Microsoft

Microsoft Azure Fundamentals

SECOND EDITION

Exam Ref AZ-900

Jim Cheshire

Microsoft

Exam Ref AZ-900 Microsoft Azure Fundamentals

Jim Cheshire

Exam Ref AZ-900 Microsoft Azure Fundamentals

Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.
Hoboken, NJ

ISBN-13: 978-0-13-687718-9
ISBN-10: 0-13-687718-4

Library of Congress Control Number: 2020940854

ScoutAutomatedPrintCode

CREDITS

EDITOR-IN-CHIEF
Brett Bartow

EXECUTIVE EDITOR
Loretta Yates

ASSISTANT SPONSORING EDITOR
Charvi Arora

DEVELOPMENT EDITOR
Rick Kughen

MANAGING EDITOR
Sandra Schroeder

SENIOR EDITOR
Tracey Croom

COPY EDITOR
Rick Kughen

INDEXER
Cheryl Ann Lenser

PROOFREADER
Abigail Manheim

TECHNICAL EDITOR
Tim Warner

EDITORIAL ASSISTANT
Cindy Teeters

INTERIOR COMPOSITOR
codeMantra

COVER DESIGNER
Twist Creative Seattle

I dedicate this book to my wife, Becky, my daughter, Hope, and my son, James.

—Jim Cheshire

This page intentionally left blank

Contents at a glance

This page intentionally left blank

Contents

This page intentionally left blank

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 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, 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 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.

This page intentionally left blank

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. Jim is heavily involved in Azure and is in his 22nd year at Microsoft. He's currently working on the design and implementation of the training ecosystem used to train Microsoft support engineers. You can follow Jim and interact with him on Twitter at @az900examref.

This page intentionally left blank

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 that are specific to a particular Azure service. 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, and there's a chance that some of the screens in the Azure portal are slightly different now than they were 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 edition 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 edition.

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 Learning website: *http://aka.ms/examlist*. 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 six major topic areas, this book contains six 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 *https://MicrosoftPressStore.com/ ExamRefAZ900SecondEdition/downloads*

The URLs are organized by chapter and heading. Every time 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:

Stay in touch

Let's keep the conversation going! We're on Twitter:

This page intentionally left blank

Skill 2.1: Describe the core Azure architectural components

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 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 who doesn't use Google to look for things on the Internet? Facebook 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.

Facebook 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 many 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
- Availability zones
- Resource groups
- Azure subscriptions
- Management groups
- Azure Resource Manager (ARM)

Azure 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 has to 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 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 regional 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. We'll cover more about this later in this chapter.

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**

As of 2018, all Microsoft's datacenters were using at least 50 percent natural power consisting of solar power, wind power, and so on. In 2020, the goal is 60 percent, and the long-term goal is to use 100 percent sustainable power.

In order 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.

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. In order to ensure that applications are still performing as quickly as possible, Microsoft guarantees round-trip network performance of 2 milliseconds or less between regions.

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 at a particular region. However, it's also important that data and applications maintain availability when a problem occurs at a particular datacenter 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 because each availability zone exists within its own datacenter in that region, each 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 housing

one zone, you will benefit from availability zones. Because availability zones are located 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.

Currently, availability zones are supported with the following Azure services.

- Windows virtual machines
- Linux virtual machines
- Virtual Machine Scale Sets
- Azure Kubernetes Service
- Managed disks
- Zone-redundant storage
- Standard Load Balancer
- Standard IP address
- VPN Gateway
- ExpressRoute Gateway
- Application Gateway V2
- Azure Firewall
- Azure Data Explorer
- Azure SQL Database
- Azure Cache for Redis
- Azure Cosmos DB
- Event Hubs
- Service Bus (Premium tier)
- Event Grid
- Azure AD Domain Services
- App Service Environments ILB

NOTE **KEEP UP WITH CHANGES IN AZURE**

You can keep up with all the news related to Azure updates by watching the Azure blog at *https://azure.com/blog*.

By deploying your service to two or more availability zones, you ensure the maximum availability for that resource. In fact, Microsoft guarantees an SLA of 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.

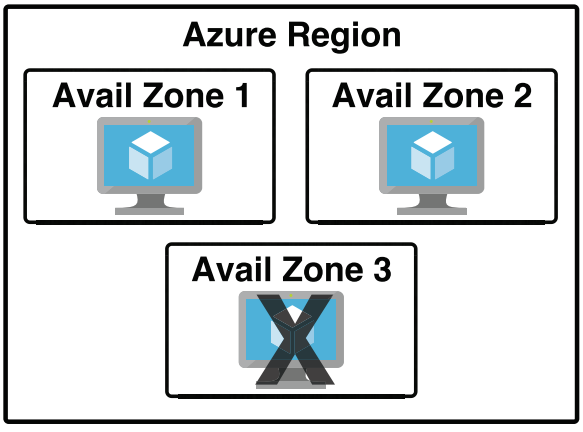


FIGURE 2-1 Azure virtual machine inside of three availability zones

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 two or more Azure services into two distinct datacenters 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. In order 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.

Resource groups

You should now be realizing that moving to the cloud might not be as simple as it first seemed. Creating a single resource in Azure is pretty simple, but when you're dealing with enterprise-level applications, you're usually dealing with a complex array of services. Not only that, but you might be dealing with multiple applications that use multiple services, and they might be spread across multiple Azure regions. Things can certainly get chaotic quickly.

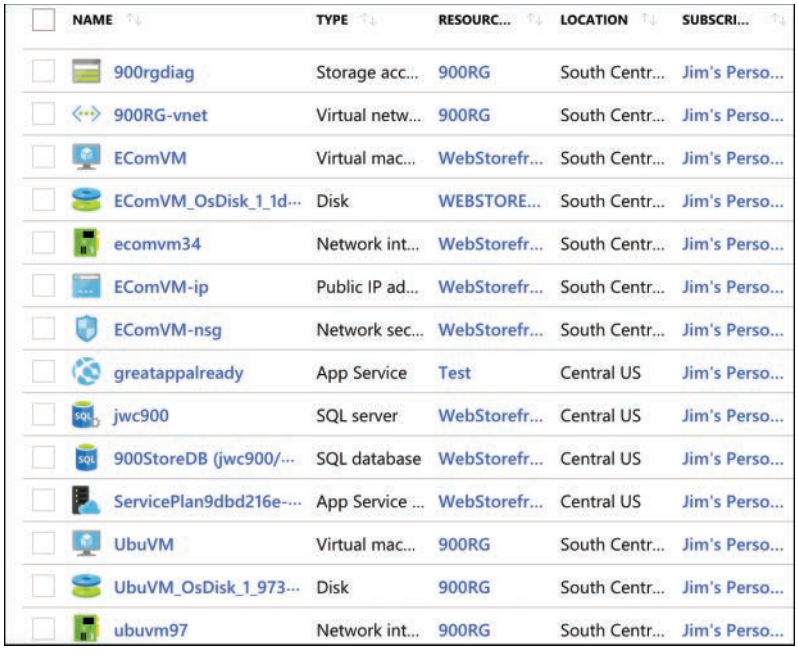
Fortunately, Azure provides a feature that helps you deal with this kind of problem: the resource group. A resource group is a logical container for Azure services. By creating all Azure services associated with a particular application in a single resource group, you can then deploy and manage all of those services 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 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 later in this chapter when we discuss Azure Resource Manager.**

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 actually start deploying Azure resources and realize that you have many more resources than you first thought. For example, when you create an Azure virtual machine, Azure creates not only a virtual machine, but it also creates a disk resource, network interface, public IP resource, and network security group. 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 in order 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.

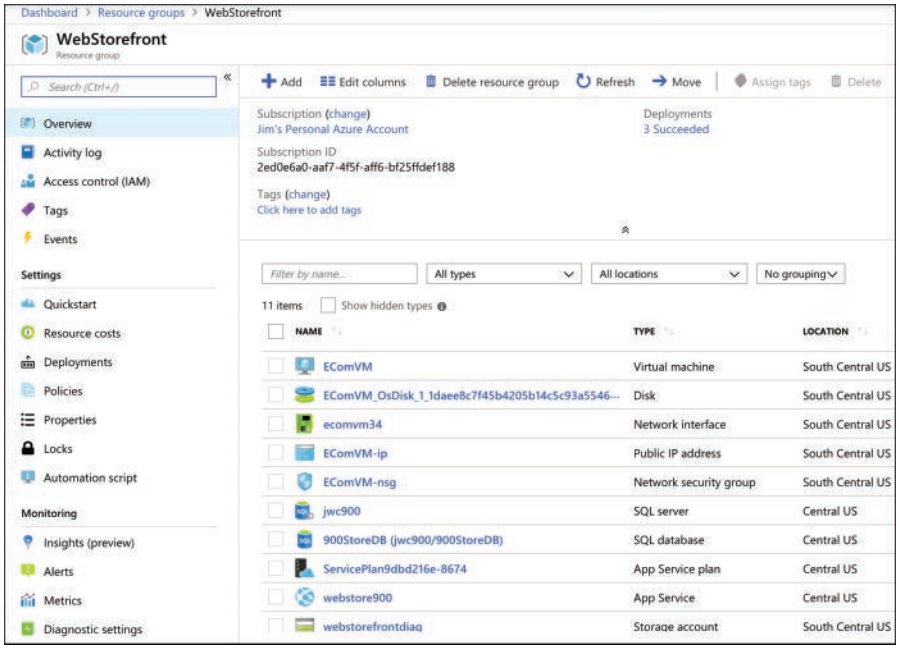


FIGURE 2-3 An Azure resource group

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 their used resources. 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 also 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 all limits for subscriptions 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.

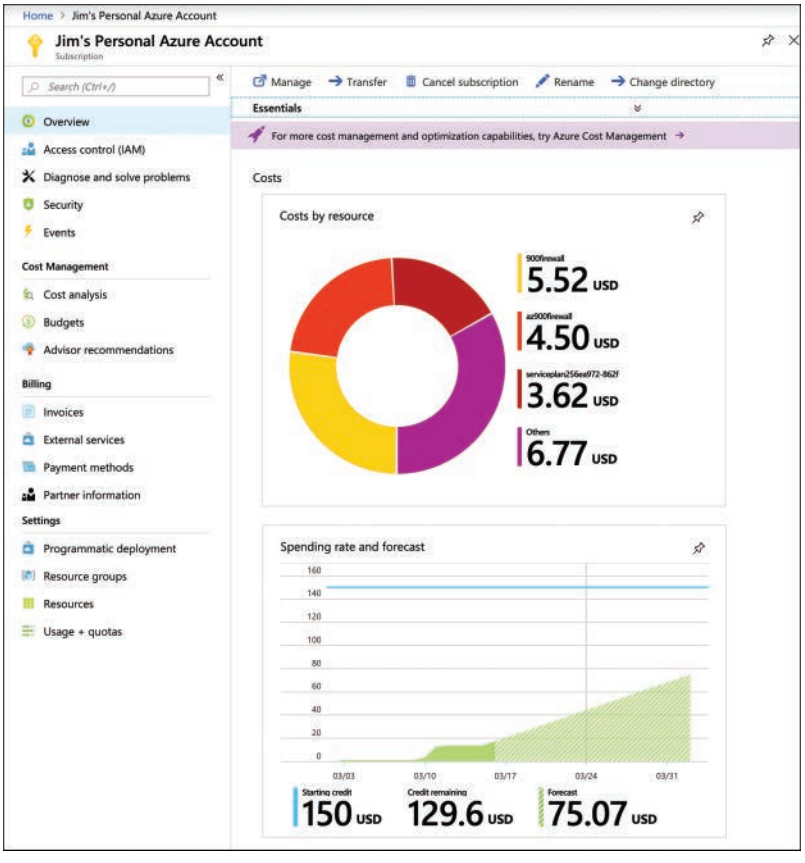


FIGURE 2-4 Azure subscription in the Azure portal

On the Overview blade, you can see a cost breakdown for each of the resources. You can also see the spending rate for the subscription, along with a forecasted cost by the end of the current month. If you click 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.

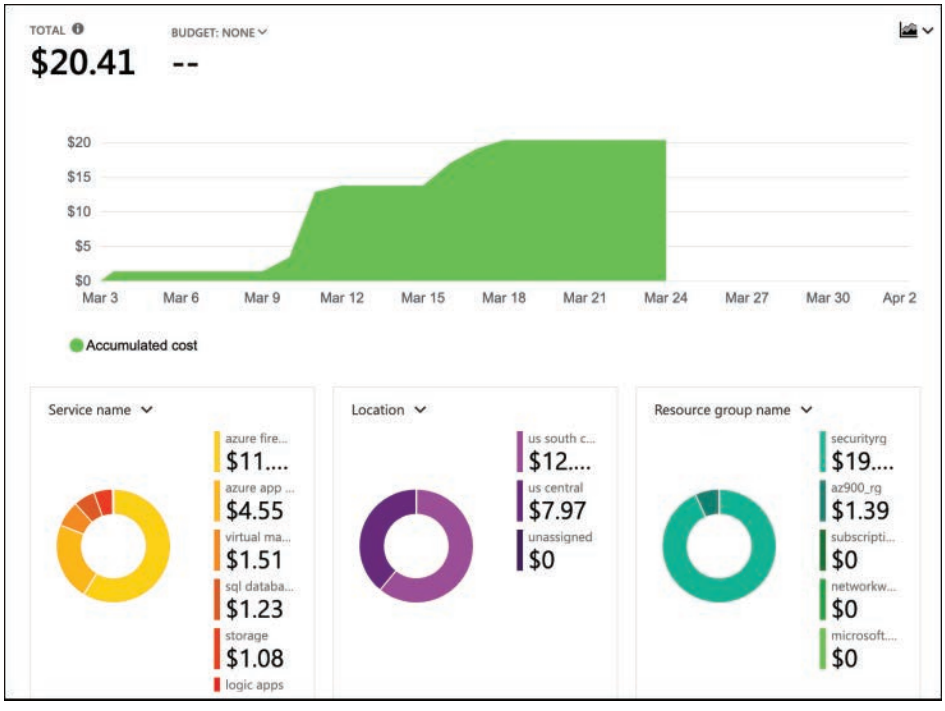


FIGURE 2-5 Azure subscription cost analysis

MORE INFO **CREATING BUDGETS**

You can manage your costs in Azure by creating budgets. You'll learn more about that in Chapter 6, "Describe Azure pricing, SLAs, and lifecycles."

Azure invoices are also available for the subscription from within the Azure portal. You can see all the past invoices by clicking Invoices in the menu for the subscription, as shown in Figure 2-6.

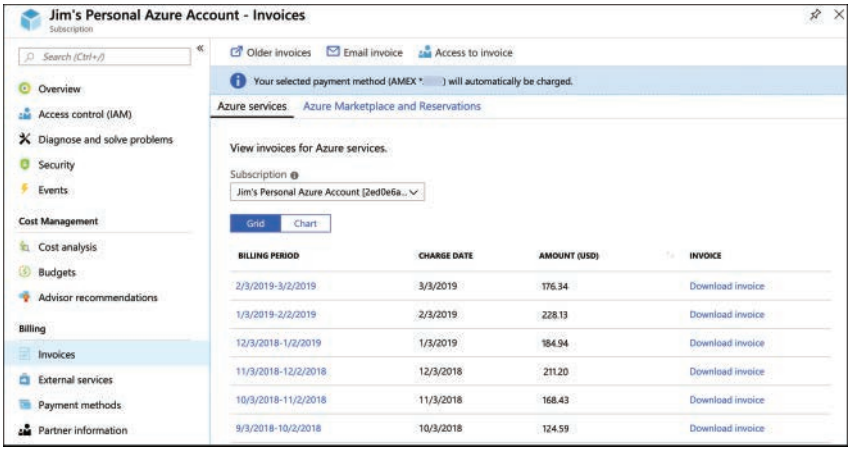


FIGURE 2-6 Azure invoices

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 2-7.

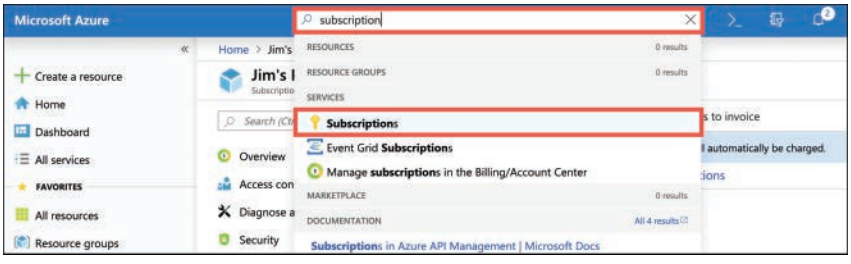


FIGURE 2-7 Azure subscriptions

To create a new subscription, click Add in the Subscriptions blade, as shown in Figure 2-8.

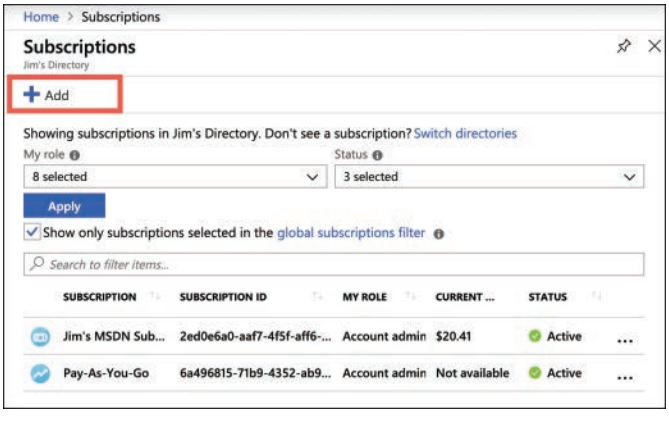


FIGURE 2-8 Creating a new subscription

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.
- **Pay-As-You-Go Dev/Test** A special subscription for subscribers to Visual Studio that can be used for development and testing. This subscription offers discounted rates on VMs, but you cannot use this for production applications.

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 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 have an understanding of 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 policies and access control. These concepts are introduced in Chapter 5, "Describe identity, governance, privacy, and compliance features."

In Figure 2-9, 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.

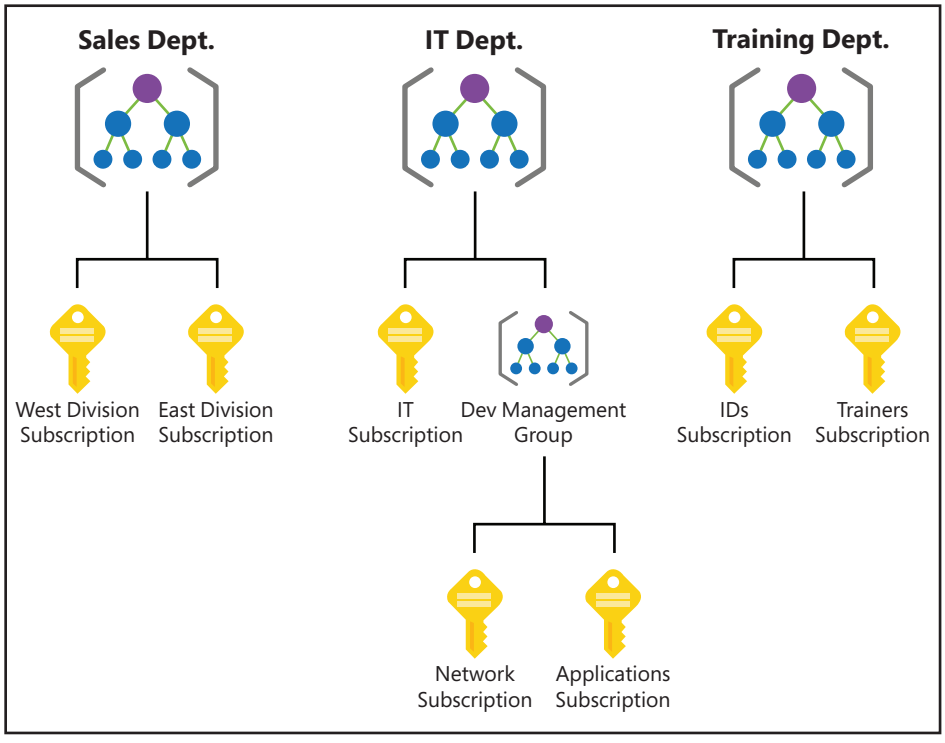


FIGURE 2-9 Management groups organizing subscriptions and other management groups

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.

Azure Resource Manager (ARM)

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.

In order to make it easier to deploy and manage Azure services, Microsoft developed Azure Resource Manager, or ARM. ARM is a service that runs in Azure, and it's responsible for all interaction with Azure services. When you create a new Azure service, ARM authenticates you to make sure 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

There are resource providers for every Azure service, but the names might not always make sense. For example, the *Microsoft.Compute* resource provider is responsible for creating virtual machine resources.

You don't have to know details on resource providers for the AZ-900 exam, but you should understand the general concept because you are expected to know about Azure Resource Manager.

In Chapter 3, you'll learn about using the Azure portal to create and manage Azure services. You'll also learn about how you can use command-line tools to do the same thing. Both the portal and the command-line tools work by using ARM, and they interact with ARM using the ARM application programming interface, or API. The ARM API is the same whether you're using the portal or command-line tools, and that 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 writing 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 2-10 illustrates this flow and features a small sampling of the many Azure services that are available.

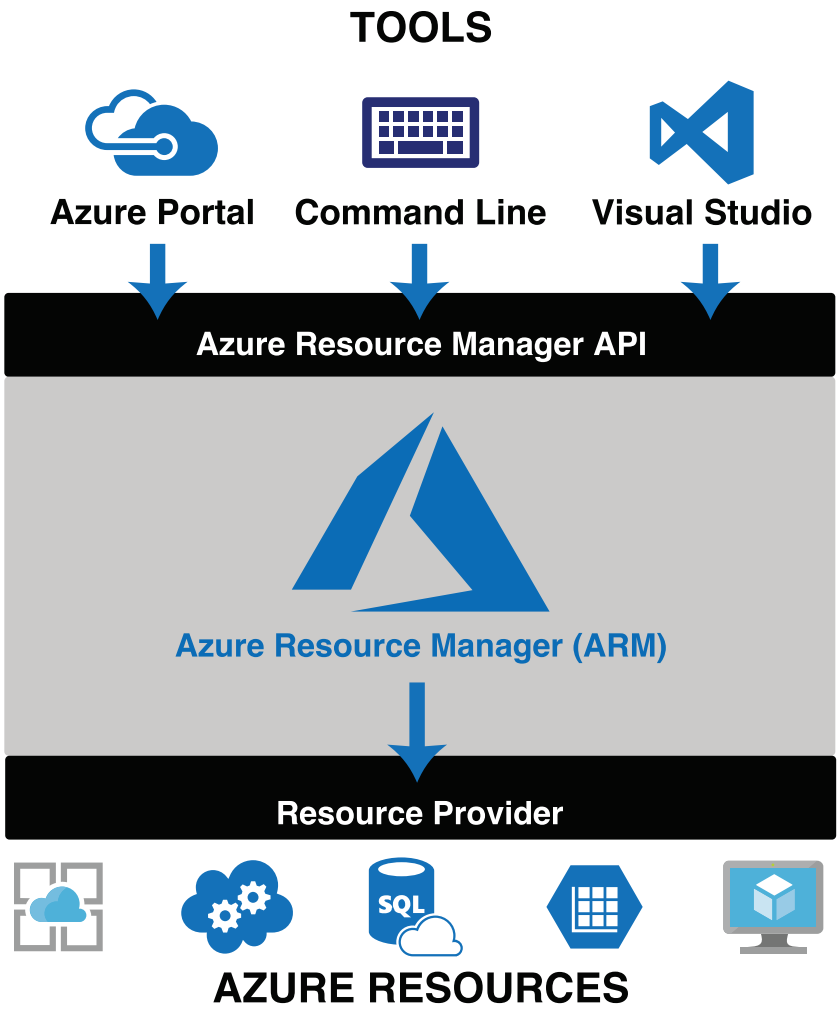


FIGURE 2-10 Azure Resource Manager

The request that is made to ARM isn't a complicated, code-based request. Instead, ARM uses *declarative syntax*. That means that, 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 have to tell it what you want. To do that, ARM uses files that are encoded in JavaScript Object Notation (or JSON) called *ARM templates*.

> ***NOTE*** **ARM TEMPLATES**
>
> **You don't need to know how to use ARM templates for the AZ-900 exam, but in order to grasp how ARM works, you really need to at least know a little about them.**

In the most basic sense, an ARM template contains a list of resources that you want to either 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 pretty 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.

As you can see, ARM has many benefits, and you should be aware of these for your exam:

- ARM allows you to easily deploy multiple Azure resources at once.
- ARM makes it possible to reproduce any deployment with consistent results at any point in the future.
- ARM allows you to create declarative templates for deployment instead of requiring you to write and maintain complex deployment scripts.
- ARM makes it possible to set up dependencies so that your resources are deployed in the right order every time.

Throughout this skill section, you've learned about some of the benefits of using Azure. Because Azure regions are spread out across the world in different geographies, you can be assured that your data and apps are hosted where you need them to be and that any regulations or data requirements are complied with. You learned that there are multiple datacenters in each region, and by deploying your applications in availability zones, you can avoid effects from a failure in a particular datacenter.

You also learned about using resource groups to organize your Azure resources and how to use Azure subscriptions. Finally, you learned about management groups and Azure Resource Manager, or ARM. In the next skill section, you'll learn details about some of the core workload products in Azure.

Skill 2.2: Describe core workload products available in Azure

As we went over the core Azure architectural components, you noticed some references to some of the products available in Azure. In this skill section, we'll talk about some of the core workload products available in Azure.

This section covers:

- Azure virtual machines
- Azure App Service
- Azure Container Instances (ACI)
- Azure Kubernetes Service (AKS)
- Windows Virtual Desktop
- Virtual networks
- ExpressRoute
- Container (Blob) Storage
- Disk Storage
- Azure Files
- Storage tiers
- Cosmos DB
- Azure SQL Database
- Azure Database for MySQL
- Azure Database for PostgreSQL
- The Azure Marketplace and its usage scenarios

Azure virtual machines

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 10, 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.

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 2-13.

1. Click Create A Resource.
2. Click Compute.
3. Click the See All link.
4. Click Ubuntu Server.

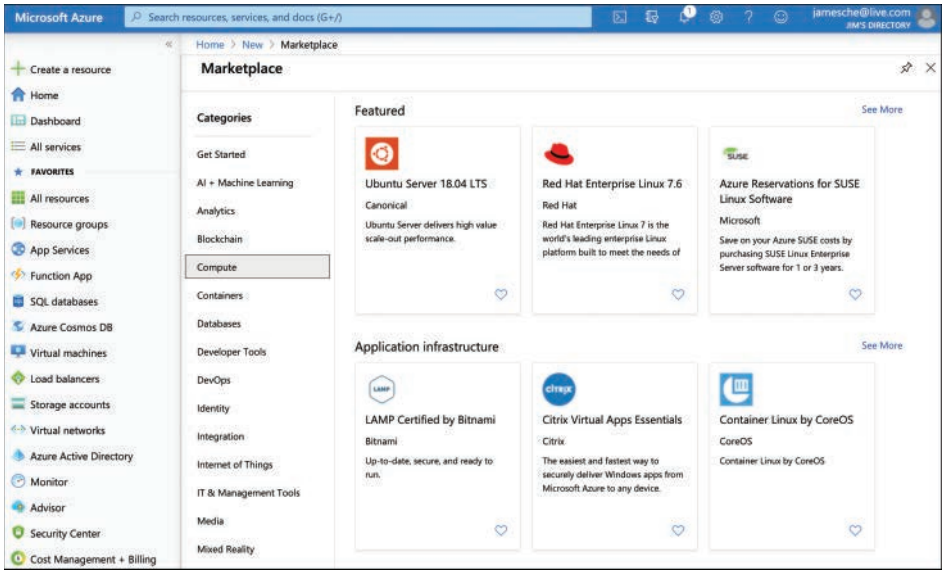


FIGURE 2-11 Creating a virtual machine

5. Click the Create button.
6. Next to Resource Group, click **Create New** to create a new resource group.
7. Enter **TestRG** as the resource group name and click OK.
8. Enter **TestVM** as your VM name.

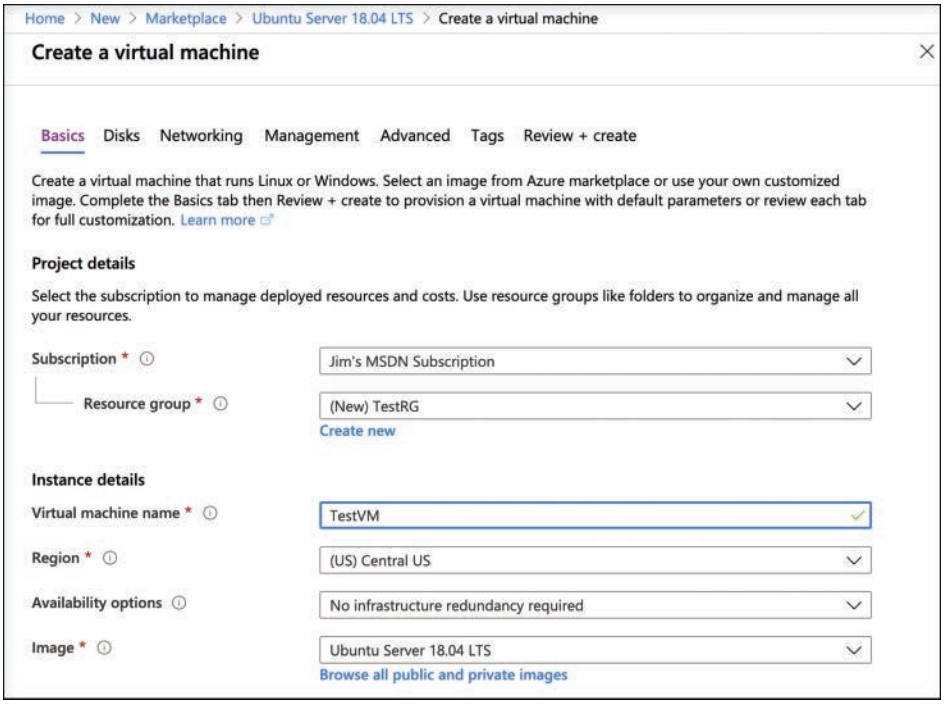


FIGURE 2-12 Virtual machine settings

9. Scroll down and select **Password** for the authentication type.
10. Enter a username for your administrator account.
11. Enter a password you'd like to use for your administrator account.
12. Confirm the password.
13. Leave all the other settings as they are and click **the Next button three times to move to the Management screen.**
14. In the Monitoring section, set Boot Diagnostics to Off.
15. Click **Review + Create** to create your VM.

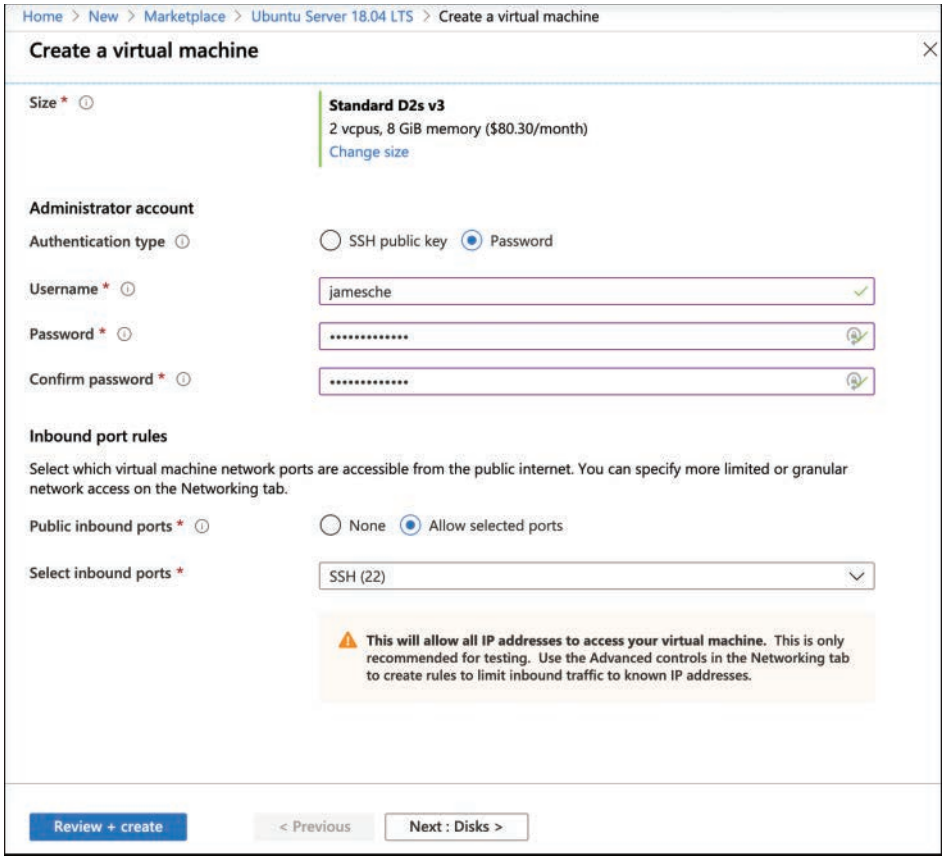


FIGURE 2-13 Virtual machine settings

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.

MORE INFO **HOW AZURE DEPLOYS YOUR VM**

When you click Create to create your VM, the Azure portal is actually using an ARM template to deploy your VM. That ARM template contains parameters that are replaced with the information you entered for your VM. Every VM that is created in Azure is created using an ARM template. This ensures that the deployments are consistent.

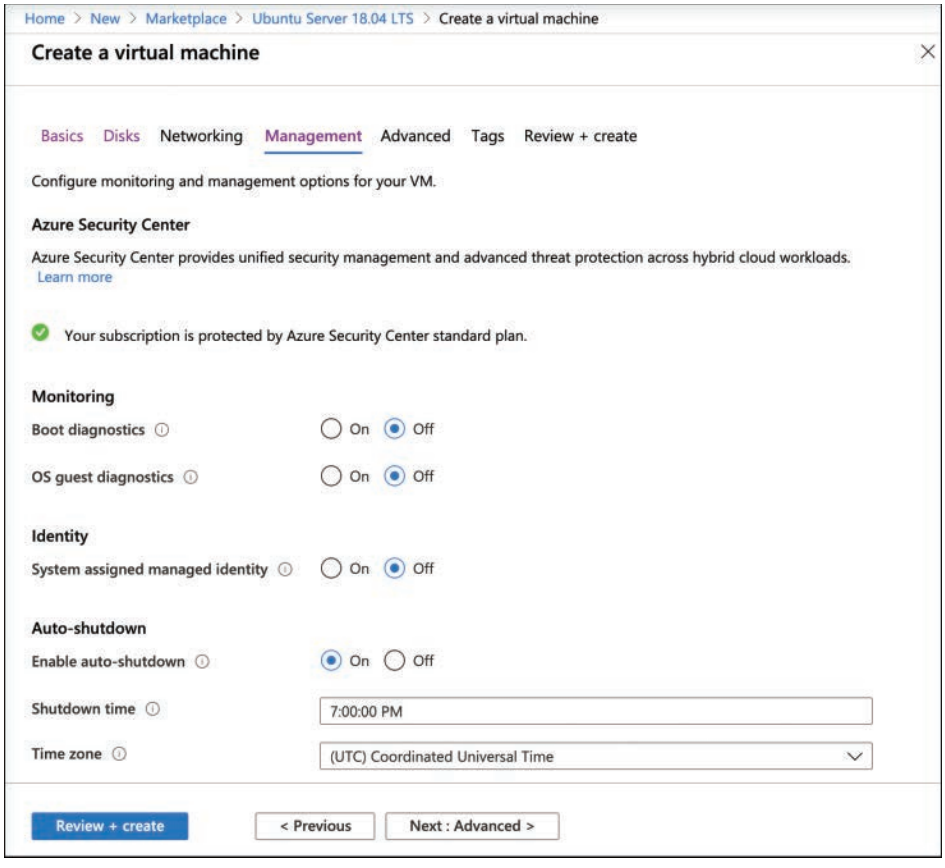


FIGURE 2-14 Virtual machine management settings

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 (which starts with the resource provider), 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.

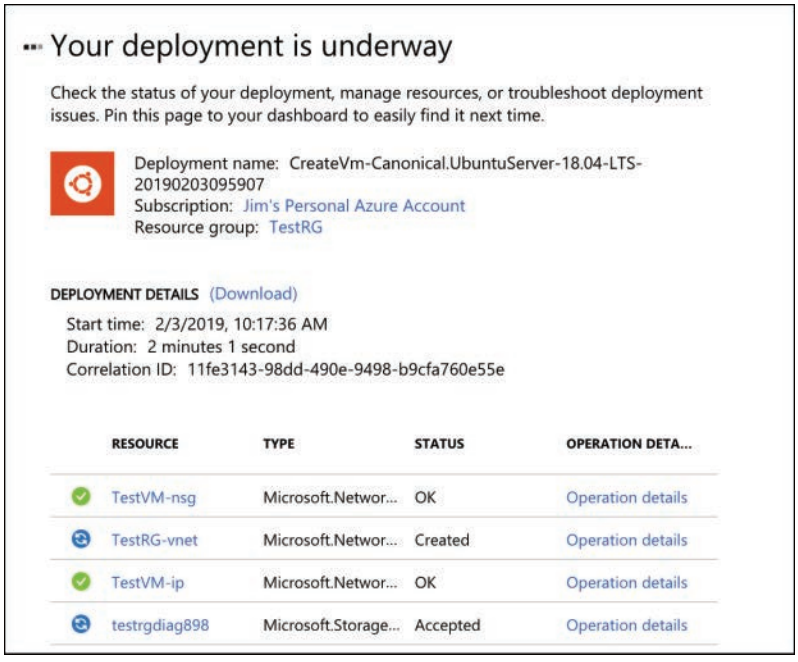


FIGURE 2-15 Virtual machine deployment

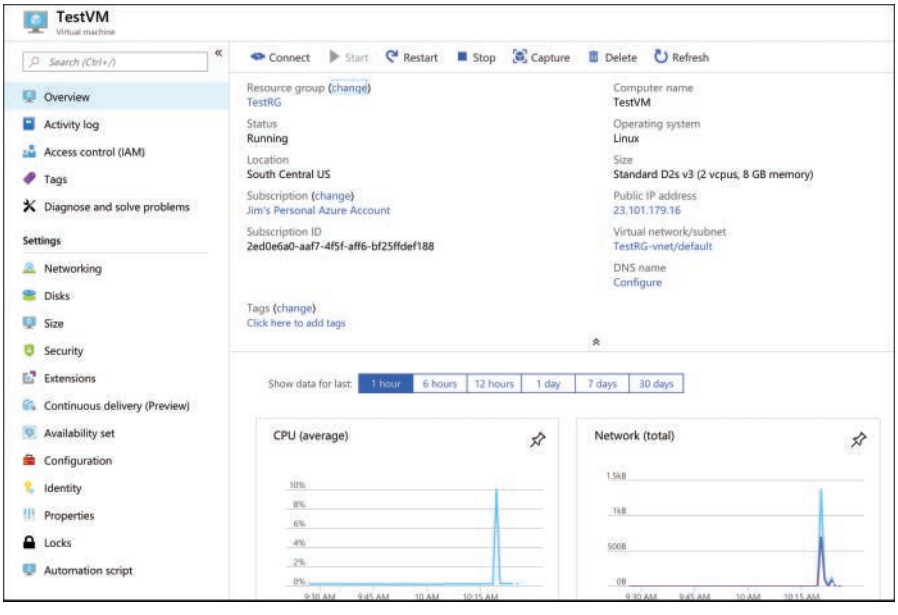


FIGURE 2-16 Viewing a virtual machine

Our 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 the VM is managed by you 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 2-15. 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 have configured a static 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. Once you finish this chapter, 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.

In order 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 the diagram that Microsoft uses to represent an availability set. In this diagram, the fault domains FD0, FD1, and FD2 encompass three physical racks of computers. UD0, UD1, and UD2 are update domains within the fault domains. You will see this same representation of an availability set within other Azure training as well, but it's a bit misleading because update domains are not tied to a particular fault domain.

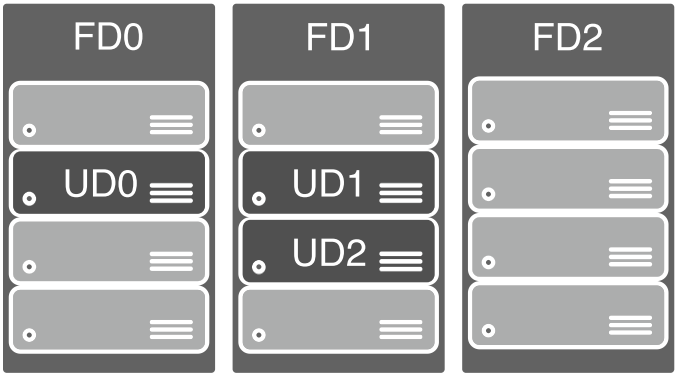


FIGURE 2-17 Microsoft documentation representation of an availability set

Figure 2-18 shows a better representation of an availability set, with five VMs in the availability set. 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.

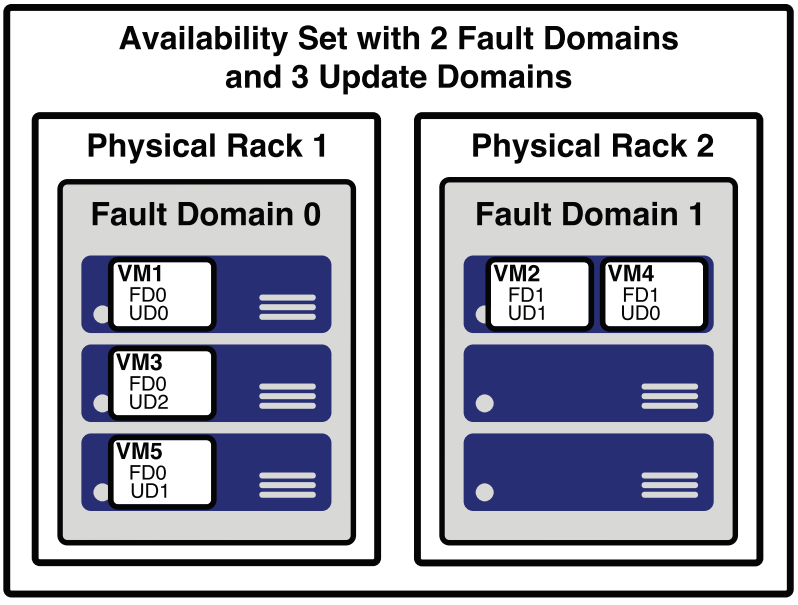


FIGURE 2-18 A better representation of an availability set

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 2-19, you can see the same configuration depicted in Figure 2-18.

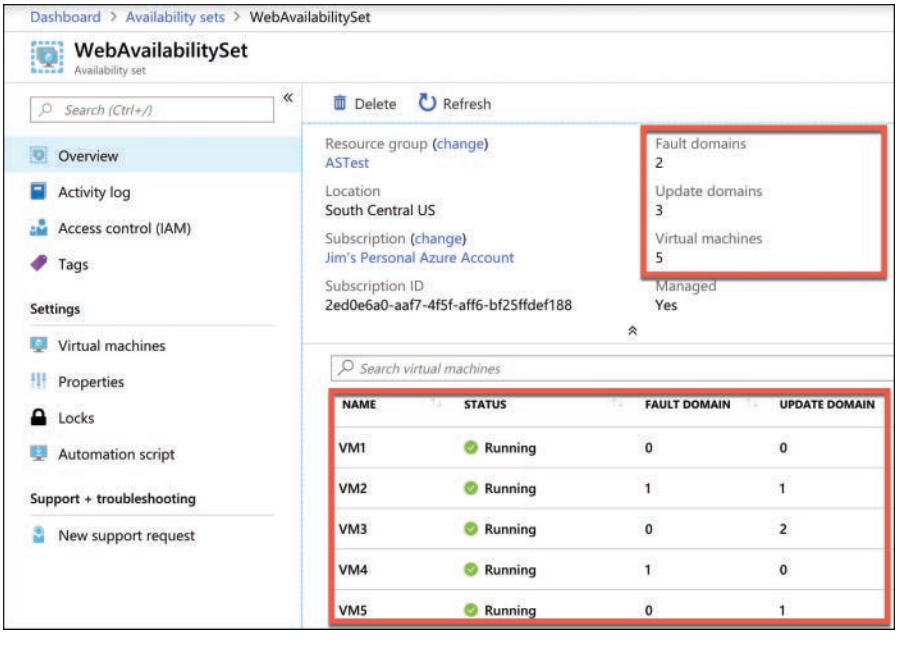


FIGURE 2-19 An availability set in the Azure portal showing fault domains and update domains

Notice in Figure 2-19 that the availability set is named *WebAvailabilitySet*. 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 of all, every machine in an availability set has to be explicitly created. While you can use an ARM template to deploy multiple virtual machines in one deployment, you still have to 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 to 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 of 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 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.

Azure App Service

As mentioned in Chapter 1, Azure App Service is a PaaS offering in Azure for hosting websites. 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.

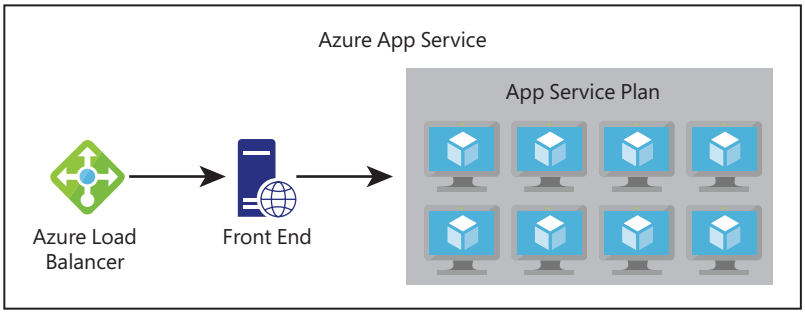


FIGURE 2-20 A high-level representation of Azure App Service

App Service plans

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 of 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.

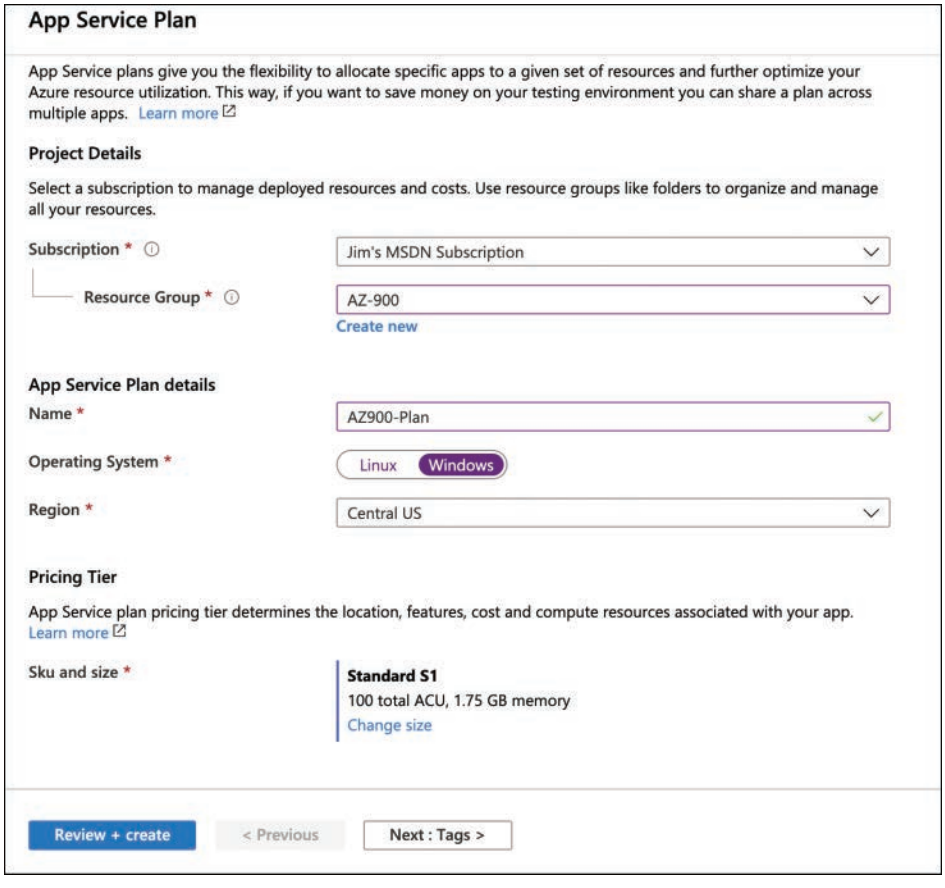


FIGURE 2-21 Creating an App Service plan in the Central US region

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.

Web apps

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) to run your app or a Docker container. If you choose to run a preconfigured runtime stack, you can choose between multiple versions that App Service provides.

MORE INFO **DOCKER CONTAINERS**

You'll learn about Docker contains in the next section when we cover Azure Container Instances.

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 3.0 apps on a Windows VM.

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.

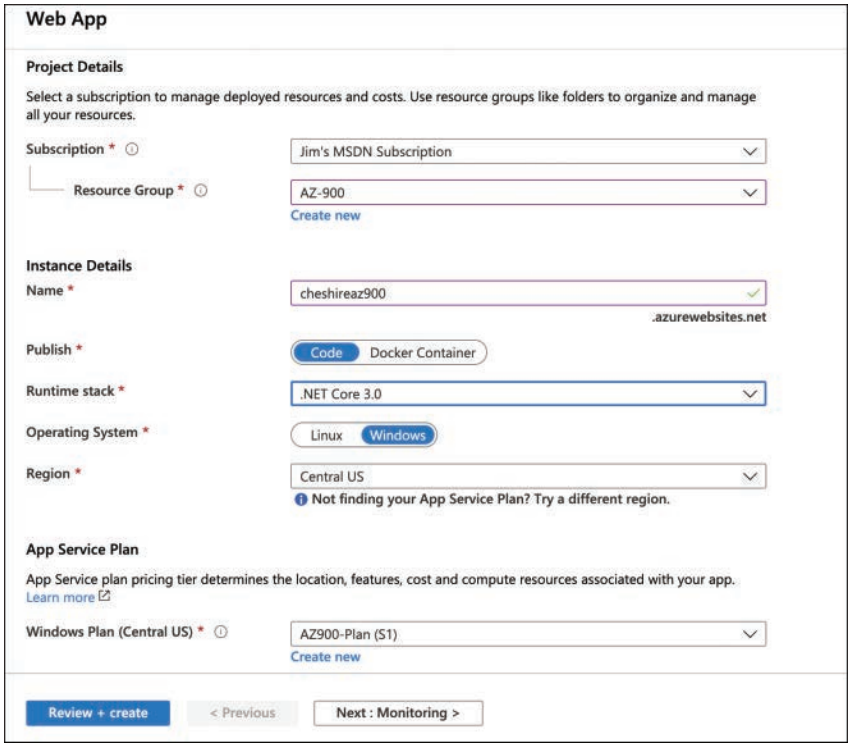


FIGURE 2-22 Creating a web app to run a .NET Core 3.0 website

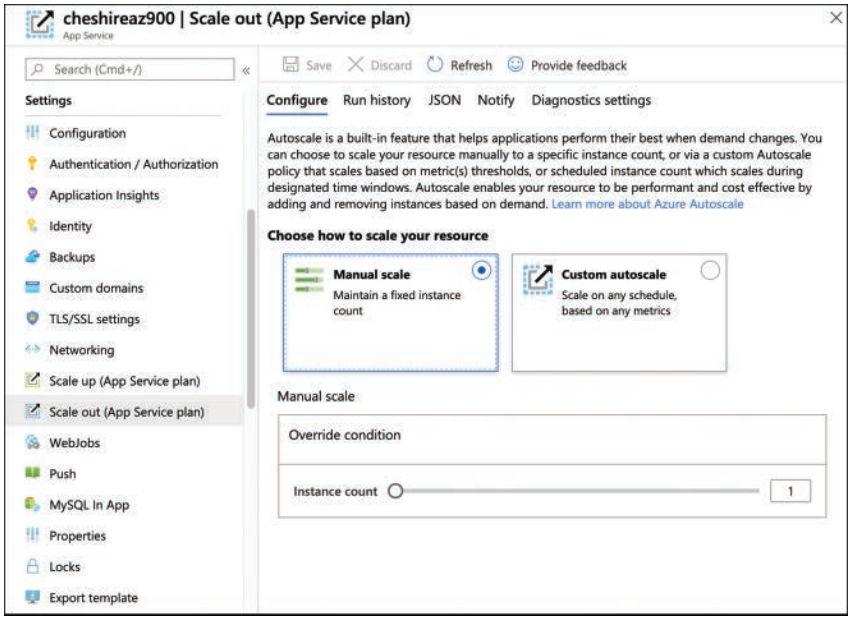


FIGURE 2-23 Settings for a web app make it easy to add features and scale your app

Azure Container Instances (ACI)

Azure Container Instances (ACI) is a PaaS service that offers the ability to run a containerized application easily. In order to understand how ACI works, it's necessary to have a basic understanding of containers.

Containers

It's becoming pretty 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. 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.

In order 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.

Running containers in ACI

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!

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Z