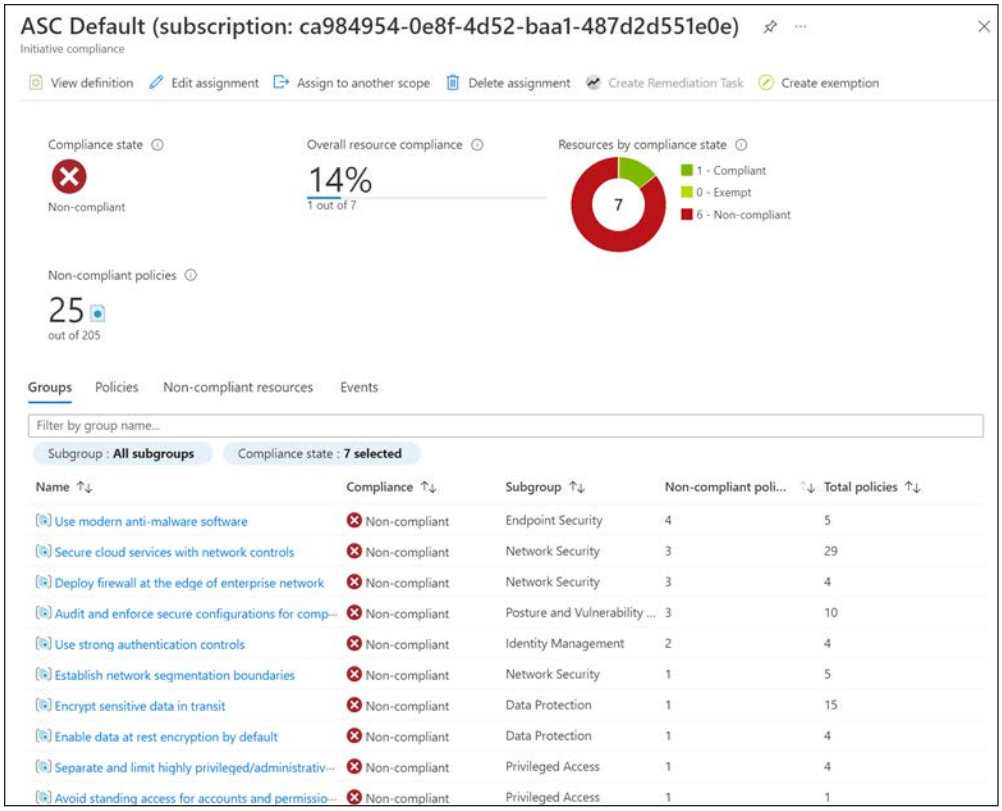


FIGURE 3-24 Details on compliance

Notice that the title of this item is `ASC Default` followed by a subscription ID. ASC Default is a collection of multiple policies that are defined by Azure Security Center. Azure Policy makes it easy to impose a full suite of policies by combining them into a group called an *initiative*. By defining an initiative, you can easily define complex rules that ensure governance of your company's policies.

You can assign a new policy either by selecting a policy from a list of included policies or by creating your own policy. To assign a policy from the list of included policies, click **Assignments > Assign Policy**, as shown in Figure 3-25.

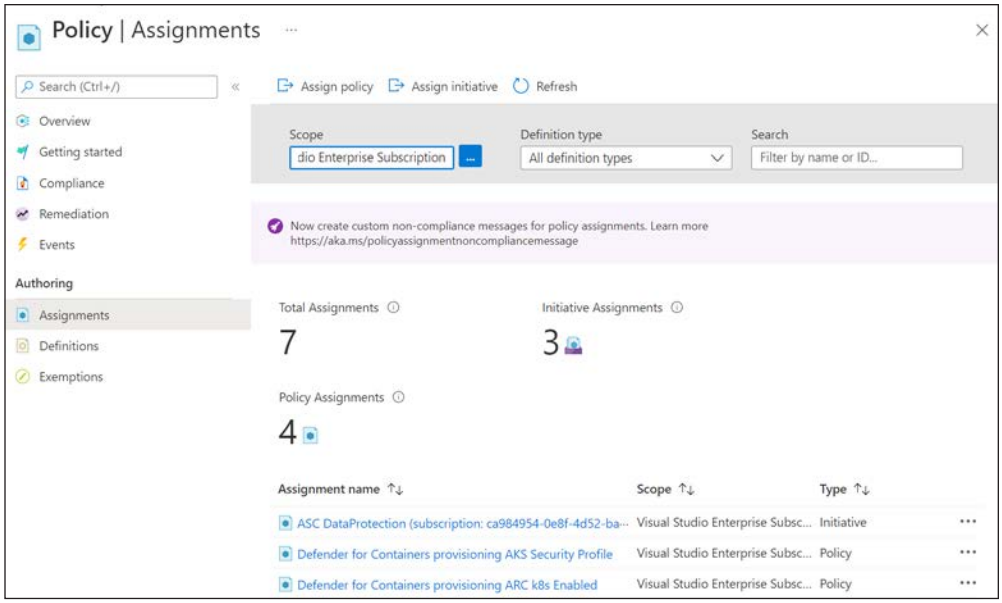


FIGURE 3-25 Assigning a policy

To select a policy, click the ellipses (**...**) next to Policy Definition, as shown in Figure 3-26.

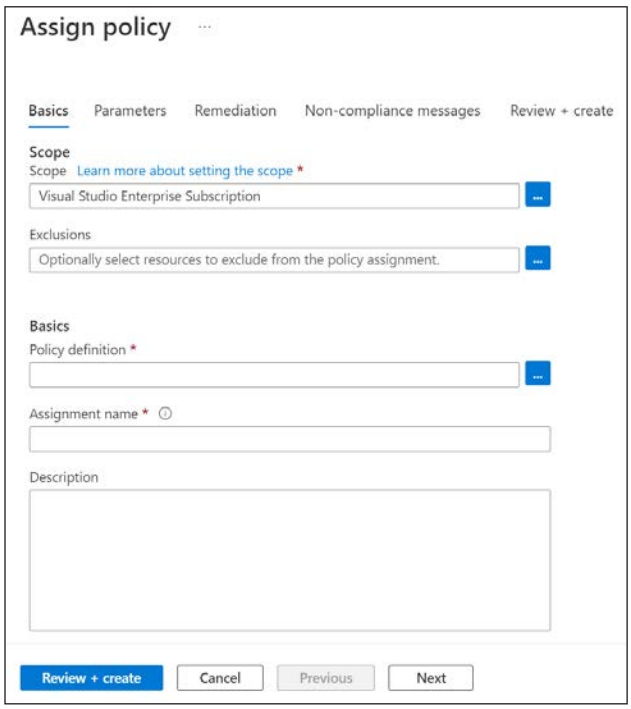


FIGURE 3-26 Selecting a policy definition

In this case, you apply a policy that will flag any App Service app that is not configured to use a virtual network service endpoint. You can do that by entering **app service** in the Search box and selecting the built-in policy that applies to that policy, as shown in Figure 3-27.

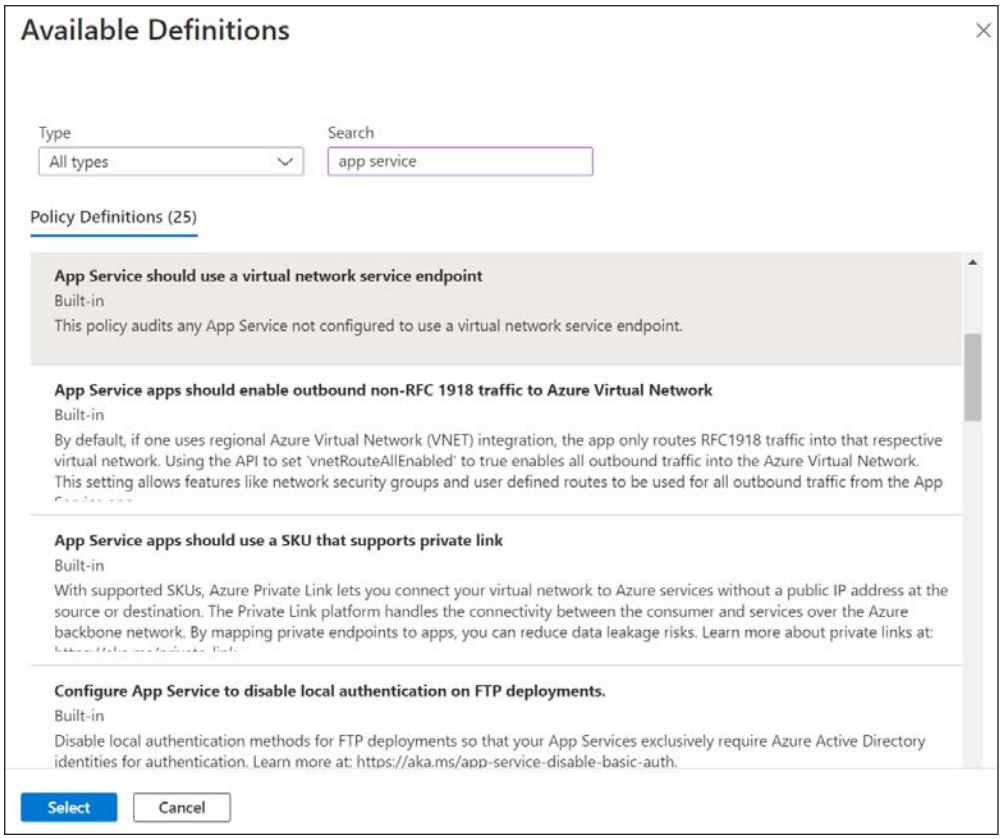


FIGURE 3-27 Adding a built-in policy definition

After clicking **Select** (shown in Figure 3-27), the details of this policy are shown. If you click the **Parameters** tab, you can see the effect of the policy. As you can see in Figure 3-28, the effect of this policy is **AuditIfNotExists**.

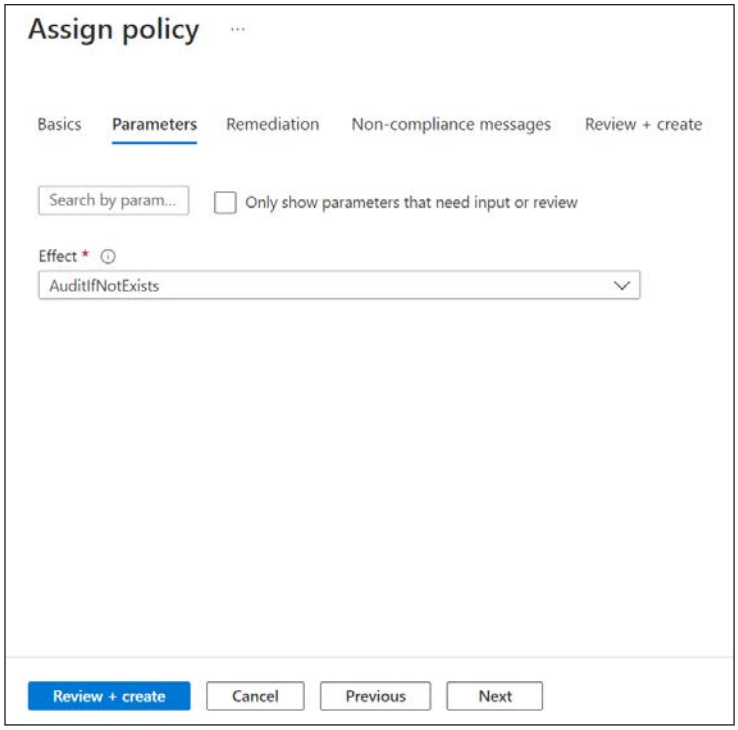


FIGURE 3-28 Completing the assignment

Six effects are supported in Azure Policy. However, not all effects are available for built-in policies. The effects are:

- **Append** Adds additional properties to a resource. It can be used to add a tag with a specific value to resources.
- **Audit** Logs a warning if the policy is not complied with.
- **AuditIfNotExists** Allows you to specify an additional resource type that must exist along with the resource being created or updated. If that resource type does not exist, a warning is logged.
- **Deny** Denies the create or update operation.
- **DeployIfNotExists** Allows you to specify an additional resource type you want deployed with the resource being created or updated. If that resource type is not included, it is automatically deployed.
- **Disabled** The policy is not in effect.

MORE INFO **MORE ON POLICY EFFECTS**

For more information on policy effects, including examples of each, see *https://bit.ly/az900-policyeffects*.

In addition to using the built-in policies, you can also define your own policies by creating a custom policy definition. Custom policy definitions are ARM templates that define the policy. For more information on creating a custom policy definition, see *https://bit.ly/az900-custompolicy*.

Resource locks

When you need to prevent changes to a resource, or prevent that resource from being deleted, resource locks (or locks) are a simple solution. Locks apply to everyone with access to the resource, regardless of which RBAC role they are assigned.

EXAM TIP

In order to create a lock, you must either be in the Owner or the User Access Administrator role in RBAC. Alternatively, an administrator can create a custom role that grants the right to create a lock.

Locks can be applied at the resource level, the resource group level, or at the subscription level. To apply a lock to a resource, open the resource in the Azure portal and click **Locks** in the **Settings** section of the menu on the left, as shown in Figure 3-29.

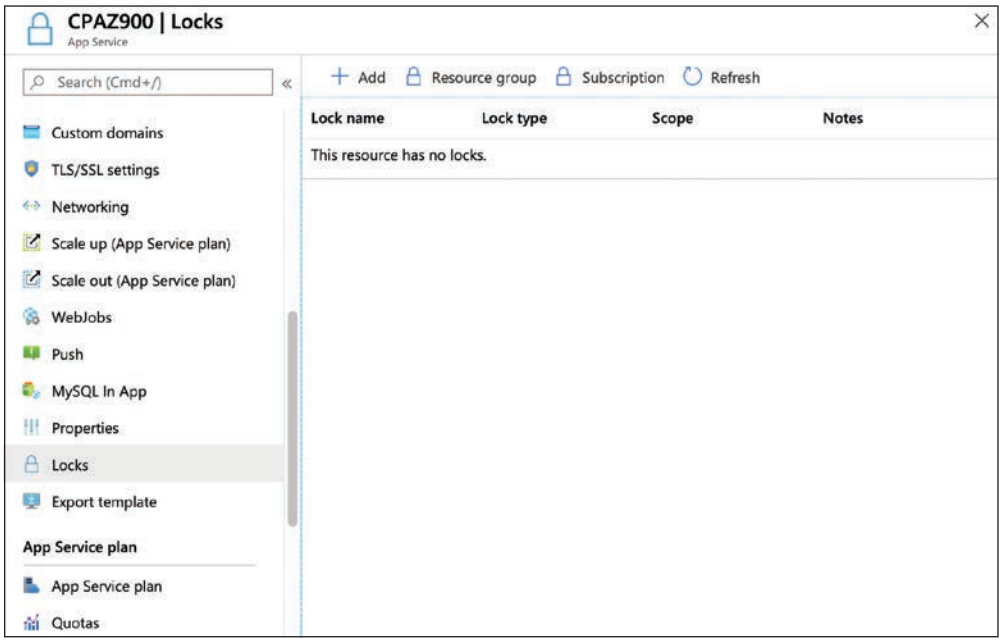


FIGURE 3-29 Locking a resource

To add a lock to the resource, click **Add**. (You can also review and add locks to the resource group by clicking **Resource Group**, or to the subscription by clicking **Subscription**.) In the **Lock Name** box, provide a name for the lock; set the **Lock Type**, and add an optional note, as shown in Figure 3-30.

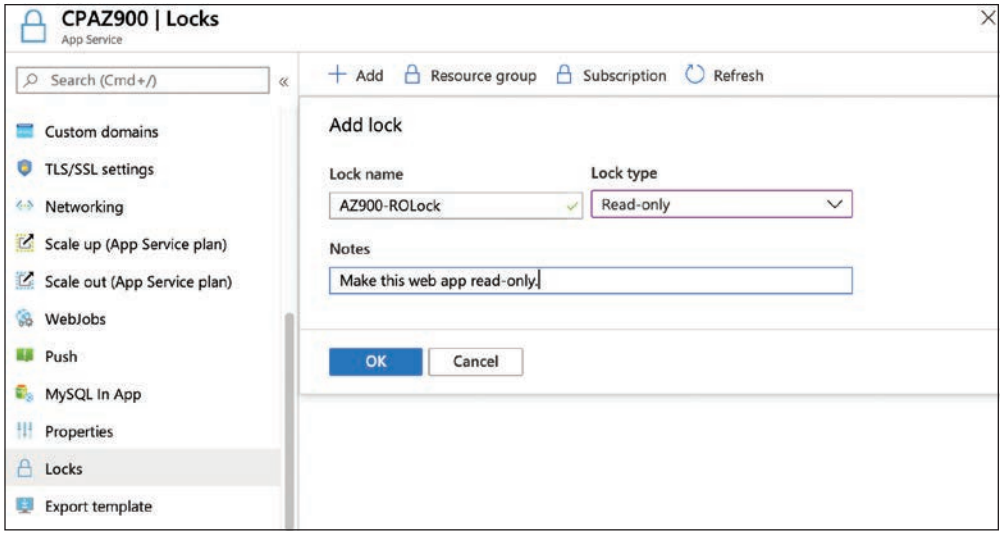


FIGURE 3-30 Adding a read-only lock

A read-only lock is the most restrictive lock. It prevents changing properties of the resource or deleting the resource. A delete lock prevents the resource from being deleted, but properties can still be changed. The result of a read-only lock is often unpredictable because of the way locks are handled by Azure.

Locks only apply to operations that are handled by Azure Resource Manager (ARM), and some operations specific to a resource are handled internally by the resource instead of being handled by ARM. For example, if you set a read-only lock on an instance of Azure Key Vault, it will prevent a user from changing access policies on the vault, but users can still add and delete keys, secrets, and certificates because those operations are handled internally by Key Vault.

MORE INFO **AZURE RESOURCE MANAGER (ARM)**

You'll learn about ARM in Skill 3.3, "Describe features and tools for managing and deploying Azure resources."

There are other situations where a read-only lock can prevent operations that occur unexpectedly. For example, if you place a read-only lock on a storage account, it will prevent all users from listing the access keys for the storage account because the operation to list keys makes the keys available for write access.

If a lock is applied to a resource group, all resources in that resource group inherit the lock. Similarly, if a lock is applied at the subscription level, all resources in the subscription inherit the lock. It is possible to nest locks, and in such situations, the most restrictive lock is the effective lock. For example, if you have a read-only lock on a resource group and a delete lock on a resource in that resource group, the resource will have a read-only lock applied to it because a read-only lock is more restrictive. The explicit delete lock will be irrelevant.

EXAM TIP

Locks are also inherited by newly created resources. If you apply a delete lock to a resource group and add a new resource to the resource group later, the new resource will automatically inherit the delete lock.

When an operation is attempted in the portal and denied because of a lock, an error will display, as shown in Figure 3-31.

EXAM TIP

Not all resource types will tell you that a lock prevented an operation that was attempted in the portal. There are times when you will see only a generic error message. If you try the same operation in the Azure CLI or using the Az module in PowerShell, you should see details on the lock.

You can edit or delete a lock by clicking **Locks** and clicking either **Edit** or **Delete** in the portal, as shown in Figure 3-32. In most cases, you will need to scroll to the right to see the **Edit** and **Delete** buttons.

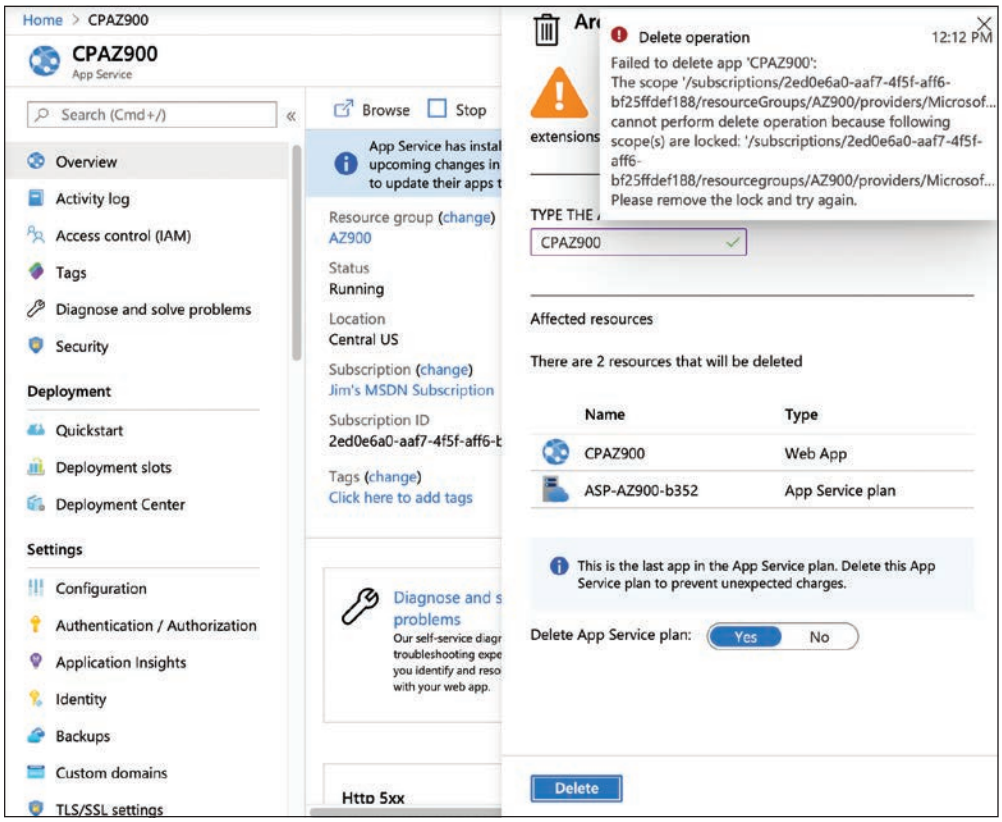


FIGURE 3-31 Denied by a lock

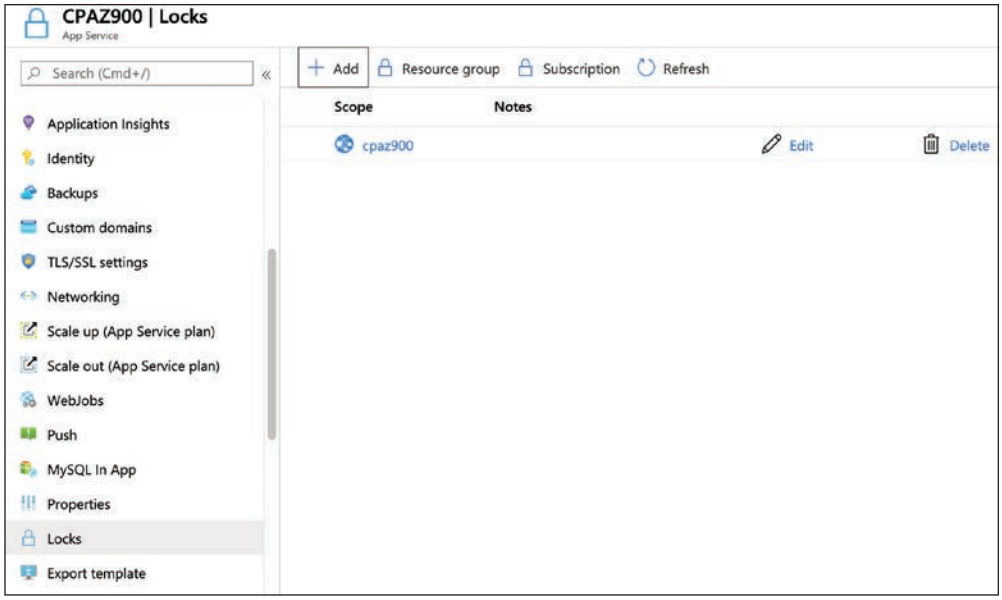


FIGURE 3-32 Editing or deleting a lock

Service Trust Portal

Microsoft has made all its information and tools on trust, security, and compliance in one convenient web portal called the Service Trust Portal. By browsing to *https://aka.ms/STP*, you can find documentation, white papers, compliance guides, FAQs, compliance tools, and much more.

You can also read about Microsoft's approach to trust and privacy, along with the results of audit reports from third parties who have checked Microsoft for compliance related to data protection and regulatory organizations.

The Service Trust Portal is also where you'll find links to Compliance Manager (a compliance tool for enterprises), the Security & Compliance Center with tools to aid in securing resources and maintaining compliancy, and numerous other resources related to governance, compliance, security, and privacy.

Skill 3.3: Describe features and tools for managing and deploying Azure resources

We've talked a lot about the Azure portal, and you've had the opportunity to see it used when interacting with several different Azure services. However, there are many other ways that you can manage and deploy your Azure services.

Many Azure users want to script interactions with their Azure services, especially when they have a need to interact with many VMs or other Azure resources. For those situations, Azure offers command line tools to help.

There are also situations where users want to apply some of the governance features such as policies, RBAC, and tags to resources that live outside of Azure, whether on another cloud provider or on-premises. Azure Arc makes doing that possible.

As we wrap up this section, you'll find out about Azure Resource Manager (ARM), the common link between all these management tools. You'll also learn how easy ARM makes it to redeploy an environment using ARM templates.

This section covers:

- Azure portal
- Azure PowerShell
- Azure command-line interface (CLI)
- Azure Cloud Shell
- Azure Arc
- Azure Resource Manager (ARM) and ARM templates

Azure portal

The Azure portal that is in use today is the third major iteration of the Azure portal.

The first time you open the Azure portal, you'll be prompted to take a tour of the portal. If you're completely unfamiliar with the portal, taking a tour will help you to get a feel for how it works. If you choose not to and change your mind later, you can click the question mark in the top toolbar to access the guided tour at any time.

The default view in the portal is Home, as shown in Figure 3-33. From here, you can see icons for various Azure services, and if you click one of those icons, it will show you any resources of that type that you've created. You can also see some of my resources listed. Clicking the menu button at the upper left displays a menu on the left side that includes these same icons and more.

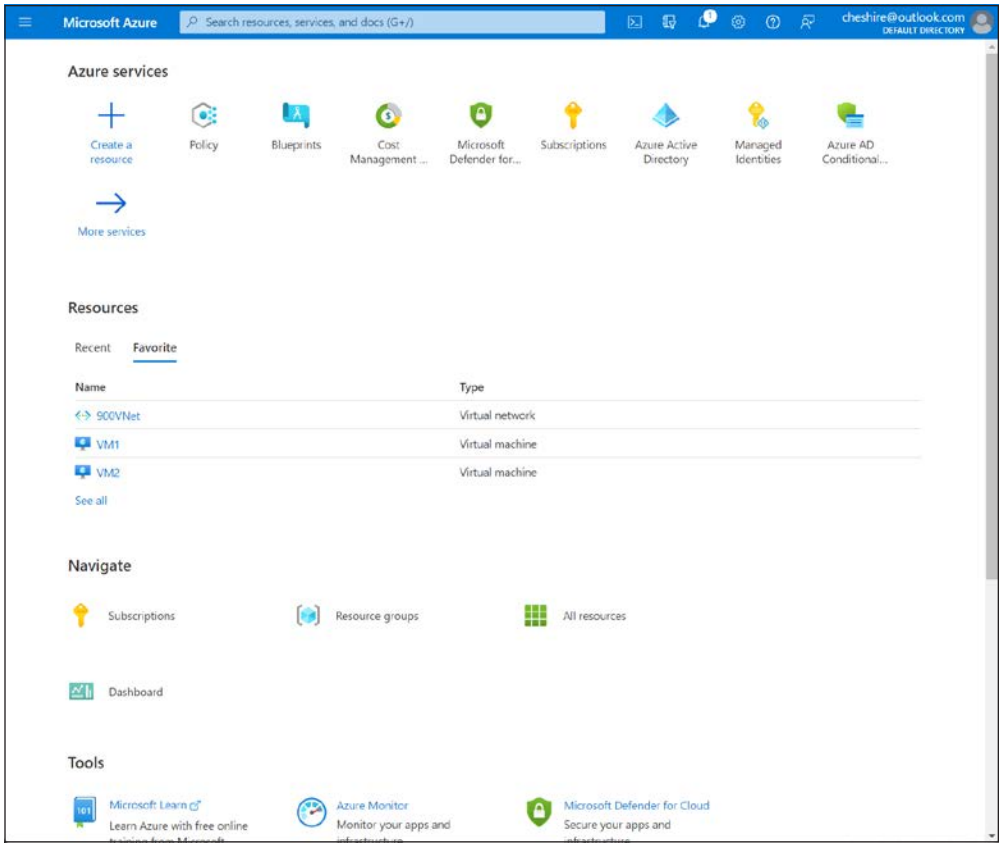


FIGURE 3-33 The Home screen in the Azure portal

You can navigate to resources of a certain type by clicking one of the links in the **Navigate** section. If you've recently viewed one of your resources, they'll appear in the list of resources so you can easily access them with one click.

Along the top bar, you'll find a search bar where you can search for Azure services, docs, or your Azure resources. To the right of the search box is the Cloud Shell button, which will launch Azure Cloud Shell. Cloud Shell is a web-based command shell where you can interact with Azure from the command line. You can create Azure resources and more. As you're reading through Azure documentation, you might see a **Try It** button, and those buttons use Cloud Shell to help you test out different services and features.

To the right of the **Cloud Shell** button is a **Filter** button that allows you to configure the portal to only show resources in a certain Azure subscription or Azure Active Directory. To the right of that is the **Notification** button. This is where you'll see notifications from Azure that are related to your services and subscription.

To the right of the notifications button is the **Settings** button. Clicking **Settings** shows the **Portal Settings** screen where you can alter portal settings, as shown in Figure 3-34.

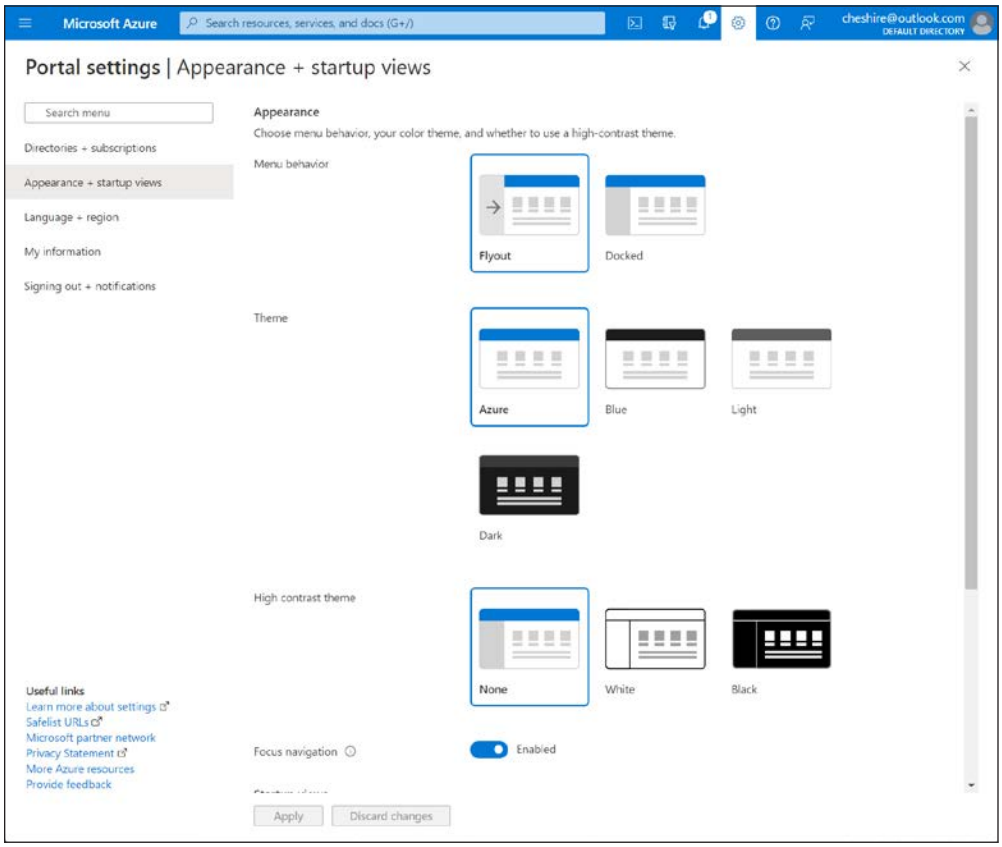


FIGURE 3-34 Portal settings

From **Settings**, you can change your AD directory, your default view, choose whether the menu is docked or flies out when you click the menu button, alter the theme of the portal, disable pop-up notifications, and more.

If you click your name in the upper-right corner (shown previously in Figure 3-34), you can log out or switch to other Azure accounts. You can also change the AD directory to access resources in another directory. This is helpful if your company has a corporate directory and you also have a personal directory.

As mentioned earlier, if you click the menu button in the upper-left portion of the portal, you'll see a menu that contains a default list of Azure resources. Clicking one of those will display all resources of that type. If you'd like to add an Azure service to the list, click **All Services**, as shown in Figure 3-35.

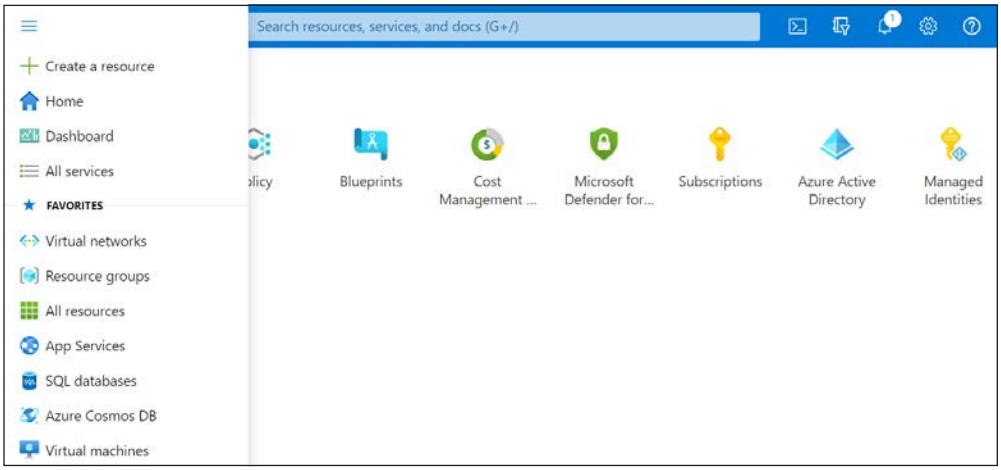


FIGURE 3-35 The Azure portal menu showing the All Services menu item

> ***NOTE*** **MOVING MENU ITEMS**
>
> **You can reorder items on the menu. Click and hold an item, and then drag it to a new location in the menu.**

From the list of all Azure Services, hover over the service you want to add to the list and click the star to the right of the service pop-up to mark it as a favorite, as shown in Figure 3-36.

In Figure 3-37, we clicked **Virtual Machines** on the menu to see all the VMs. From this list, you can click a resource to see that resource. You can also click a column header to sort by that column, assuming you have more than one resource of that type. Click **Manage View** to edit the columns that are displayed here, save the current view, and more. To create a new resource of this type, click **Create**.

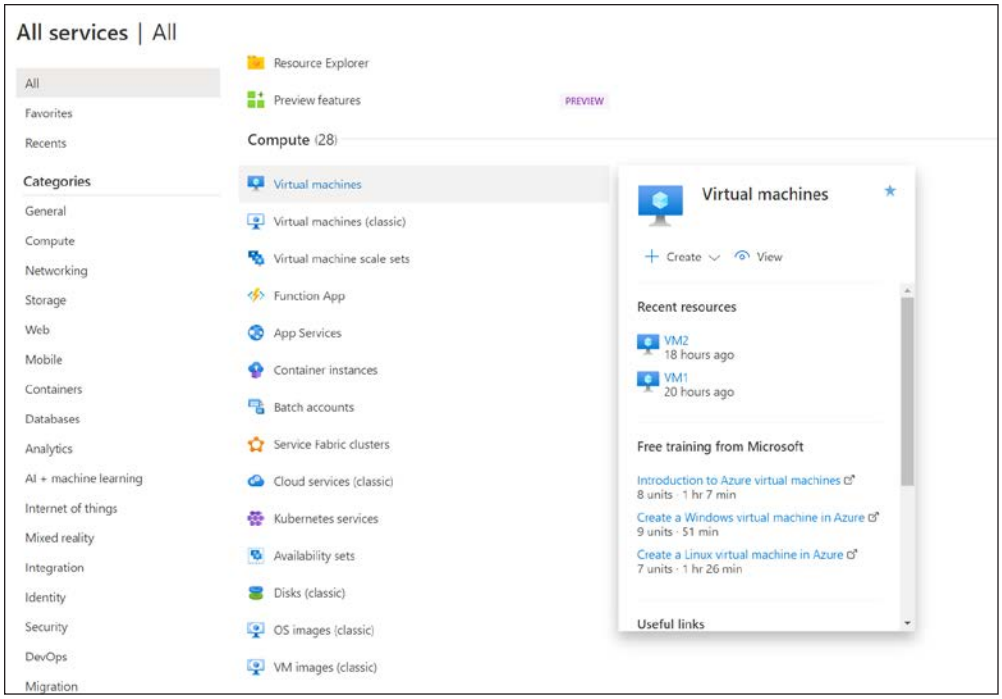


FIGURE 3-36 Making an Azure service a favorite so it will appear on the main menu

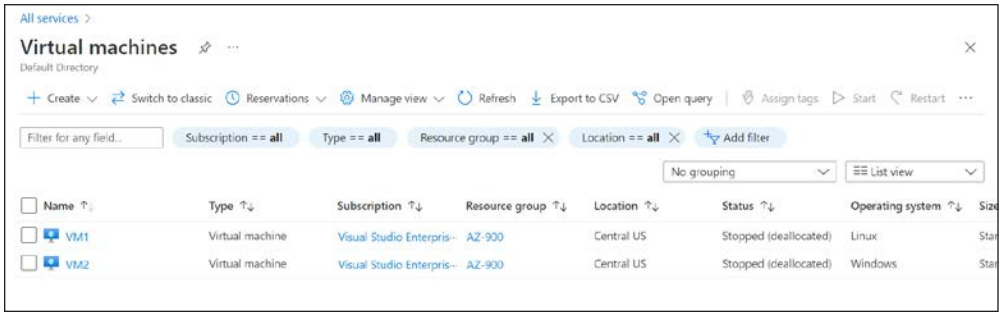


FIGURE 3-37 Viewing virtual machines in the portal

When you click a particular resource, it will open that resource in the portal. Along the left side will be a menu that's specific to the type of resource you opened. In the main window, you'll see different items based on the type of resource you're viewing. These window areas in the portal are often referred to as blades.

In Figure 3-38, you see a VM in the portal. The **Overview** blade is a blade that's common to most Azure resources, but the information that appears there will differ based upon the resource. When you're looking at a VM, you can see the resource group it's in, the status, the location, and more. In the upper left (next to the VM name) is a pin button. If you click that pin, it will add this VM to the portal dashboard.

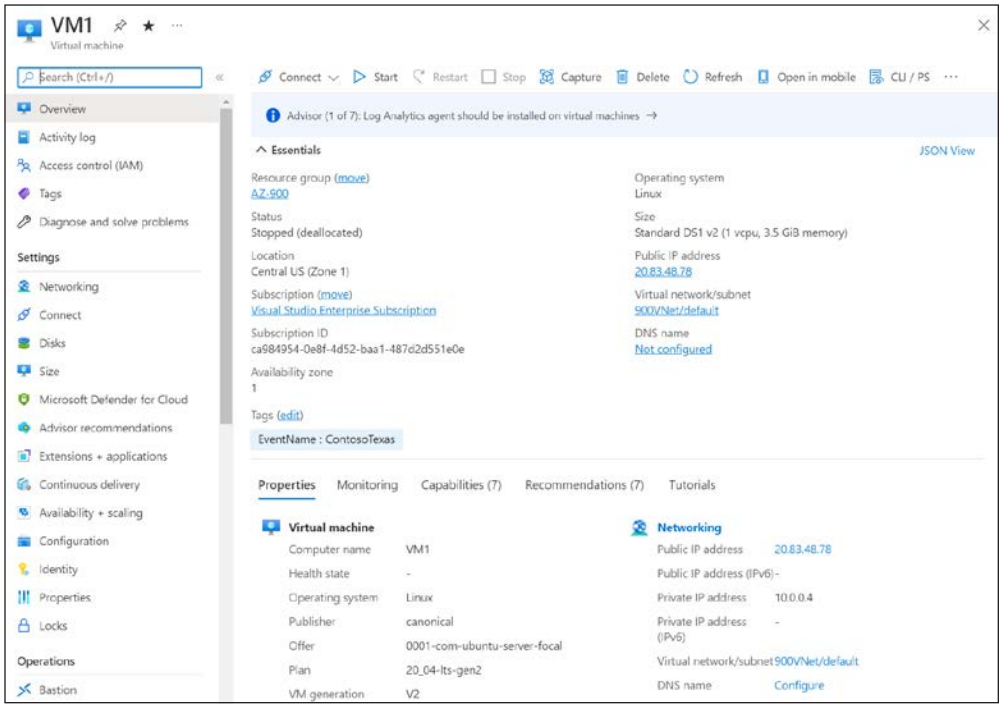


FIGURE 3-38 Viewing a VM in the portal

Along the top of the blade for the VM are several buttons for interacting with the resource. For a VM, you have a **Connect** button that will allow you to connect to the VM, a **Start** button to start the VM if it's stopped, a **Restart** button to restart the VM, and so on. Each resource type will have different buttons available to you so you can easily interact with the resource from the **Overview** blade.

If you click an item in the menu at the left, the content from the **Overview** blade is replaced with the selected new item. In Figure 3-39, we have clicked **Diagnose And Solve Problems**, which replaces the **Overview** blade with new content from the **Diagnose And Solve Problems** blade.

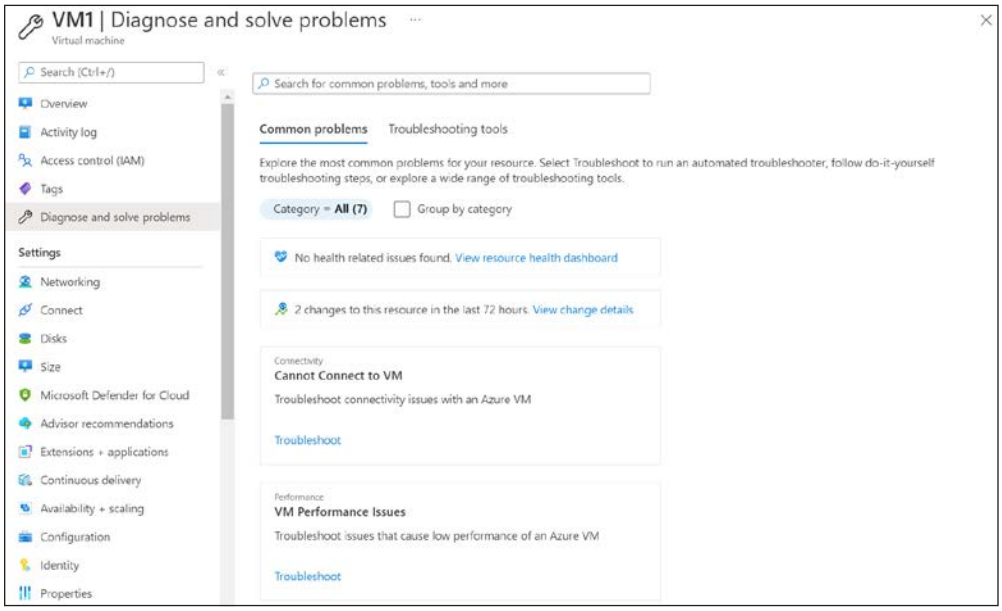


FIGURE 3-39 A new blade

As you use the portal, you'll find that there is inconsistency between different services. Each Microsoft team has its own portal development team and tends to design portal interfaces that make sense for its own team. For that reason, you might see buttons on the top in some blades and buttons on the bottom in other blades.

You can customize your portal experience using the dashboard. If you click Dashboard from the portal menu, you'll see your default dashboard. As you're managing your resources, click the pin icon (as shown in the upper-left portion of Figure 3-38) to pin tiles to your dashboard. You can then move these tiles around and customize them in other ways to create a view that's unique to your needs.

To customize your dashboard, click **Dashboard** in the menu to show the dashboard and then click **Edit** at the top of the screen, as shown in Figure 3-40.

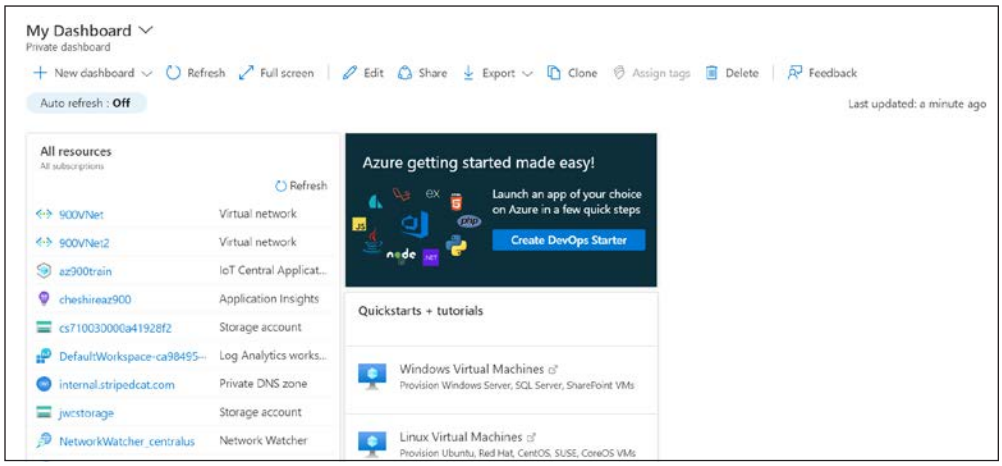


FIGURE 3-40 The Edit button allows for customization of your dashboard

From the screen shown in Figure 3-41, you can change the name of your dashboard by clicking inside the current name and changing it to a new name. You can add tiles to the dashboard by choosing from one of the hundreds of tiles available in the **Tile Gallery** on the right side of the portal, and you can search and filter the list if necessary. If you hover over an existing tile, you'll see a **Delete** button and a menu button that is represented by three dots. Click the **Delete** button to remove the tile from the dashboard. Click the menu button to access a context menu where you can resize the tile.

When you're satisfied with your dashboard, click **Done Customizing** to close the customization screen.

You can create new dashboards for specific purposes by clicking the New Dashboard button shown previously in Figure 3-40. This takes you into a customization screen for your new dashboard, just like the one shown in Figure 3-41.

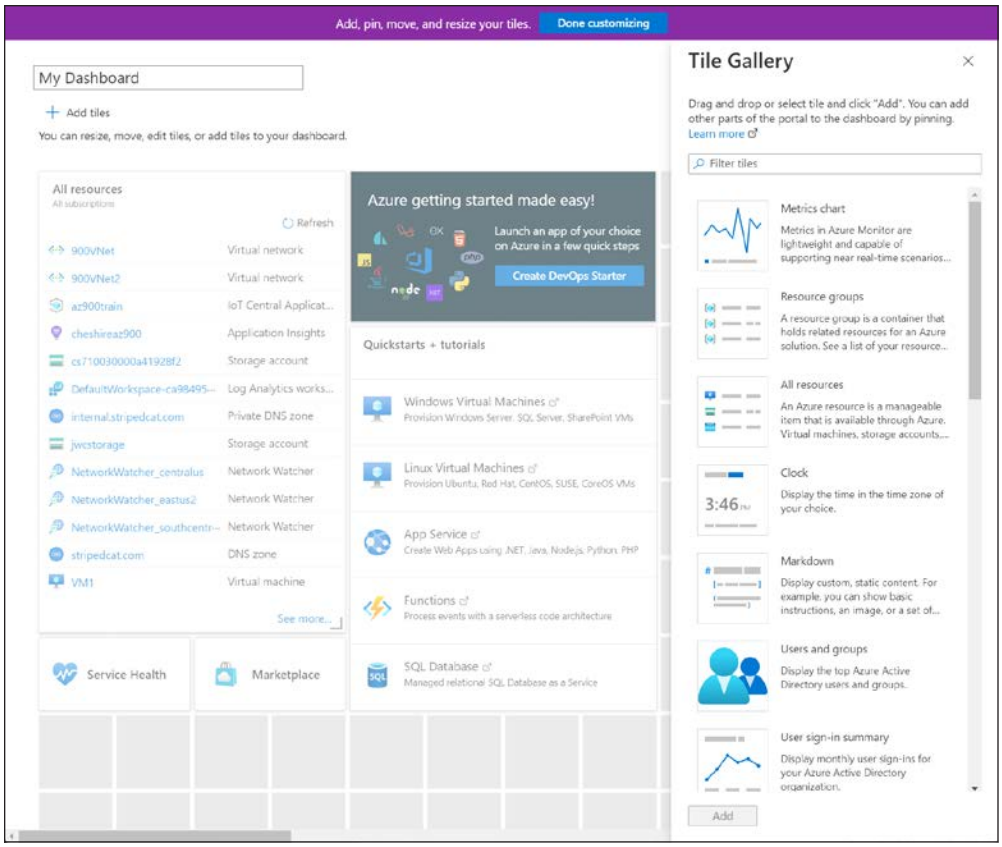


FIGURE 3-41 Customizing a dashboard

In Figure 3-42, we've created a dashboard specific to VMs. You can easily switch between this dashboard and the default dashboard by clicking the down arrow next to the dashboard name.

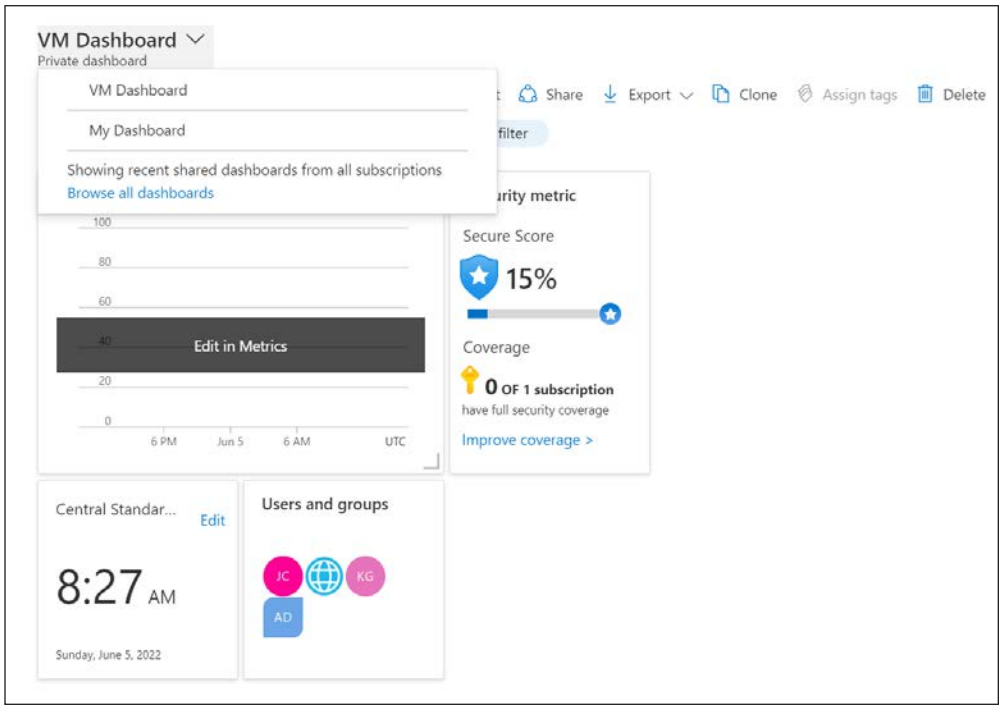


FIGURE 3-42 Switching between dashboards

Azure PowerShell

If you're a PowerShell user, you can take advantage of that knowledge to manage your Azure resources using the Azure PowerShell `Az` module. This module offers cross-platform support, so whether you're using Windows, Linux, or macOS, you can use the PowerShell `Az` module.

> ***MORE INFO*** **INSTALL POWERSHELL ON LINUX OR macOS**
>
> **If you're running Linux, you can find details on installing PowerShell at *https://bit.ly/az900-powershellonlinux*. macOS users can find steps at *https://bit.ly/az900-powershellonmac*.**

EXAM TIP

The PowerShell `Az` module uses the .NET Standard library for functionality, which means it will run with PowerShell version 5.x, 6.x, or 7.x. PowerShell 6.x and 7.x are cross-platform and can run on Windows, Linux, or macOS.

Before you can use the PowerShell `Az` module, you'll need to install it. To do that, you first need to run PowerShell elevated. In Windows, that means running it as an Administrator. In Linux and macOS, you'll need to run it with superuser privileges using Sudo.

To install the module, run the following command.

```
Install-Module -Name Az -AllowClobber
```

When you install a new PowerShell module, PowerShell checks all existing modules to see if they have any command names that are the same as a command name in the module you're installing. If they do, the installation of the new module fails. By specifying `-AllowClobber`, you are telling PowerShell that it's okay for the *Az* module to take precedence for any commands that also exist in another module.

If you are unable to run PowerShell elevated, you can install the module for your user ID only by using the following command:

```
Install-Module -Name Az -AllowClobber -Scope CurrentUser
```

Once you've installed the module, you need to sign in with your Azure account. To do that, run the following command:

```
Connect-AzAccount
```

This command will display a token in the PowerShell window. You'll need to browse to *https://microsoft.com/devicelogin* and enter the code to authenticate your PowerShell session. If you close PowerShell, you'll have to run the command again in your next session.

MORE INFO **PERSISTING CREDENTIALS**

It is possible to configure PowerShell to persist your credentials. For more information on doing that, see *https://docs.microsoft.com/powershell/azure/context-persistence*.

If you have more than one Azure subscription, you'll want to set the active subscription so that commands you enter will affect the desired subscription. You can do that using the following command:

```
Set-AzContext -Subscription "subscription_id"
```

Replace `subscription_id` with the subscription ID of your Azure subscription you want to use with the `Az` module.

All *Az* module commands will have a common syntax that starts with a verb and an object. Verbs are things like `New`, `Get`, `Move`, or `Remove`. The object is the thing that you want the verb to affect. For example, the following command will create a resource group called `MyRG` in the South Central US region:

```
New-AzResourceGroup -Name MyRG -Location "South Central US"
```

If this succeeds, you'll see a message letting you know that. If it fails, you'll see an error. To remove the resource group, run the following command:

```
Remove-AzResourceGroup -Name MyRG
```

When this command is entered, you'll be asked to confirm whether you want to delete the resource group. Type a **y** and the resource group will be removed, as shown in Figure 3-43.

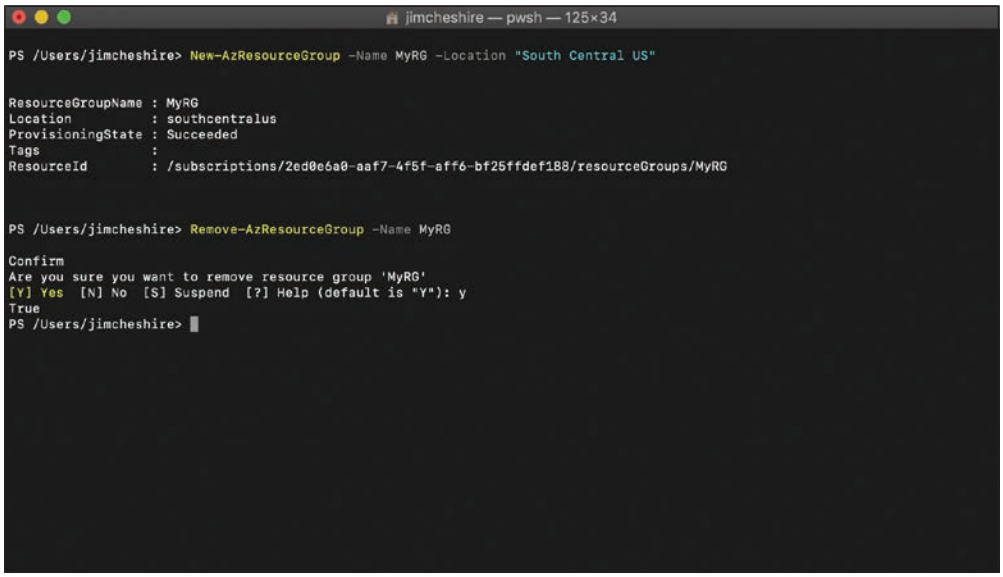


FIGURE 3-43 Creating and deleting a resource group with the *Az* module

In many situations, you will be including PowerShell commands in a script so that you can perform several operations at once. In that case, you won't be able to confirm a command by typing **y**, so you can use the `-Force` parameter to bypass the prompt. For example, you can delete the resource group using the following command and you won't be prompted.

```
Remove-AzResourceGroup -Name MyRG -Force
```

You can find all the commands available with the PowerShell `Az` module by browsing to *https://bit.ly/az900-powershellaz* and clicking Reference in the left menu.

Azure command-line interface (CLI)

As I pointed out earlier, one of the main benefits of PowerShell is the ability to script interactions with Azure resources. However, if you want to script with PowerShell, you'll need to know PowerShell development. If you don't have that knowledge, the Azure command-line interface (Azure CLI) is a great choice. Azure CLI can be scripted using shell scripts in various languages like Python, Ruby, and so on.

Like the PowerShell *Az* module, the Azure CLI is cross-platform and works on Windows, Linux, and macOS if you use the 2.0 version or later. Installation steps are different depending on your platform. You can find steps for all operating systems at *https://bit.ly/az900-installcli*.

Once you install the Azure CLI, you'll need to log in to your Azure account. To do that, run the following command:

```
az login
```

When you run this command, the CLI will open a browser automatically for you to log in. Once you log in, if you have multiple Azure subscriptions, you can set the default subscription by entering the following command:

```
az account set --subscription "subscription_id"
```

Replace `subscription_id` with the subscription ID you want to use.

To find a list of commands you can run with the CLI, type **az**, and press Enter. You'll see a list of all the commands you can run. You can find detailed help on any command by entering the command and adding a `--help` parameter. Figure 3-44 shows the help for `az resource`.

You can take this a step further if you aren't sure what the commands do. You can, for example, run the following command to get help on the syntax for `az resource create`:

```
az resource create --help
```

This provides you with help and example commands to understand the syntax.

EXAM TIP

Like PowerShell, most commands in the Azure CLI have a `--force` parameter that you can include so that no prompts are displayed. When scripting PowerShell or the CLI, you need to include this parameter, or your script won't work. Watch out for examples in the AZ-900 exam that test for this kind of knowledge.

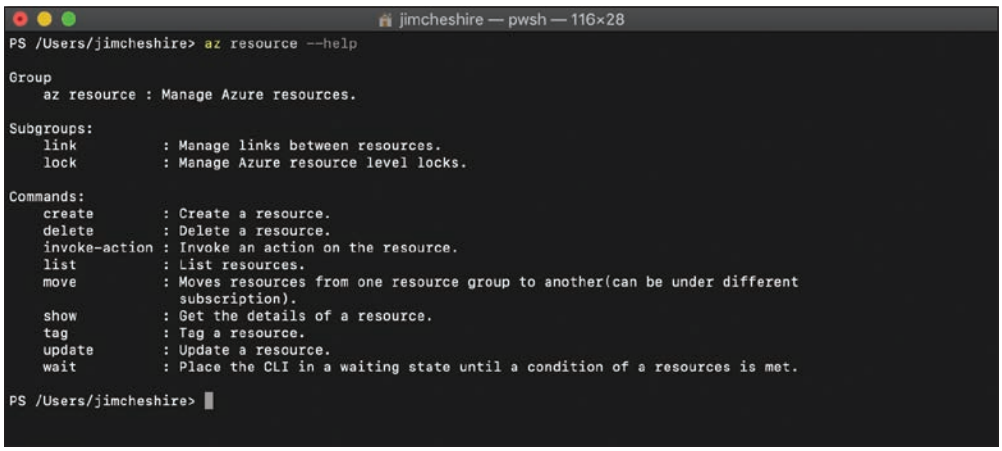


FIGURE 3-44 Azure CLI help

An even easier way to learn the CLI is to switch into interactive mode. This provides you with auto-complete, the scoping of commands, and more. To switch into interactive mode, enter **az interactive** at the command prompt. The CLI will install an extension to add this functionality. Figure 3-45 shows the Azure CLI with interactive mode active. At the command prompt, **we** has been typed, and it's displaying the rest of the command in dimmed text. You can press the right arrow key to enter the dimmed text in one keystroke.

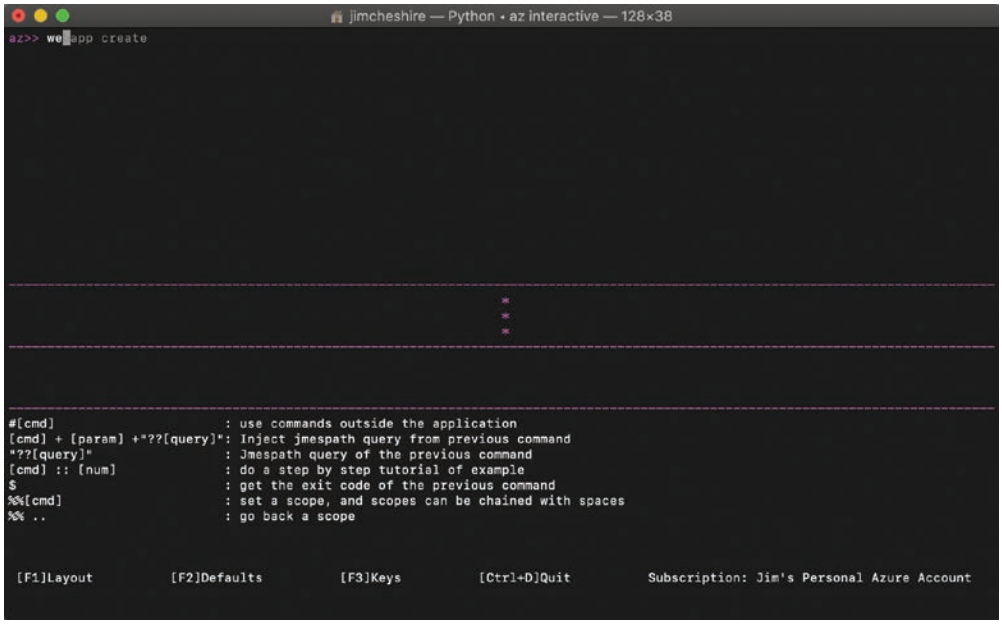


FIGURE 3-45 CLI interactive mode

You can install additional extensions for added functionality. Because the CLI uses an extension architecture, Azure teams can provide support for new functionality without having to wait for a new CLI release. You can find a list of all available extensions that Microsoft provides by running the following command:

```
az extension list-available --output table
```

This will not only show you available extensions, but it will show you if you already have the extension installed and whether there's an update you should install. To install an extension, run the following command:

```
az extension add --name extension_name
```

Replace `extension_name` with the name of the extension you want to install.

Azure Cloud Shell

As we've already seen, command-line access to your Azure resources with PowerShell and the Azure CLI is powerful and flexible, and we also learned that you can install extensions to add more power to your command line. However, if you use multiple machines, you'll need to install those extensions on each machine, and that might be a hassle. You're also limited to running these command-line tools on your computer. You can't run them on your phone or your tablet when you're away from your computer.

The natural solution to these problems is to have your command-line tools available to you in the cloud, and that's exactly what Microsoft did with Azure Cloud Shell.

To access Cloud Shell, click the **Cloud Shell** button in the Azure portal, as shown in Figure 3-46.

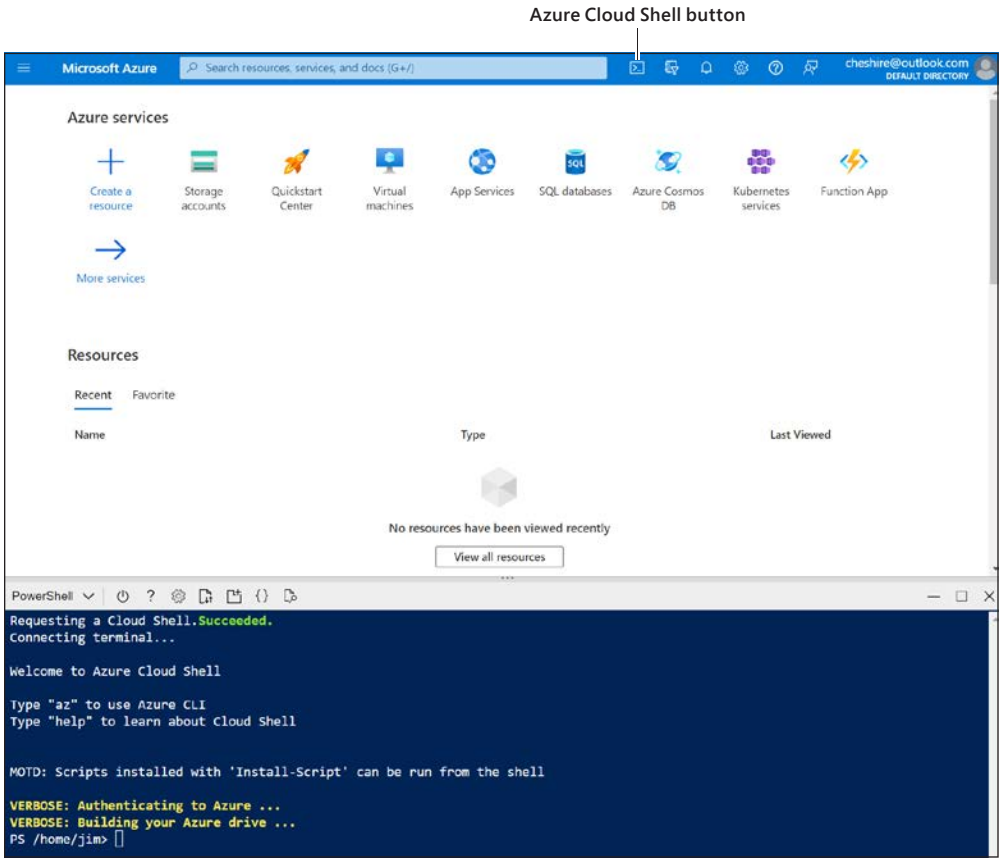


FIGURE 3-46 Cloud Shell in the Azure portal

The first time you use Cloud Shell, it will ask you to create a storage account. Cloud Shell persists anything you install and your settings throughout all your devices, so you need a storage account to persist those. In Figure 3-47, a storage account is being created for Cloud Shell.

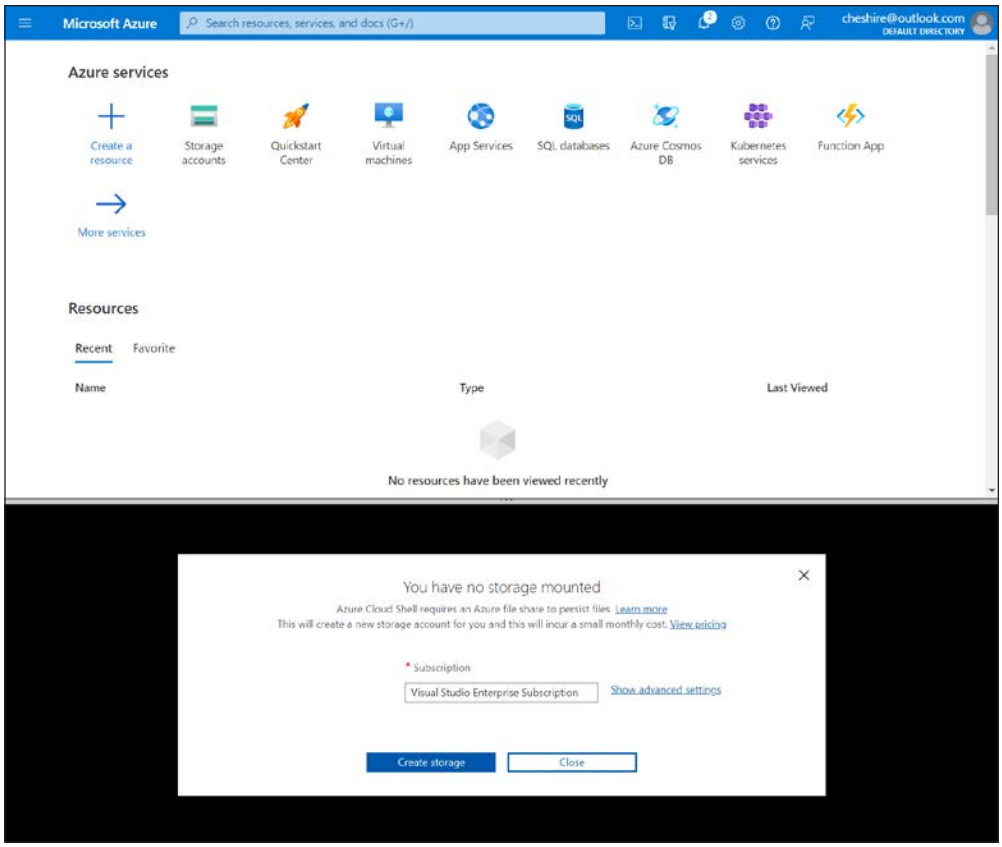


FIGURE 3-47 Creating a storage account for Cloud Shell

Once the storage account is created, Cloud Shell will launch a session, as shown previously in Figure 3-46.

From Cloud Shell, you can run any of the commands you can run in the Azure CLI and the PowerShell Az module, depending on which you choose. At the top of the Cloud Shell window is a toolbar. To change between PowerShell and Bash (CLI), select the desired environment from the dropdown menu. Next to that dropdown menu is the "power" button that restarts the Cloud Shell session, followed by a **Help** button and a **Settings** button for changing the text size and the font. To the right of the **Settings** button is a button that allows you to upload or download files into your file share for Cloud Shell, followed by a button that opens a new session of Cloud Shell in a new browser tab.

EXAM TIP

Any files that you upload will be available to you in Cloud Shell on any of your devices. That's because your files are stored in your Azure storage account. When you open Cloud Shell, Azure will pick a Cloud Shell instance for you to connect to and it will copy files from your storage account to that Cloud Shell instance.

The **Open Editor** button on the toolbar will open an instance of the Monaco Editor, a code editor that makes it easy to edit scripts and other files. In Figure 3-48, a JSON file is open in the editor, and a file browser is shown on the left.

Open Editor button

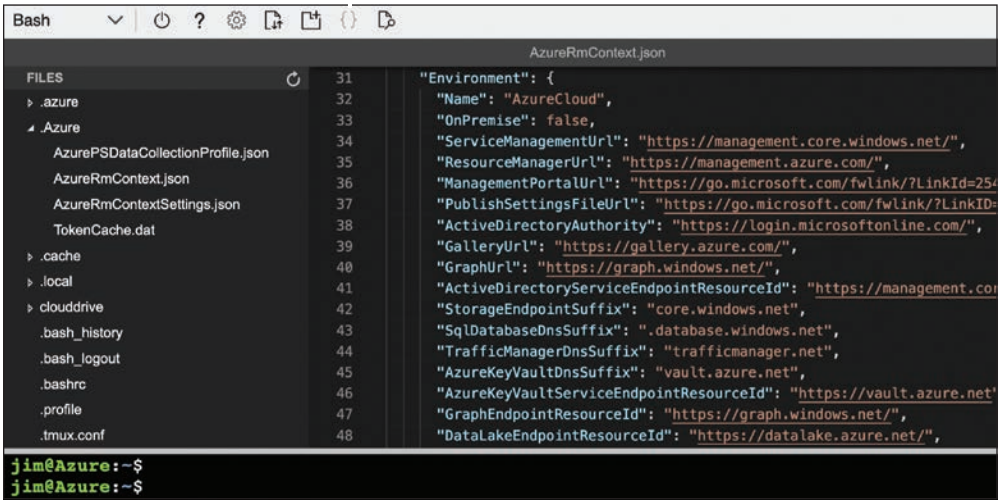


FIGURE 3-48 The file editor in Cloud Shell

NOTE **CLOSING THE EDITOR**

To exit the editor, right-click in the window and click Quit.

The last button on the toolbar is the **Web Preview** button. This button allows you to run a web application using the files in the current folder inside of your web browser. This is a powerful tool for developers who might be developing web applications using Cloud Shell.

In Figure 3-49, I'm running a .NET Core web app in Cloud Shell using the `dotnet run` command. (The dotnet application is used to start applications written for .NET Core.) After I do that, I can see that the app is running on the Cloud Shell instance on port 5000.

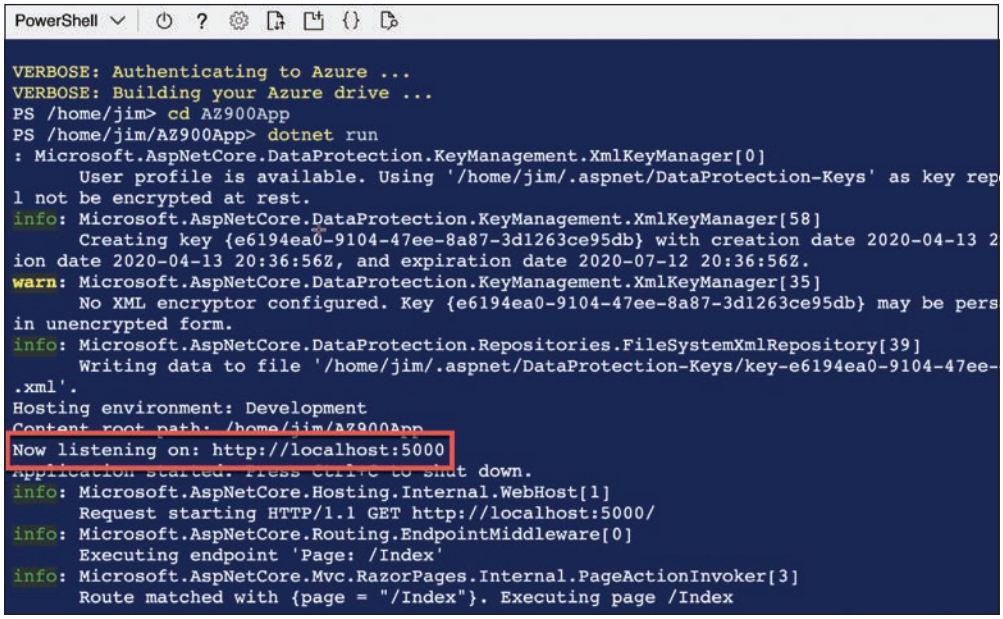


FIGURE 3-49 Running a .NET Core web app from Cloud Shell

If you want to open that in a browser and view it, click **Web Preview** in Cloud Shell and choose **Configure**. Enter the port number (**5000** in this case), as shown in Figure 3-50.

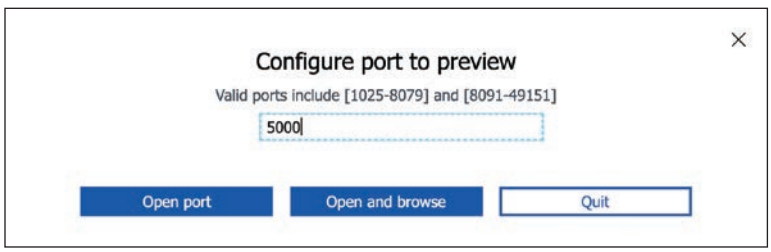


FIGURE 3-50 Configuring a port for Web Preview in Cloud Shell

I can then click **Open And Browse** and see my web app in my browser, as shown in Figure 3-51. It's clear from my site that I have some styling issues with a CSS file, and previewing it allows me to troubleshoot that problem and any other problems with my app.

> ***NOTE*** **WEB PREVIEW**
>
> **Why would someone want to preview an app in Cloud Shell instead of just debugging it locally? Many developers write applications that interact with other Azure services, and they might want to debug these applications while they're running in Azure. For command-line developers, this technique in Cloud Shell is a powerful way to enable that.**

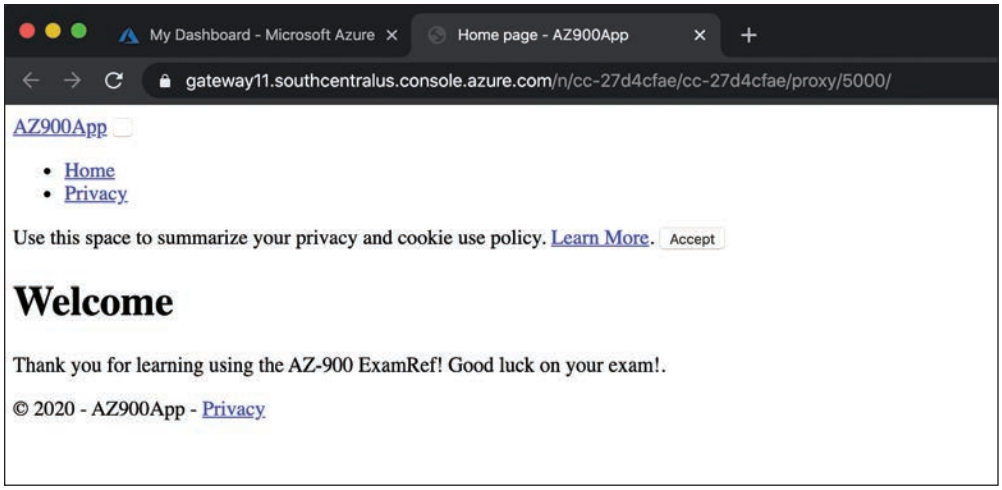


FIGURE 3-51 Browsing a web app using Web Preview

So far, we've only looked at using Cloud Shell from the Azure portal, but that's not the only way to access Cloud Shell. The Azure documentation provides many examples of PowerShell and Azure CLI scripts, and in many cases, there's a **Try It** button that you can click to try the script from within your browser. When you click the **Try It** button, an instance of Cloud Shell will open so that you can easily enter the script and run the commands, as shown in Figure 3-52.

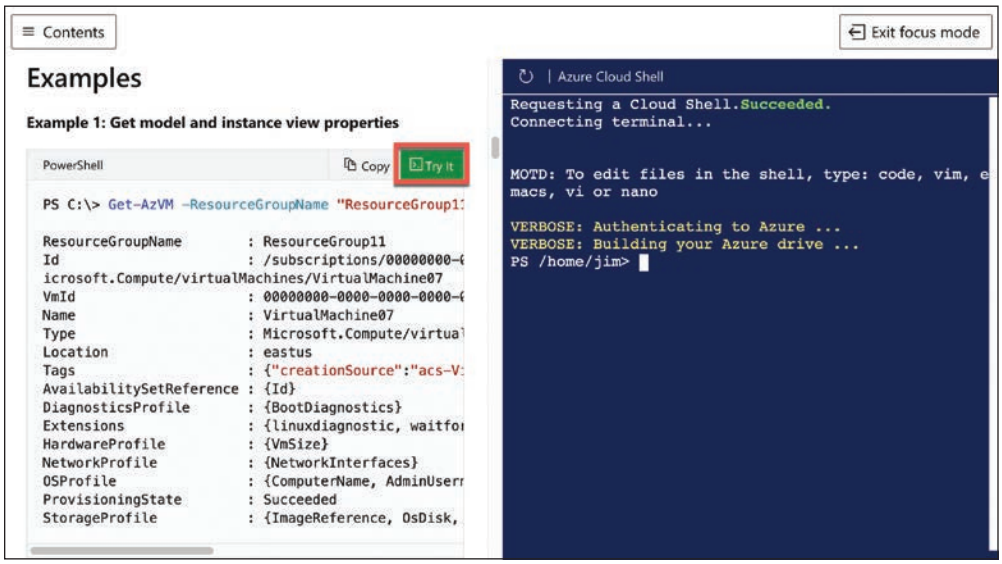


FIGURE 3-52 Cloud Shell's integration with Microsoft documentation

Azure Arc

In Chapter 1, we discussed cloud models, and in that discussion, I pointed out that many companies are adopting the hybrid cloud model. It's common for companies to have some resources in Azure and some on-premises. It's also not uncommon for people to have some resources in other cloud providers.

When resources are spread out in multiple environments, managing those resources can be difficult, especially when you want to apply some governance to those resources. To help with this problem, Microsoft released Azure Arc.

Azure Arc extends management and governance features of Azure to resources outside of Azure. It does this using one of two different methods: Azure Arc-enabled servers or Azure Arc–enabled Kubernetes.

Azure Arc-enabled servers allow you to extend Azure management features to your physical servers and virtual machines running outside of Azure. To use Azure Arc-enabled servers, you will need to install the Azure Connected Machine agent to the machine. This agent collects information about the server or VM and makes it available to Azure. It also installs some extensions from Azure onto your VMs that are used for VM management.

EXAM TIP

Azure Arc-enabled servers can be Windows servers or Linux servers.

Once the Azure Connected Machine agent is installed, your server or VM will have a managed identity in Azure, allowing you to easily manage the server or VM using Azure management tools. This includes RBAC, Azure Policy, tags, protection with Azure Defender for Cloud, and more.

NOTE **SQL SERVER ON ARC-ENABLED SERVERS**

You can register your SQL Server instances running on Arc-enabled servers. Doing so makes it possible to manage those SQL Server instances using Azure.

Azure Arc-enabled Kubernetes extends Azure management and governance features to your Kubernetes clusters running on-premises or in other cloud providers. Like Azure Arc-enabled servers, Azure Arc-enabled Kubernetes works via the installation of an agent that you install. Once you've created your Kubernetes cluster outside of Azure, you can use the Azure CLI or PowerShell to begin the registration of your cluster with Azure Arc.

When you register your cluster, Azure connects your Kubernetes cluster to Azure via a secured connection, enabling you to manage the cluster using Azure features.

EXAM TIP

Azure Arc-enabled Kubernetes uses a feature of the Git source control system called GitOps to deploy and manage Arc-enabled Kubernetes clusters.

Arc-enabled Kubernetes enables a lot of functionality besides simply managing Kubernetes clusters. Arc-enabled data services make it possible to run Azure SQL Managed Instance or PostgreSQL Hyperscale services on your Kubernetes clusters outside of Azure. Azure application services allow you to run apps on several Azure services on Arc-enabled Kubernetes, including Azure App Service, Azure Functions, Azure Logic Apps, Azure Event Grid, and Azure API Management.

The functionality of Azure Arc has grown significantly since it was announced, and you can expect Microsoft to continue to leverage this technology to differentiate its offerings from competitors.

Azure Resource Manager (ARM) and ARM templates

Almost all systems that are moved to the cloud consist of more than one Azure service. For example, you might have an Azure virtual machine for one part of your app; your data might be in an Azure SQL Database; you might have some sensitive data stored in Azure Key Vault; and you might have a web-based portion of your app hosted in Azure App Service.

If you must manage all these different Azure services separately, it can be quite a headache, and if you have multiple applications in the cloud, it can be even worse. Not only would it be confusing to keep track of which services are related to which applications, but when you add in the complexity of deploying updates to your application, things can really become disorganized.

To make it easier to deploy and manage Azure services, Microsoft developed Azure Resource Manager (ARM). ARM is a service that runs in Azure, and it's responsible for all interactions with Azure services. When you create a new Azure service, ARM authenticates you to ensure you have the right access to create that resource, and then it talks to a *resource provider* for the service you're creating. For example, if you're creating a new web app in Azure App Service, ARM will pass your request on to the `Microsoft.Web` resource provider because it knows all about web apps and how to create them.

EXAM TIP

All management tools for Azure use ARM on the back end. That includes the Azure portal, Azure PowerShell, and the Azure CLI.

Management tools interact with ARM using the ARM application programming interface (API). The ARM API is the same whether you're using the portal or command-line tools, which means you get a consistent result. It also means that you can create an Azure resource with the portal and then make changes to it using command-line tools, allowing you the flexibility that cloud consumers need.

MORE INFO **VISUAL STUDIO AND ARM**

Visual Studio, Microsoft's development environment for developing applications, also can create Azure resources and deploy code to them. It does this using the same ARM API we've mentioned previously. In fact, you can think of the ARM API as your interface into the world of Azure. You really can't create or manage any Azure services without going through the ARM API.

The flow of a typical ARM request to create or manage a resource is straightforward. Tools such as the Azure portal, command-line tools, or Visual Studio make a request to the ARM API. The API passes that request to ARM, where the user is authenticated and authorized to perform the action. ARM then passes the request to a resource provider, and the resource provider creates the new resource or modifies an existing resource. Figure 3-53 illustrates this flow and features a small sampling of the many Azure services that are available.

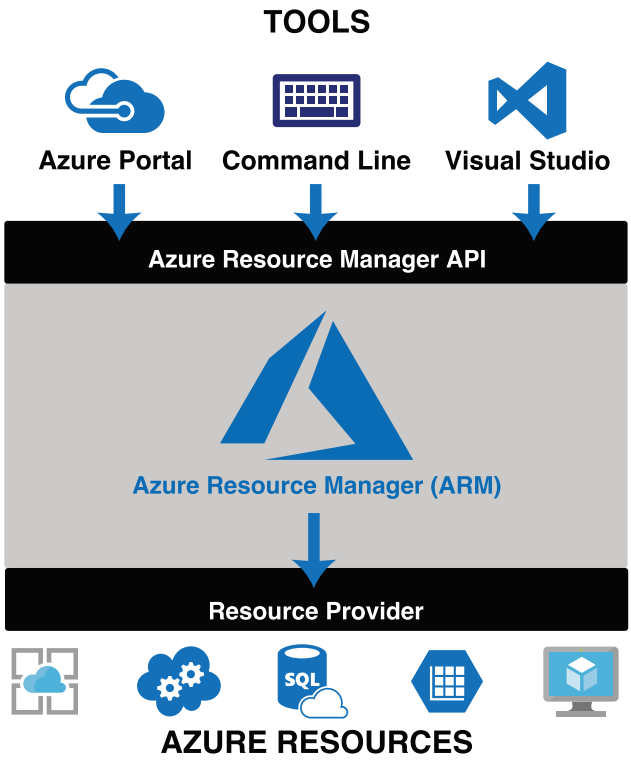


FIGURE 3-53 Azure Resource Manager

The request that is made to ARM isn't a complicated, code-based request. Instead, ARM uses *declarative syntax*. That means as a consumer of Azure, you tell ARM what you want to do, and ARM does it for you. You don't have to tell ARM *how* to do what you want. You simply tell it what you want. To do that, ARM uses files that are encoded in JavaScript Object Notation (or JSON) called *ARM templates*.

In the most basic sense, an ARM template contains a list of resources you want to create or modify. Each resource is accompanied by properties such as the name of the resource and properties that are specific to that resource. For example, if you were using an ARM template to deploy a web app in App Service, your ARM template would specify the region you want your app to be created in, the name of the app, the pricing plan for your app, any domain names you want your app to use, and so forth. You don't have to know how to set all those properties. You simply tell ARM to do it (you declare your intent to ARM), and ARM takes care of it for you.

> ***MORE INFO*** **MORE ON ARM TEMPLATES**
>
> **ARM templates are incredibly powerful, but they're also simple. If you want to read more about how to use ARM templates, check out the documentation at *https://bit.ly/az900-armtemplates*.**

There's one more important aspect to ARM template deployment. When you're deploying multiple resources (which, as pointed out, is a typical real-world scenario), you often have service dependencies. In other words, you are deploying one or more services that rely on other services already being created.

For example, think of a situation where you're deploying a certificate to be used with a web app. One of the properties you need to set on the web app is the certificate that you want to use, but if that certificate hasn't been deployed yet, your deployment will fail. ARM allows you to specify dependencies so you can avoid issues like this. You simply tell ARM that the web app depends on the certificate, and ARM will ensure the certificate's deployment is completed before it deploys the web app.

If you want to see an example of an ARM template, you can see a real-world example using the Azure portal. Open any resource and click **Export Template** in the menu, as shown in Figure 3-54. The ARM template displayed in the portal can be used to deploy that exact resource.

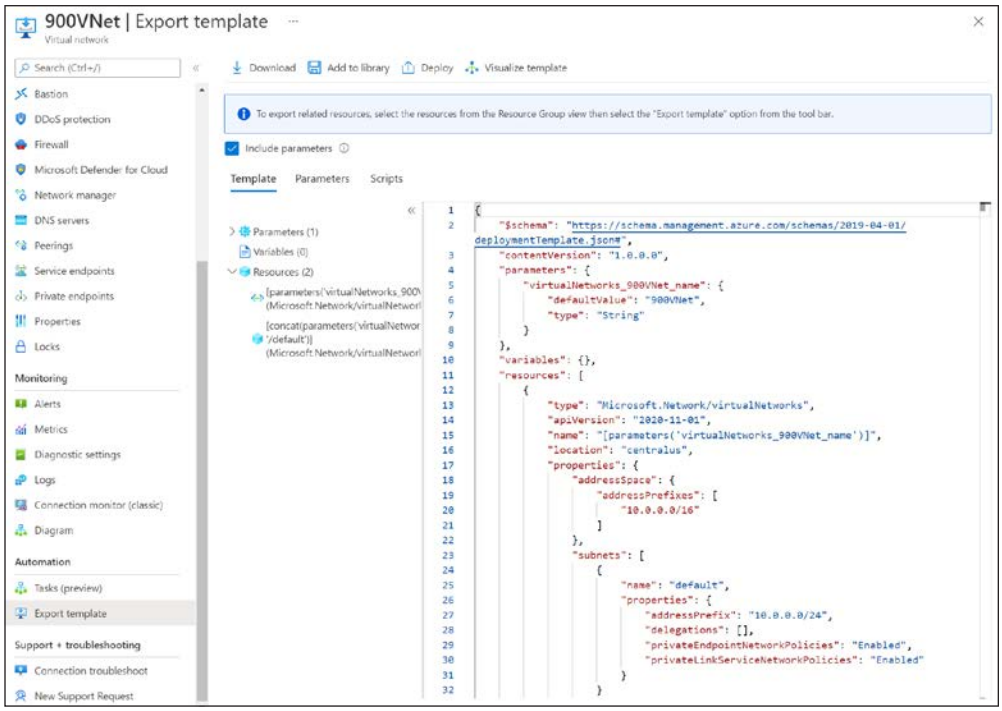


FIGURE 3-54 An ARM template in the Azure portal

Skill 3.4: Describe monitoring tools in Azure

We've covered a lot of ground related to managing and governing resources. However, a proper discussion would be incomplete without covering monitoring your resources once they're deployed.

Azure offers numerous tools for monitoring your resources and ensuring they are compliant, healthy, and operating as expected.

This section covers:

- Azure Advisor
- Azure Service Health
- Azure Monitor

Azure Advisor

Managing your Azure resources doesn't just include creating and deleting resources. It also means ensuring that your resources are configured correctly for high availability and efficiency. Figuring out exactly how to do that can be a daunting task. Entire books have been written on best practices for cloud deployments. Fortunately, Azure can notify you about problems in your configuration so you can avoid problems. It does this via Azure Advisor.

Azure Advisor can offer advice about high availability, security, performance, and cost. To access Azure Advisor, log in to the Azure portal and click **Advisor** in the menu on the left. Figure 3-55 shows Azure Advisor in the Azure portal.

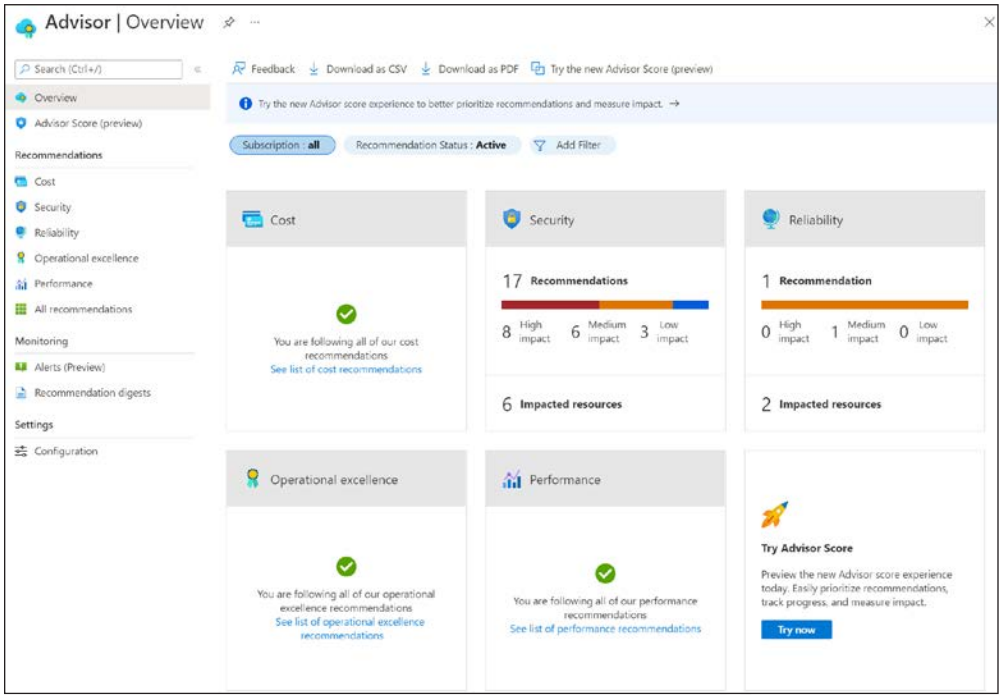


FIGURE 3-55 Azure Advisor

To review details on a recommendation, click the tile. In Figure 3-56, we have clicked the **Security** tile, and you can see a recommendation to enable MFA (multifactor authentication) and add another owner to my subscription.

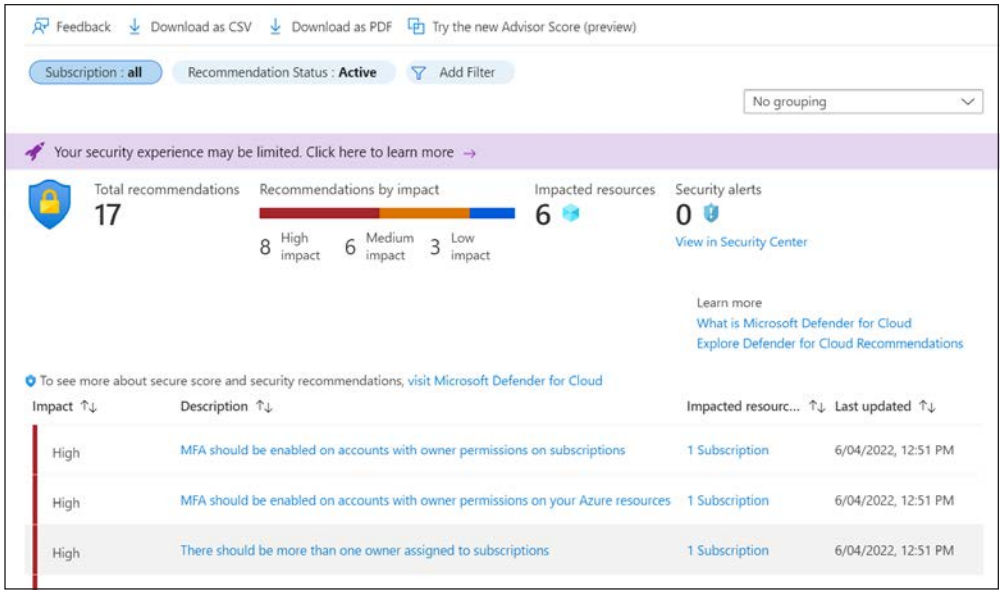


FIGURE 3-56 Advisor recommendations

You can click on a recommendation to get more details, including the steps you should take to remediate the issue. In some situations, Advisor can fix the issue for you. In Figure 3-57, I can select one or more of my VMs and let Advisor fix the issue for me by simply clicking the Fix button.

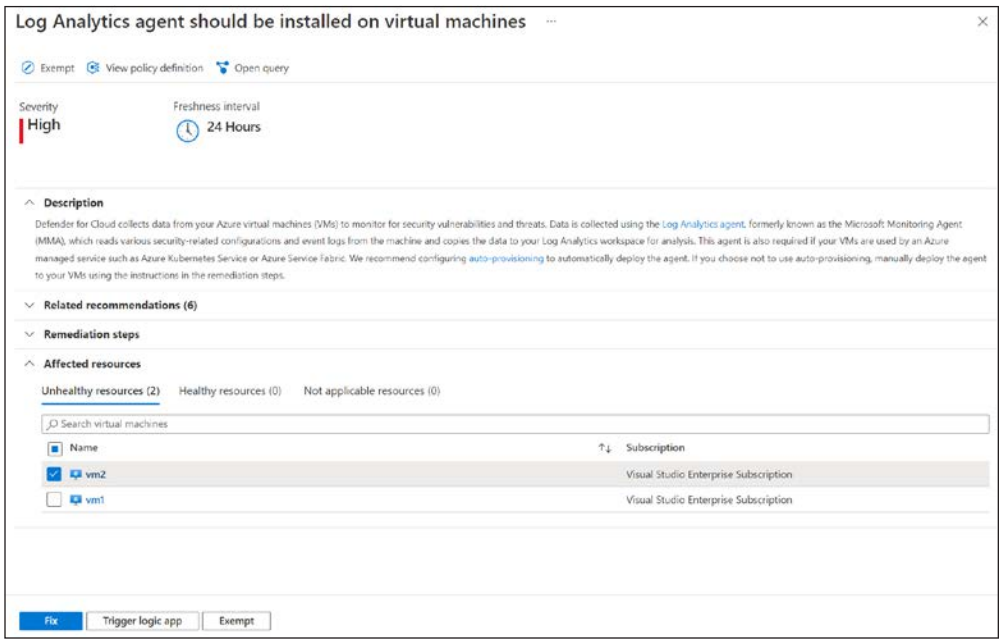


FIGURE 3-57 Acting on a recommendation

If you have many recommendations, or if you're not the right person to act on the recommendations, you can download Azure Advisor recommendations as either a comma-separated values file or a PDF. Click Download As CSV or Download As PDF, as shown previously in Figure 3-56.

Azure Service Health

Microsoft operates an Azure Status web page where you can view the status of Azure services in all regions where Azure operates. While it is a helpful view of overall Azure health, the enormous scope of the web page doesn't make it the most effective way to get an overview of the health of your specific services. Azure Service Health can provide you with a view specific to your resources.

To access Service Health, search for it in the Search bar in the Azure portal, as shown in Figure 3-58.

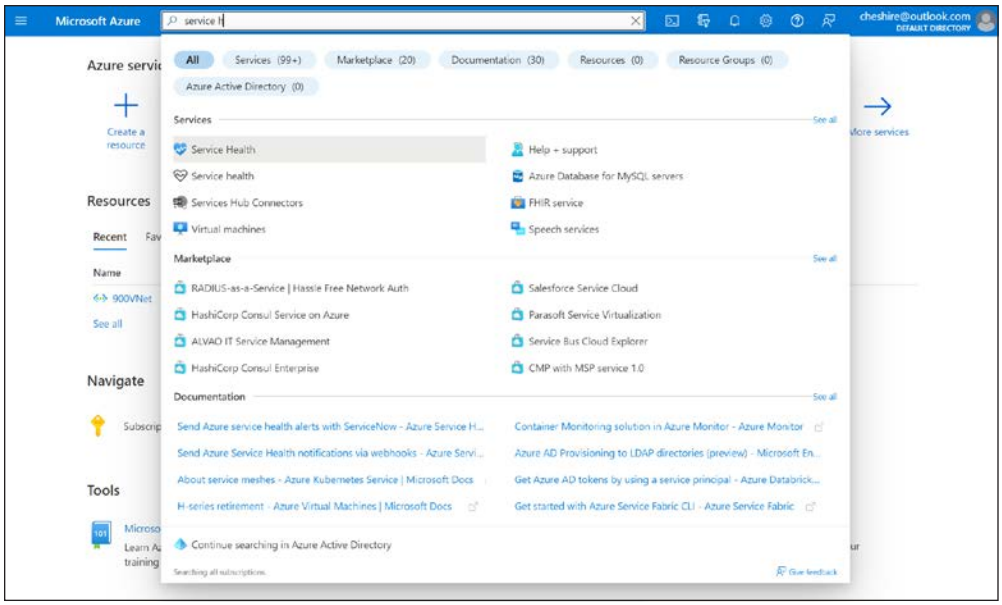


FIGURE 3-58 Azure Service Health

Figure 3-59 shows the **Service Issues** blade showing the health and status of the resources. The map shown has six green dots representing the health of the six Azure regions where my resources are deployed. (These dots might be difficult to see in print.) By clicking the pin icon, you can have a quick reference of Azure health for just the regions where you have resources.

You can also view any upcoming planned maintenance that might affect you by clicking **Planned Maintenance** in the menu on the left. By clicking **Health Advisories**, you can see health information that might be related to your own configuration and not a problem somewhere in Azure.

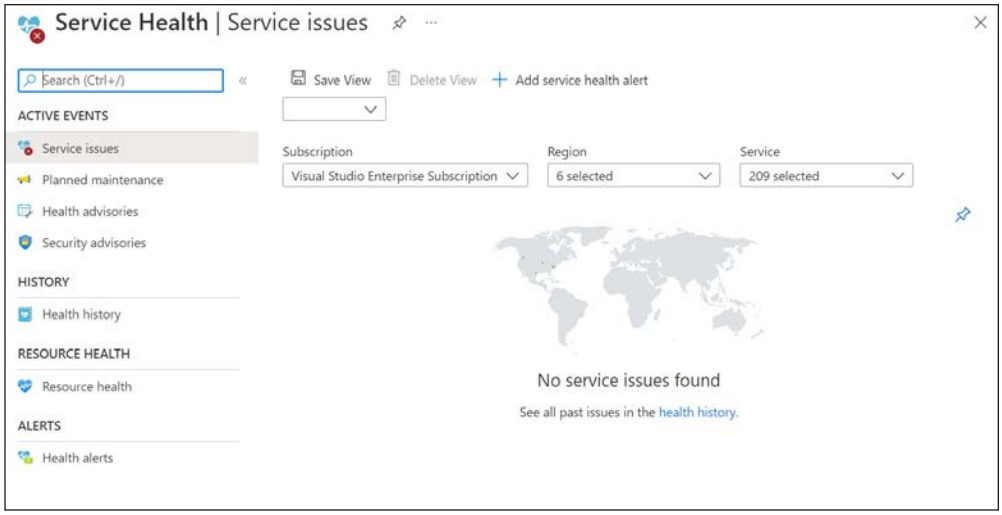


FIGURE 3-59 The Service Issues blade in Service Health

When a service issue is affecting you, you'll see details on the issue, as shown in Figure 3-60. In addition to the full details of the incident, you also see a link that refers to details on the incident. You can also download a PDF that contains an official Microsoft notice of the incident.

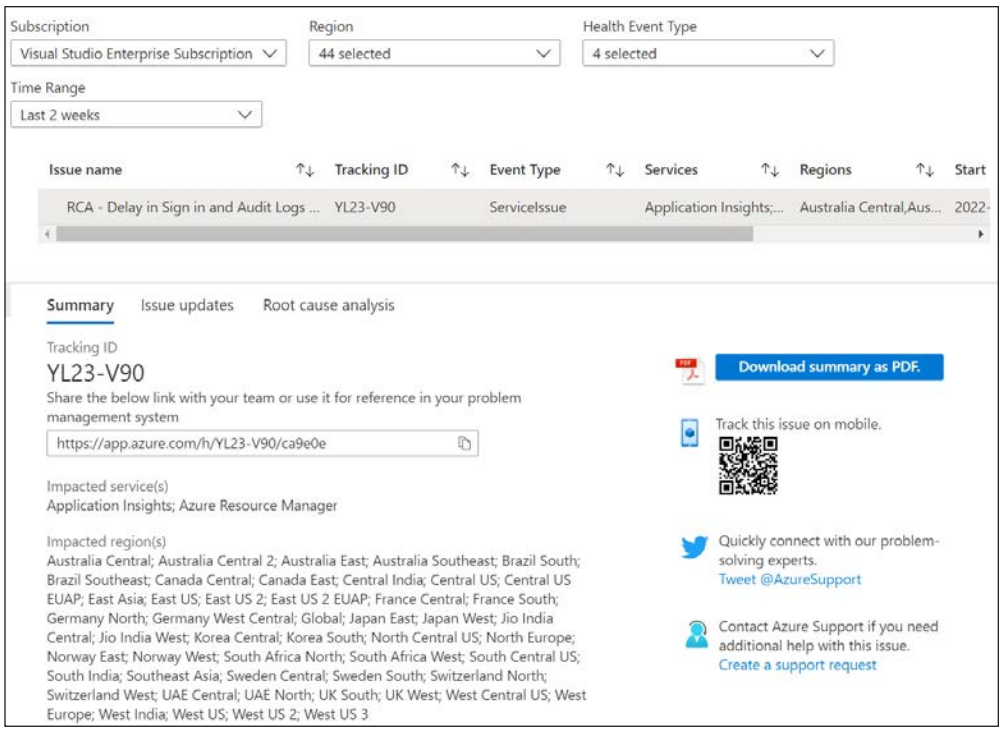


FIGURE 3-60 Azure Service Health incident

Azure Monitor

Azure Monitor aggregates metrics for Azure services and exposes them in a single interface. You can also create alerts that will notify you or someone else when there are concerns you might want to address.

To access Azure Monitor, click **Monitor** in the Azure portal to display the Azure Monitor blade, as shown in Figure 3-61. Azure Monitor is customizable, so you can see exactly what interests you the most. For that reason, it doesn't show any metrics until you configure them. To view metrics, click **Metrics** and then select a scope.

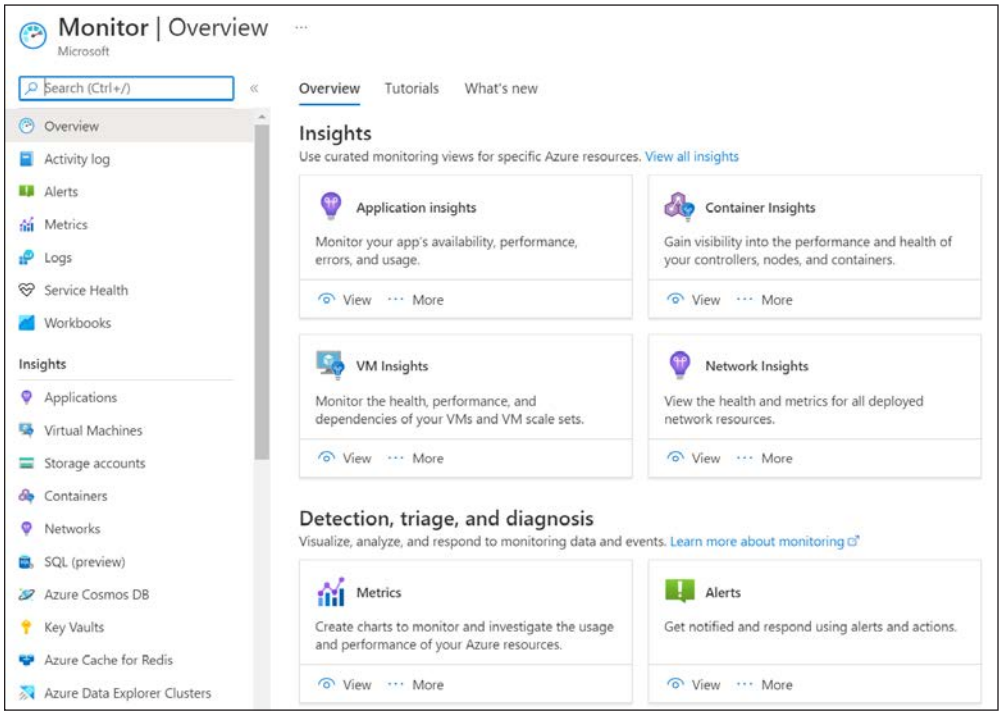


FIGURE 3-61 Azure Monitor

In Figure 3-62, a VM in the az-900 resource group has been selected for monitoring.

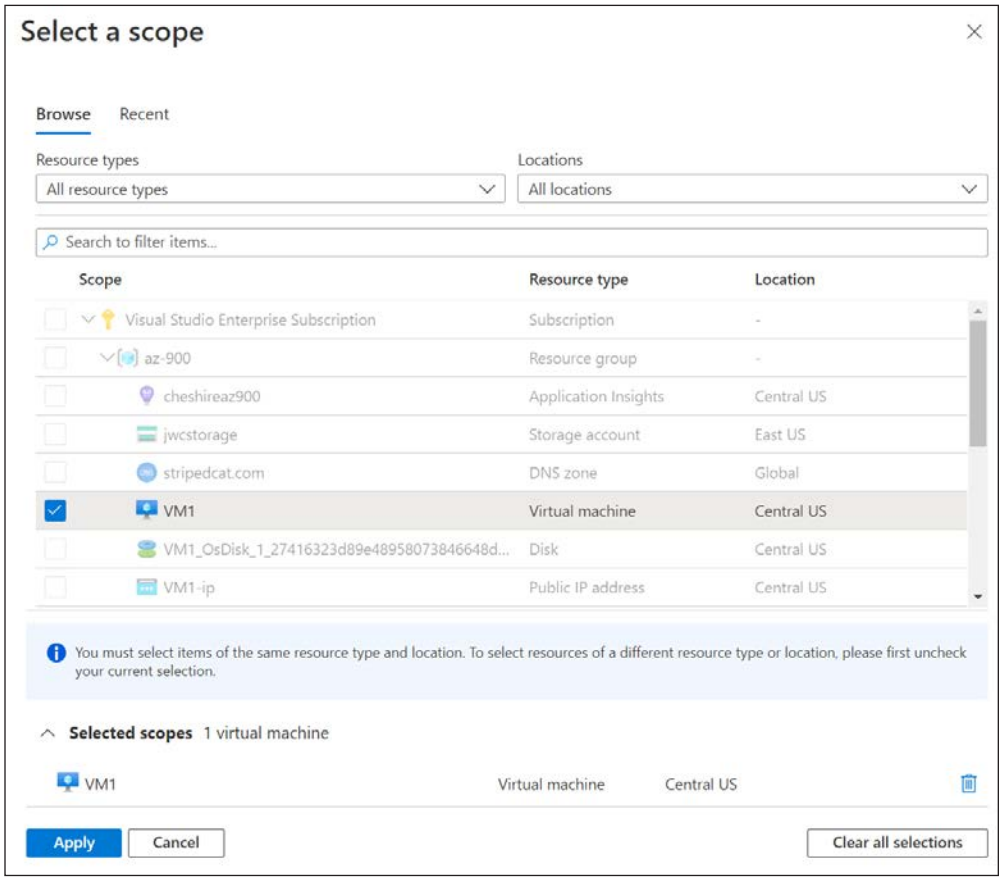


FIGURE 3-62 Selecting a resource to monitor

Once you select a resource, you are presented with a list of metrics related to that resource. Metrics for VMs are shown in Figure 3-63.

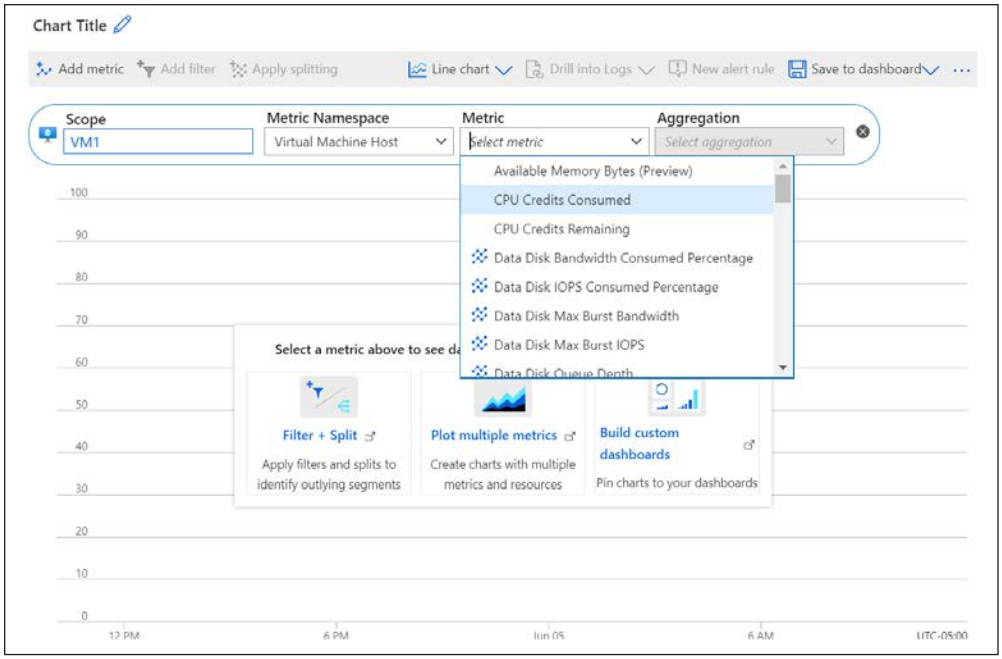


FIGURE 3-63 Metrics for VMs

When you select a metric, the chart updates to show a graph of that metric. You can add additional metrics to your chart by clicking **Add Metric**, as shown in Figure 3-64.

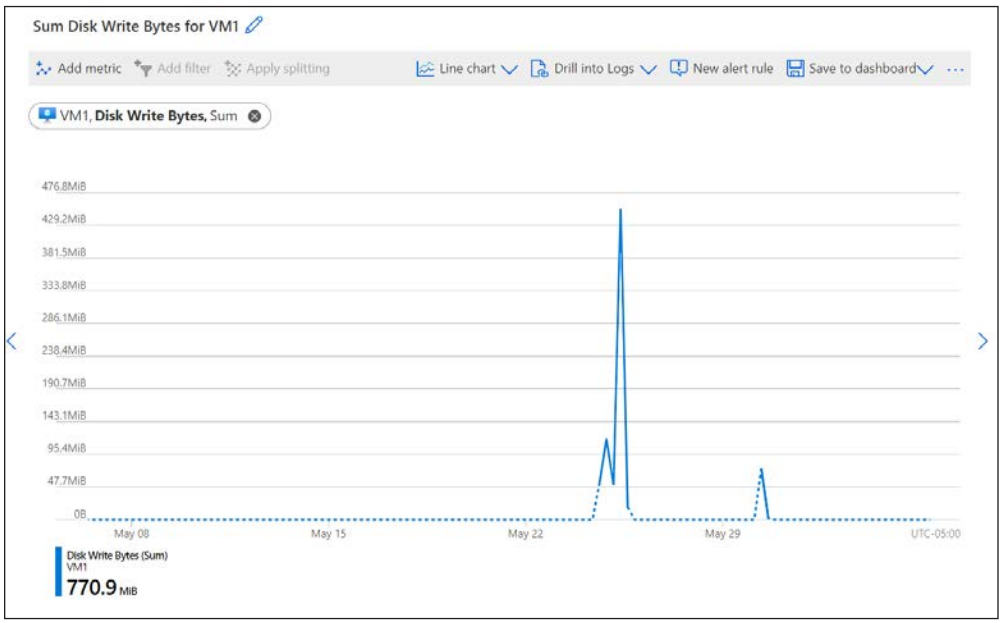


FIGURE 3-64 Monitoring VM disk usage

When adding multiple metrics, you'll want to include only those metrics that share a common unit of measurement. For example, if you were to add a CPU metric to the chart shown in Figure 3-64, it wouldn't make a lot of sense because the Percentage CPU is measured as a percentage, and disk units are measured in bytes.

In Figure 3-65, we've added **Disk Read Bytes** to the chart. Azure Monitor color codes each metric automatically to distinguish between them. We've also selected **Area Chart** as the type of chart to see the patterns more clearly.

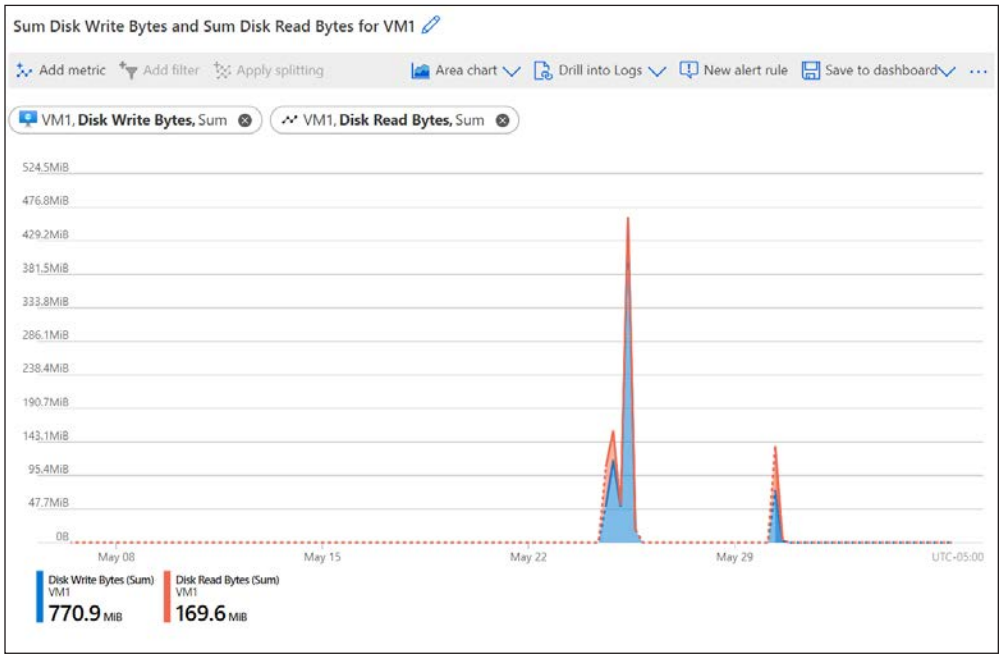


FIGURE 3-65 Chart showing disk usage

By default, charts are shown for the past 24-hour period, and the real-time value is shown at the right edge of the chart. However, you can customize the timeframe that is shown by clicking the timeframe and adjusting it as you like, as shown in Figure 3-66.

Once you have a chart that you find useful, you can pin that chart to the portal dashboard by clicking **Save To Dashboard**. As shown in Figure 3-67, you can pin your chart to a portal dashboard, pin it to Grafana (an extensible third-party metrics dashboard platform), or send it to an Azure Monitor Workbook. (Workbooks allow you to analyze data in an attractive, visual way that is sourced from multiple sources across Azure.)

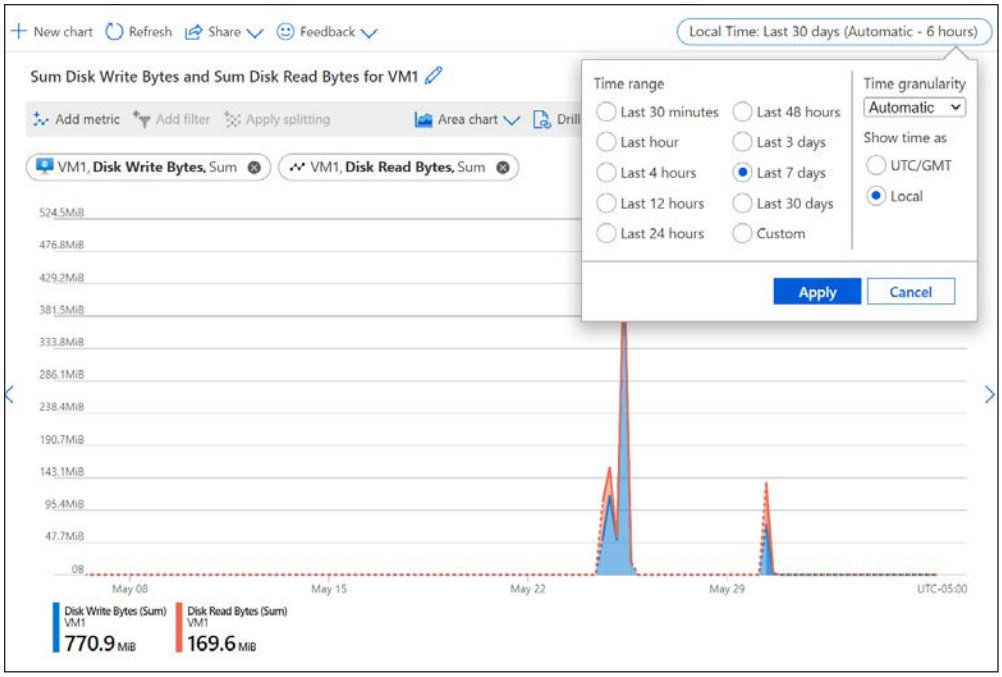


FIGURE 3-66 Changing the chart timeframe

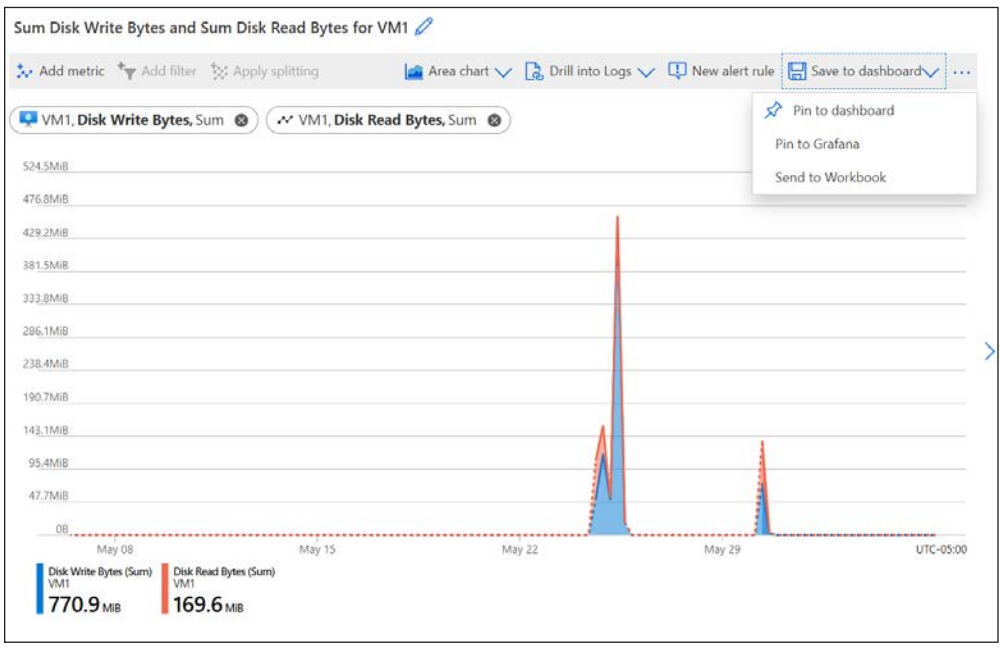


FIGURE 3-67 Saving to a dashboard

When certain conditions are met, Azure Monitor Alerts can notify you or others with an email or SMS text message, run a Logic App flow, call a Function App, make a request to a webhook, and more. Alerts are based on rules that you define. When a rule's condition is met, an alert performs the action you specify.

You can create an alert rule that is automatically configured for the metrics you've selected in your chart by clicking N**ew Alert Rule** at the top of your chart. You can also start from scratch by clicking **Alerts** in the menu for Azure Monitor, clicking **Create**, and clicking **Alert Rule**, as shown in Figure 3-68.

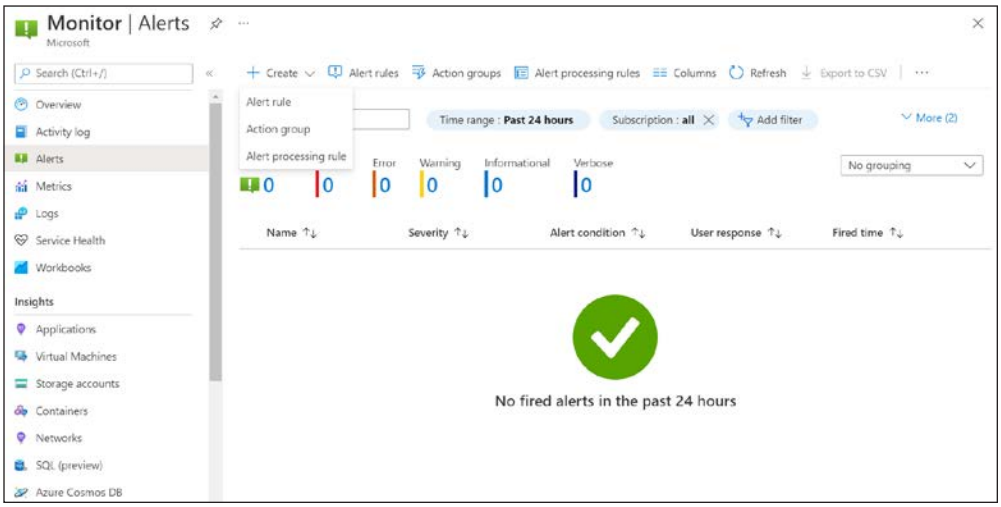


FIGURE 3-68 Creating an alert rule

To start your rule, click the Filter By Resource Type dropdown, enter the resource type, and select the resource for which you want to configure an alert. In Figure 3-69, a VM is selected for a new alert rule. When you've selected the desired resources, click the **Done** button.

Next, you'll need to specify the condition for your alert. Click the **Condition** tab, and then select the signal you want to monitor for your alert. In Figure 3-70, an alert has been configured based on the Percentage CPU signal of the VM.

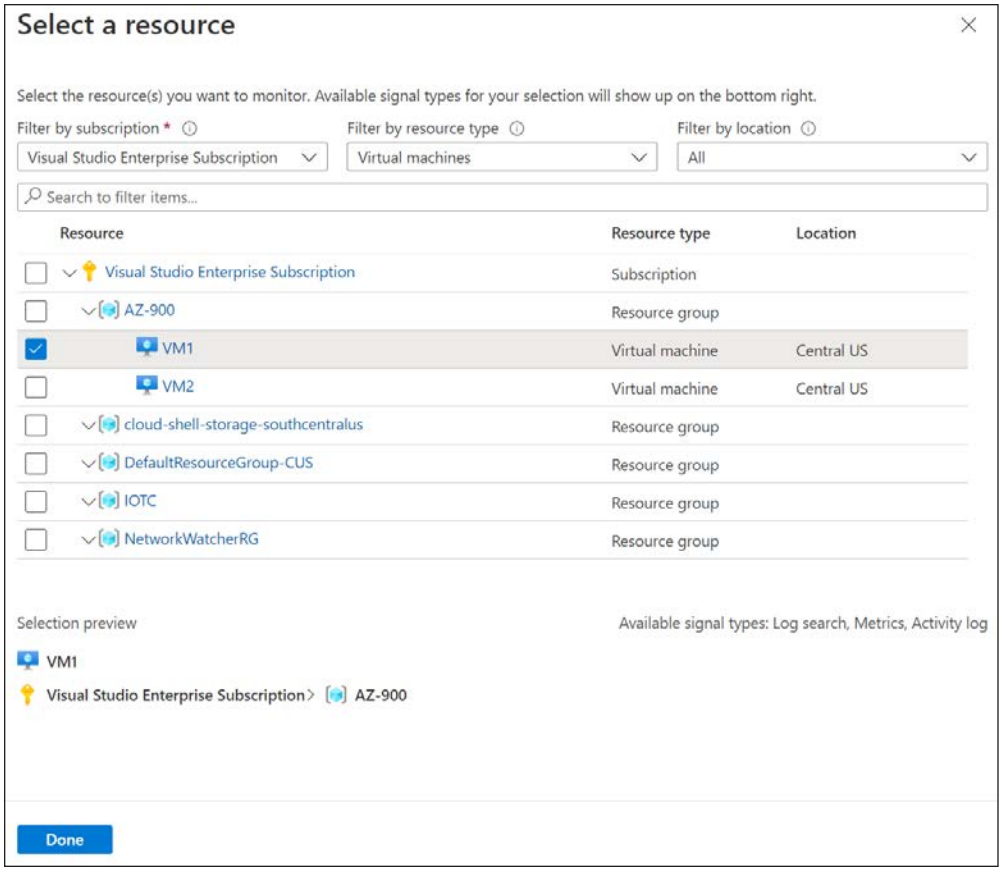


FIGURE 3-69 Selecting a resource for an alert

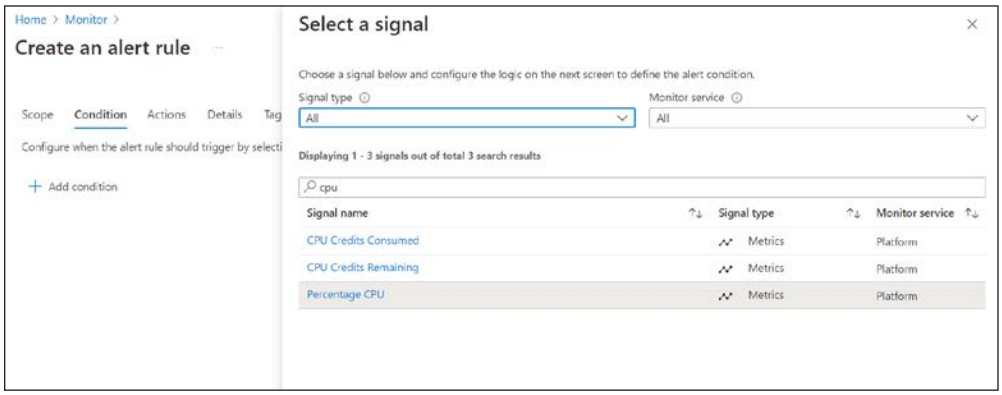


FIGURE 3-70 Configuring a condition

Once you select a signal, the logic for the signal is configured. As shown in Figure 3-71, Monitor displays an interactive graph of the signal you've chosen, which helps you get a feel for how your resource has been performing historically. By default, this shows the last six hours, although you can adjust the chart period. You can specify an operator, aggregation type, and threshold, or you can click **Done** to create the logic for the alert.

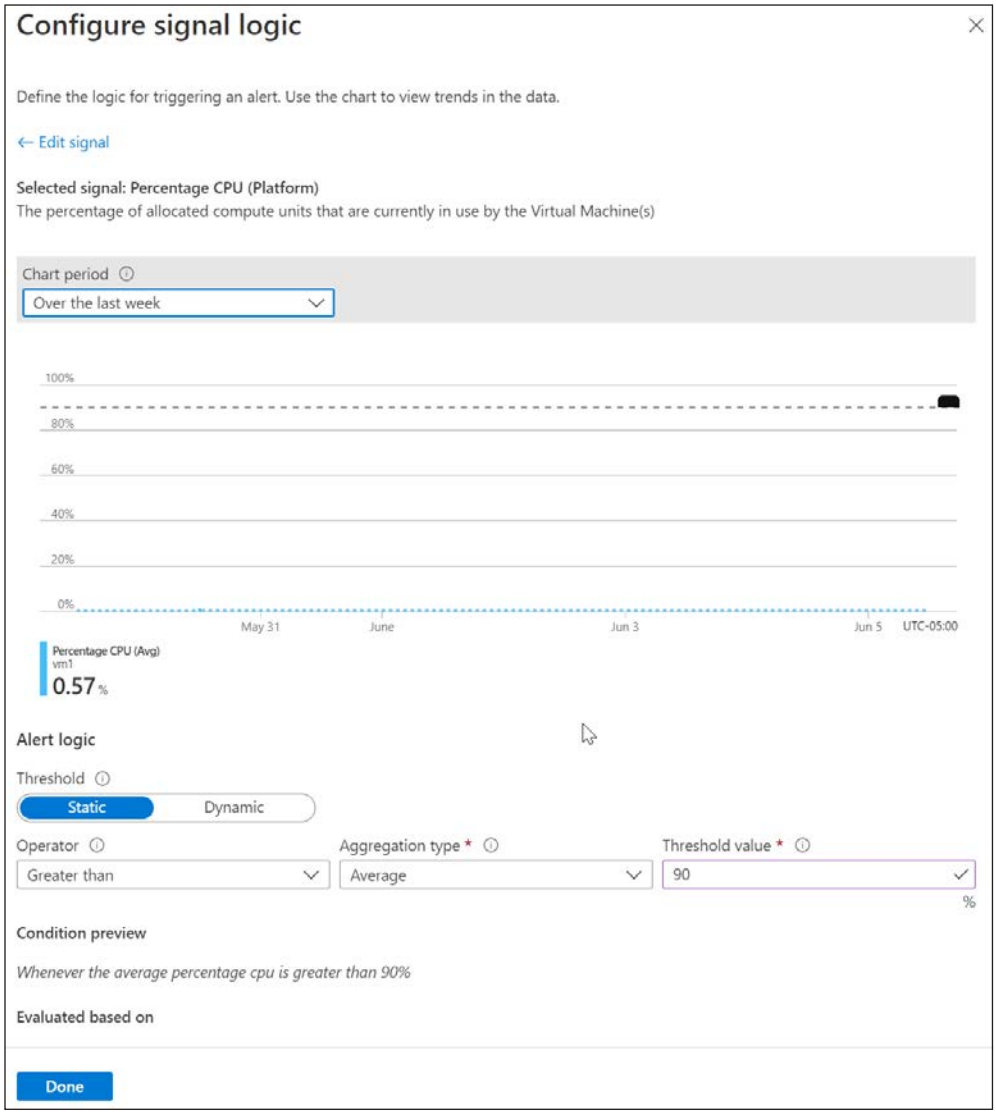


FIGURE 3-71 Alert rule logic

NOTE **MULTIPLE CONDITIONS**

An alert rule can consist of multiple conditions. For example, you can have a rule that only triggers if CPU averages above 70 percent and disk usage is also high. The choice is yours.

When an alert is triggered, it performs an action that you specify using an *action group*. An action group contains notifications and actions to take when an alert is triggered. To create a new action group, click the **Actions** tab and click **Create Action Group**, as shown in Figure 3-72.

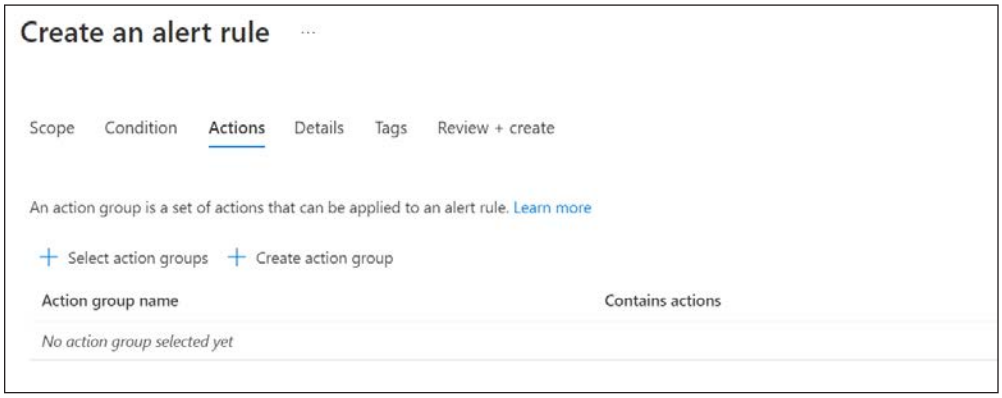


FIGURE 3-72 Creating an action group

In Figure 3-73, we are creating a notification to notify the IT director. In this case, the notification will send a text message to the IT director, and it will also send a push notification using the Azure mobile app.

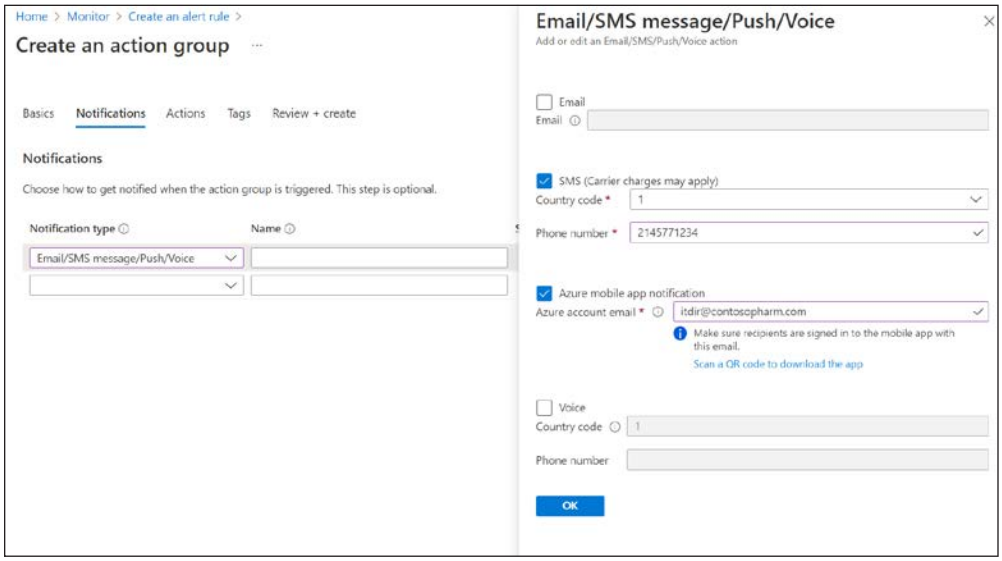


FIGURE 3-73 Creating a notification

You can add an action to your action group by clicking on the Actions tab and selecting an action from the Action Type dropdown, as shown in Figure 3-74. In this case, an action to call a webhook is being configured.

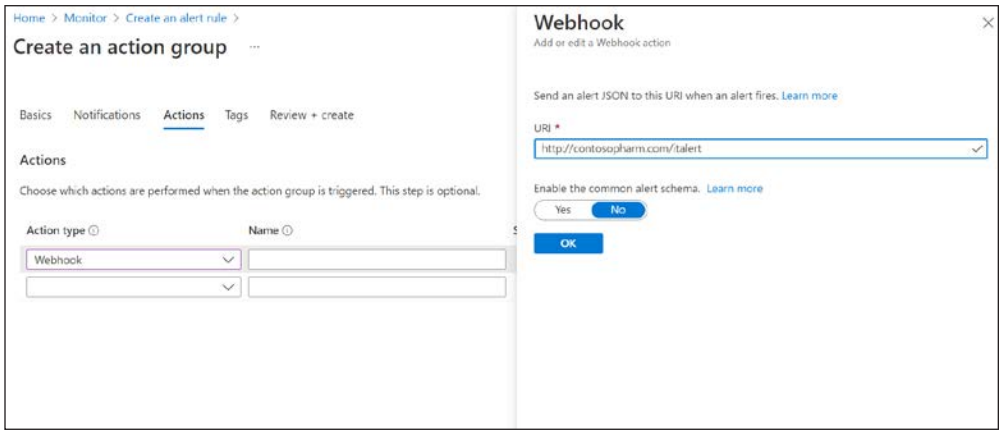


FIGURE 3-74 Adding another action

Azure Monitor includes a feature called Application Insights that provides a dashboard of metrics for your Azure web apps and VMs with only minimal configuration. When you create a web app, you have the option of enabling Application Insights automatically. If you want to use Application Insights with a VM, you can enable it easily from the Insights blade in the portal.

Application Insights monitors a range of metrics automatically. In web apps, it monitors request rates and response times, errors (exceptions), page views and load times, user and session counts, and more. In VMs, it monitors performance counters, diagnostic logs, and more.

There are numerous ways you can view data from Application Insights. You can view VM data using the Insights menu when viewing the VM in the portal. Figure 3-75 shows the CPU utilization and available memory over time for a VM.

To view Application Insights data for a web app, open the Application Insights resource in the portal. In Figure 3-76, metrics for a web app are shown, including Server Response Time and the number of requests (Server Requests).

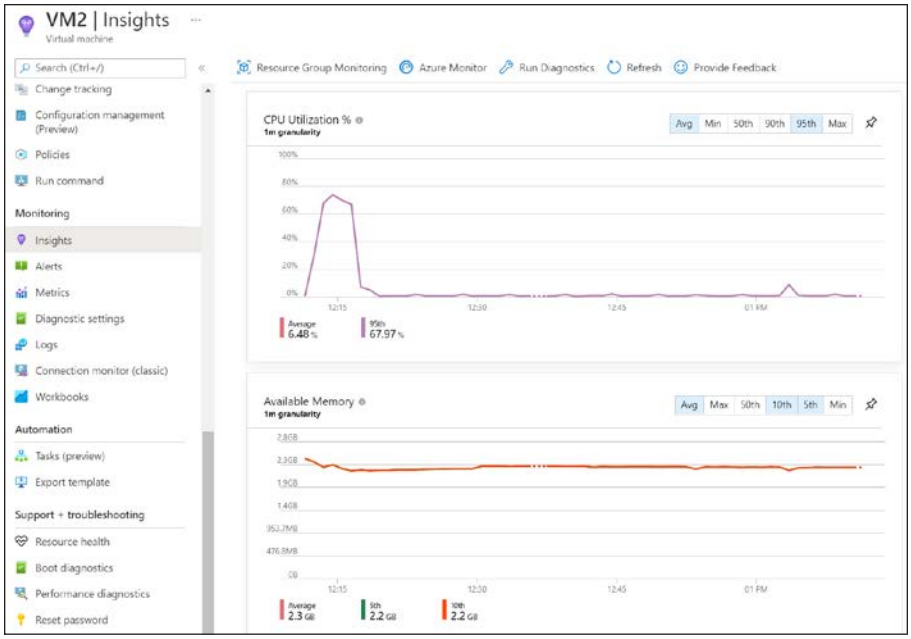


FIGURE 3-75 Insights on a VM in the portal

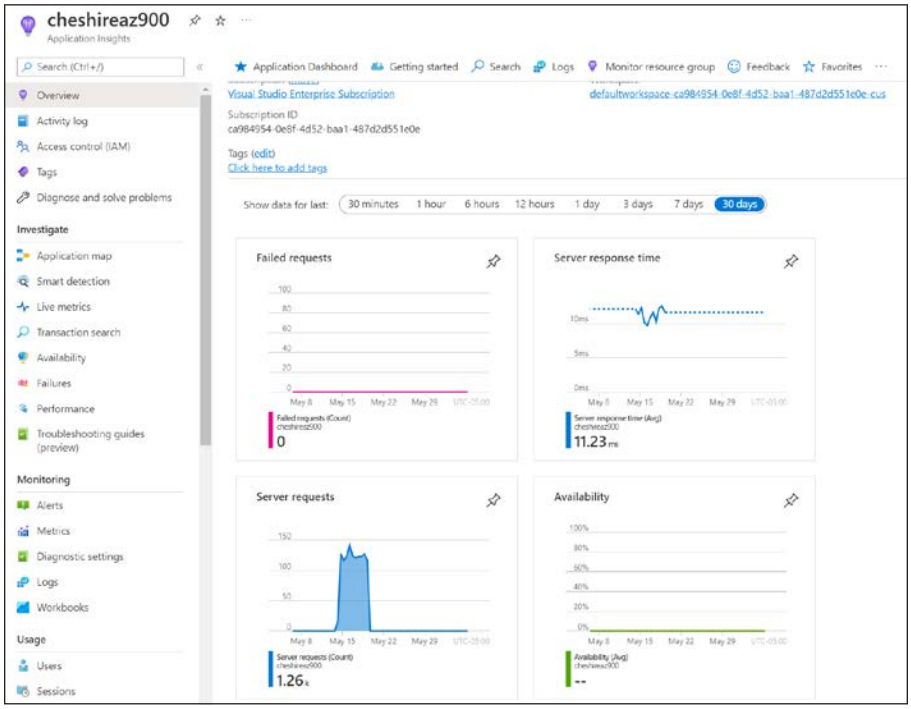


FIGURE 3-76 Application Insights for a web app

Data from Application Insights is also stored in a Log Analytics workspace. Log Analytics is an Azure feature for aggregating data from Azure Monitor and making it available via queries. Log Analytics queries are written using the Kusto Query Language (KQL), but you can also choose from a collection of queries included with Log Analytics.

> ***NOTE*** **KQL AND LOG ANALYTICS**
>
> **KQL and Log Analytics are optimized for querying enormous amounts of data. In fact, Microsoft uses both to analyze the vast amounts of data from the Azure infrastructure. Also, Microsoft support staff use KQL when troubleshooting complex issues in Azure.**

To access Log Analytics, click **Logs** from your Application Insights resource. As shown in Figure 3-77, a list of existing queries is displayed. You can click the category of query to scroll to queries for that category. When you find the query you are interested in, click the **Run** button to run that query in Log Analytics. You can also hover over the query and click Load To Editor to load the query into the editor without running it. This is useful if the query might be long-running, and you prefer to edit it before running it.

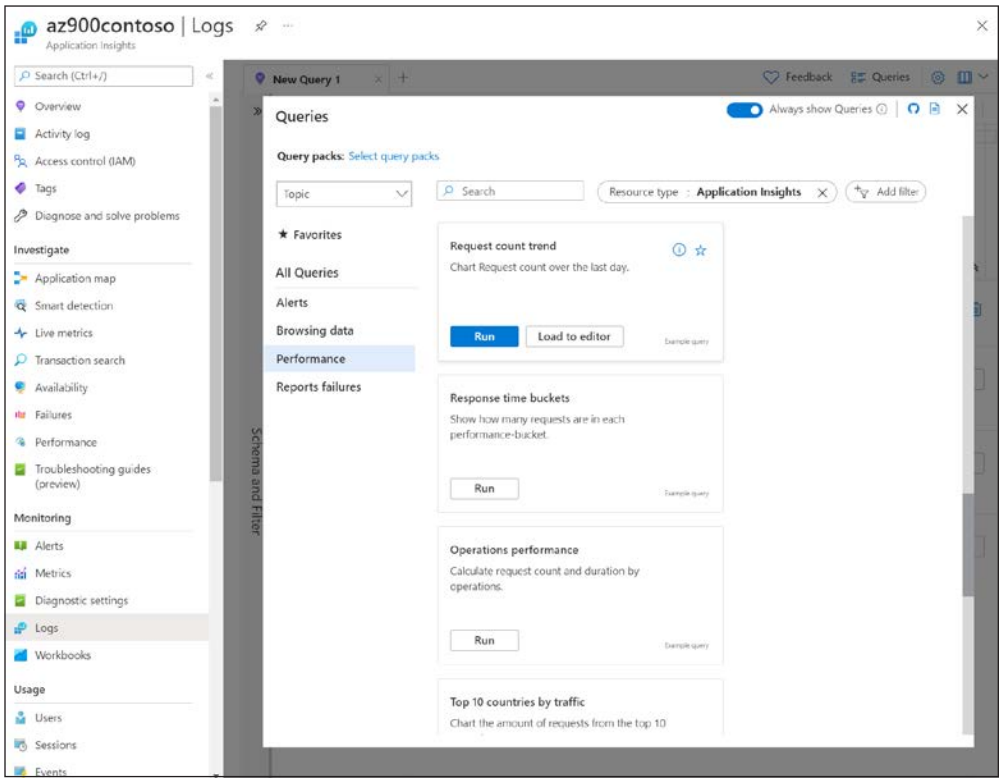


FIGURE 3-77 Log Analytics

In Figure 3-78, you see the Request Count Trend query results in the Log Analytics editor. At the top of the screen, you see the KQL query text, and the bottom part of the screen shows the results. In this case, the results are rendered as a line graph (called a timechart in KQL) so that you can visualize the results of the query.

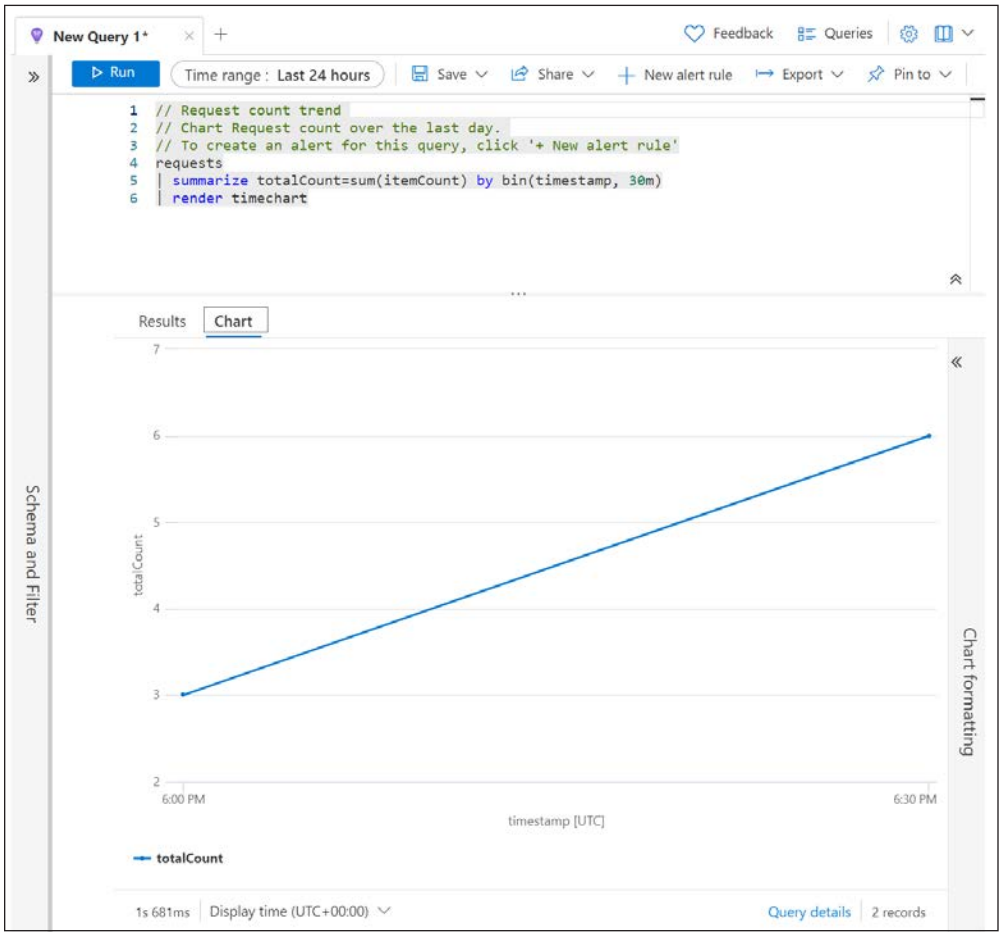


FIGURE 3-78 Results of a query

The KQL in the Log Analytics editor is editable, so you can tweak the query if necessary to get to the data you need. Once you've got the query you want, you can save it by clicking the **Save** button. You can also export the query results by clicking the **Export** button.

MORE INFO **KQL REFERENCE**

If you're interested in learning more about using KQL, you can find the KQL reference at *https://bit.ly/az900-kql*.

Thought experiment

Let's wrap up this chapter with a thought experiment to help test your knowledge of the material we covered. The answers to this thought experiment are in the section that follows.

Forecasting expenses

So far, your record with Contoso Medical Group is positive. You've convinced the company that they need to move some of their deployments to Azure. As they plan that migration, the IT director needs to forecast the expenses for moving to the cloud. The IT director has asked you to provide an estimate of cloud expenses based on the company's current plan. What tools can you use to provide the estimate to the IT director?

Categorizing expenses

The IT director has also been speaking to the CMG accounting office, and they've explained that they need an accurate breakdown of cloud expenses from each internal organization at CMG to accurately report to the CFO each month. What is your recommendation to provide this kind of breakdown of expenses?

Applying governance to resources

The IT director has been partnering with one of the application architects, and they've planned out the entire deployment. The architect has informed the IT director that as the number of users grows, they'll need to add additional resources to keep performance high. Those resources include VMs, some ARM templates, and other Azure resources that must be configured in a specific way. The architect wants to ensure that when new resources are deployed, they are configured according to her specifications.

What tools would you recommend to the architect to achieve these results?

Preventing the deletion of resources

The application also relies on a database that will be running in Azure. The database administrator needs some users to have elevated access to the database configuration in the Azure portal, but they are concerned that doing so runs the risk of someone accidentally deleting the database. They realize that the risk is low, but the result of that would be devastating, so they must account for that.

How would you ease the mind of the database administrator and ensure that no one can delete the database?

Effective deployment of Azure resources

One of the developers of the application has written an ARM template for deploying his part of the app. However, they need to deploy more than a dozen instances of the deployment defined in his template. It takes a while to deploy the resources from the Azure portal, and the developer is interested in any option they might have for automating the deployment. The developer tells you that they have experience with many different scripting languages, but they're not sure how to leverage that experience for this use. One other point the developer has raised is that they need a solution that will work for them even when using his iPad.

What recommendations would you make to the developer?

Monitoring application performance

The web administrator has designed a web app that will act as the user interface for a critical part of the application. The admin is planning on running this app in Azure App Service. The load on this app will be variable depending on the time of the month, and when the load is high, it's critical that the admin's team keeps a close eye on performance metrics. However, the admin doesn't want to pay overtime for people to constantly monitor the app.

What recommendation would you make to the web administrator?

Reporting on the health of resources

The IT director has one more concern. If the application does encounter a problem, they need to know quickly if the problem is a result of something in Azure and not their application. If it is something in Azure, the IT director needs to have documentation so that they can report it to the CTO during their business reviews.

What tool should you recommend to the IT director?

Thought experiment answers

Here are the answers to the thought experiment.

Forecasting expenses

The IT director needs to forecast expenses and get an estimate of cloud expenses. There are a few tools that can help with this.

The pricing calculator can help calculate the cost of Azure resources based on where they're deployed and how much usage you anticipate. Once you generate an estimate, you can save that and share it with the IT director.

The total cost of ownership (TCO) calculator can help show you how much you'll save by moving to Azure. One thing that it can show that might interest the IT director is how much you'll save in areas like IT staff and infrastructure. This might help offset any concerns raised by the cost of Azure resources, especially because it can show you how much money you'll save over time.

Azure Cost Management and Billing may also help to alleviate any concerns because you can use it to easily keep an eye on costs and to create budgets so you can be alerted if your costs start to reach a configured threshold.

Categorizing expenses

The accounting office has said they need a way to have an accurate breakdown of expenses from each internal organization at CMG. By using tags on resources and tagging them with the organization's name, the accounting office can easily meet this requirement. Because tags are shown on the Azure invoice, they are an excellent way to categorize cloud expenses.

Applying governance to resources

The IT director and architects want to ensure that new resources are deployed according to the exact specifications. ARM templates can certainly help with reliable and predictable deployments, and by using Azure Policy, you can ensure that deployments and their configuration meet company requirements. Also, you might want to recommend that they package their policies and templates into Azure Blueprints. This will allow them to repeat the exact deployments with policies attached as the application grows.

Preventing the deletion of resources

The database administrator must ensure that no one can delete the database. You might recommend RBAC to control this, but the admin has also said they need to give users a high level of access. To avoid problems with role assignments, you decide to recommend a delete lock on the database resource. This will allow the access levels the database administrator needs while ensuring that no one can delete the database.

Effective deployment of Azure resources

One of the developers needs to automate the deployment of many resources, and they'd like a solution that can be used on an iPad as well as a computer. You recommend that the developer use either the PowerShell `Az` module or the Azure CLI. Both can be scripted, allowing for easy deployment of his resources. Because both PowerShell and the CLI are available in Azure Cloud Shell, the developer can use the Azure mobile app on an iPad to run either. Because Azure Cloud Shell can store scripts in Azure Storage, they'll be available to the developer regardless of whether they're using a computer or mobile device.

Monitoring application performance

The web administrator needs to keep an eye on the performance metrics of the app. You recommend that they enable Application Insights for the app. Doing so will also allow the web admin to create alerts in Azure Monitor so that they can automatically alert the right people if the metrics they are monitoring fall outside of configured values.

Reporting on the health of resources

Finally, the IT director wants to make sure that he can report on any application problem that is a result of an Azure incident. Azure Service Health is the perfect solution for this. Not only does it give them the reporting they need, but they can download a PDF of any incident to share with the CTO during their business reviews.

Chapter summary

This chapter covered many topics related to costs, governance, management, and reporting. Here's a summary of everything we covered:

- Your selection of Azure regions can impact costs due to differing costs in billing zones.
- Cloud solution partners (CSP) can sometimes save money on Azure deployments by selling complete solutions.
- Azure Reservations can help save costs by committing to resource usage over time.
- Reserved capacity pricing on database solutions can save money.
- VMs and Azure SQL Database can benefit from the Azure Hybrid Benefit where you bring your own license for Windows or SQL Server.
- Azure Spot VMs make it possible to run small workloads on Microsoft's unused server capacity to save money.
- The pricing calculator allows you to create an estimate of Azure expenses for various services.
- The total cost of ownership (TCO) calculator shows you how much money you can save in Azure versus on-premises.
- Azure Cost Management and Billing makes it easy to keep an eye on expenses, and you can create budgets to control costs.
- Tags are name/value pairs that you add to Azure resources, and because they show up on your invoice, using them is a great way to categorize expenses.
- Azure Blueprints make it easy to package resources, ARM templates, policy assignments, and role assignments in a blueprint that can be reused.
- Azure Policy provides governance of corporate policies, which can respond to a breach in policy using several different effects.

- Resource locks can prevent an Azure resource from being changed or deleted.
- The Service Trust Portal provides information and tools on trust, security, and compliance.
- The Azure portal is a customizable, web-based interface for creating and managing Azure resources.
- Azure PowerShell uses the Az module to create and manage Azure resources at a command line.
- The Azure command-line interface (CLI) is a command-line tool for creating and managing Azure resources.
- Azure Cloud Shell provides easy access to Azure PowerShell and the CLI from any web browser and from the Azure mobile app.
- The Azure Arc service extends Azure management and governance tools to non-Azure resources.
- Azure Resource Manager (ARM) is the key interface used by all Azure management tools.
- ARM templates are declarative JSON files that ensure predictable Azure deployments.
- Azure Advisor provides analysis and recommendations for cost, security, reliability, operational excellence, and performance of your Azure resources.
- Azure Service Health shows you all current and historic incidents in Azure that impacted your resources.
- Azure Monitor is a monitoring platform that includes powerful tools like Application Insights and Log Analytics to help monitor web apps and VMs.
- Azure Monitor alerts make it easy to react appropriately to metrics.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S