**Chapter 1**

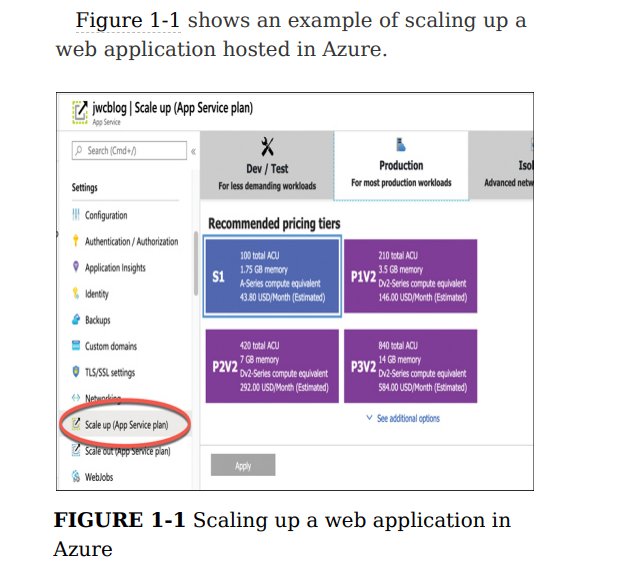
**DESCRIBE CLOUD CONCEPTS**  
This chapter covers the benefits of using the cloud, the different cloud services that are available, and cloud models that enable a variety of cloud configurations.

* High availability
* Scalability
* Elasticity
* Agility Fault tolerance and disaster recovery
* Economic benefits of the cloud

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**

**Network outage  
Application failure**: 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.  
**System outage  
Power outage**: 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

**Scaling and elasticity**  
Allow 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.  
  


**The concept of automatically scaling is referred to as elasticity.**

**This kind of speed and flexibility in the cloud is often called cloud agility.**If 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 have to do is tell Azure how many VMs you want and you’re up and running  
  
**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  
  
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.

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 particular needs

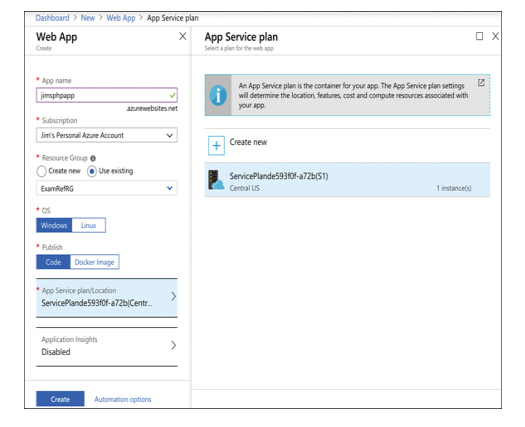
**SKILL 1.2: DESCRIBE THE DIFFERENCES BETWEEN INFRASTRUCTURE-AS-ASERVICE (IAAS), PLATFORMAS-A-SERVICE (PAAS), AND SOFTWARE-AS-A-SERVICE (SAAS)**

IAAS  
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

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.

PAAS  
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.  
  
A PaaS service also uses VMs provided by the cloud provider. However, a user typically has no visibility into those VMs. In most cases, they’re entirely managed by the cloud provider.

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. We also have the option to enable Application Insights, a service in Azure that provides deep insight into how an application is performing, making it easier to troubleshoot problems if they occur.



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.

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

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  
  
A PaaS service also benefits from all of the other enhancements offered by the cloud; you get fault tolerance, elasticity, easy and quick scaling, backup and disaster recovery features, and more

SAAS  
SaaS service is software provided by a cloud provider that’s installed on infrastructure completely controlled by the hosting provider.

* Microsoft 365
* Xbox Live
* One Drive

SKILL 1.3: DESCRIBE THE DIFFERENCES BETWEEN PUBLIC, PRIVATE, AND HYBRID CLOUD MODELS

**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 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 Platform are examples of public clouds.

**The private cloud**  
The private cloud model provides many of the attractive benefits of the cloud (things like easy scaling and elasticity) 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 thirdparty hosting provider.

**The hybrid cloud**  
As you might expect, hybrid clouds are a mixture of public and private clouds.

**Chapter 2**

**DESCRIBE CORE AZURE SERVICES**

This section covers:

* Azure regions
* Availability zones
* Resource groups
* Azure subscriptions
* Management groups
* Azure Resource Manager (ARM)

**Azure regions**   
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

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.

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.

*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.*

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  
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.

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

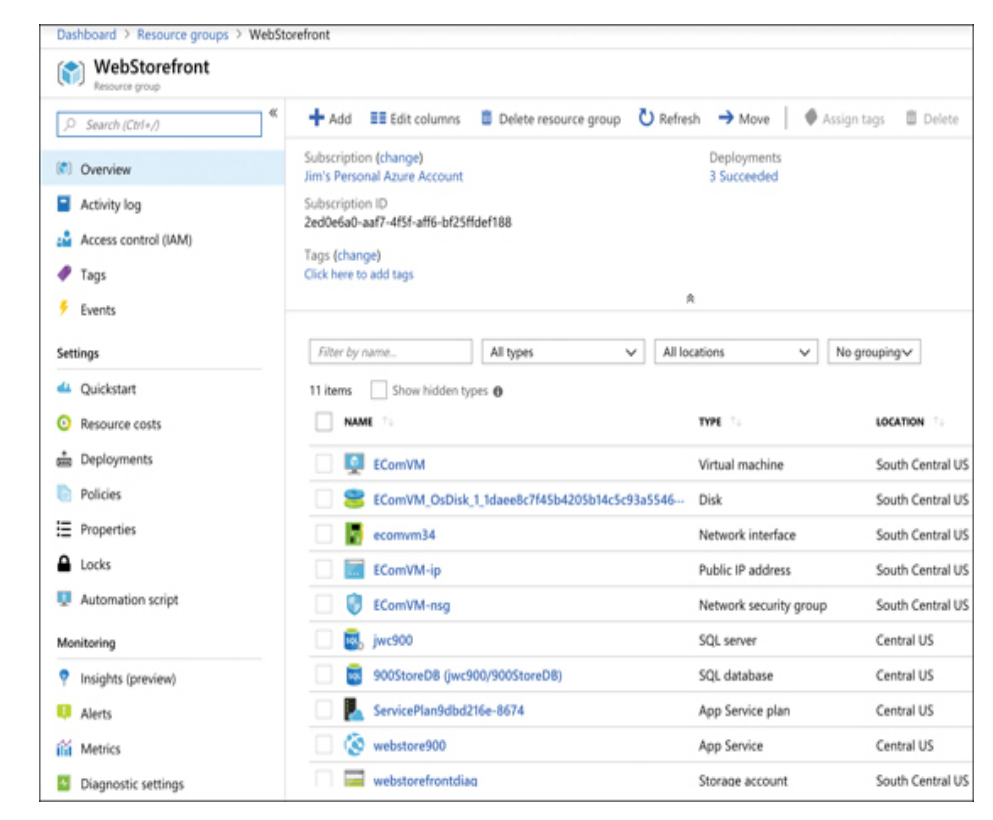
* Windows virtual machines
* Linux virtual machines
* Virtual Machine Scale Sets
* Azure Kubernetes Service
* Managed disks
* Zone-redundant storage
* Standard Load Balancer
* Standard IP address
* VPN Gateway
* ExpressRoute Gateway
* Application Gateway V2
* Azure Firewall
* Azure Data Explorer
* Azure 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

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 group.**

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.



*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*

Azure subscriptions*.*

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.

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

**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.

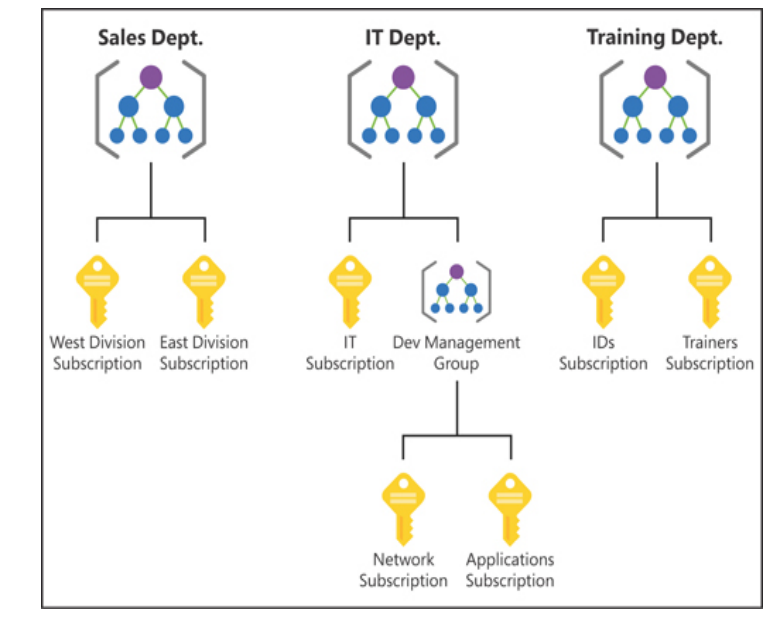
**Pay-As-You-Go** You pay only for those resources you use in Azure. There’s no up-front cost, and you can cancel the subscription at any time.

**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.

*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.*

**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.

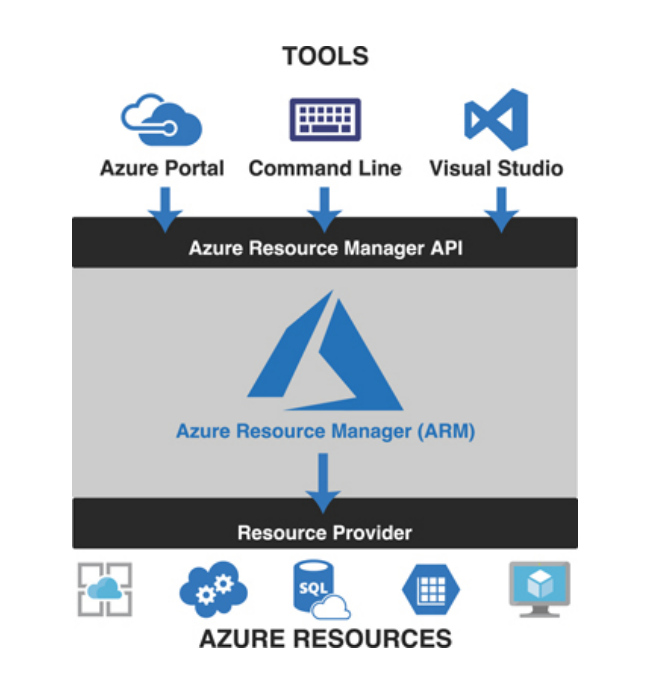


By organizing the subscriptions using management groups, you can have more precise control over who has access to which resources. You can also control the configuration of resources created within those subscriptions.

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)**

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.



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.

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.

* 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.

**SKILL 2.2: DESCRIBE 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**   
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.

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.

***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.*

**Planned maintenance** includes things like operating system updates, driver updates, and so on.

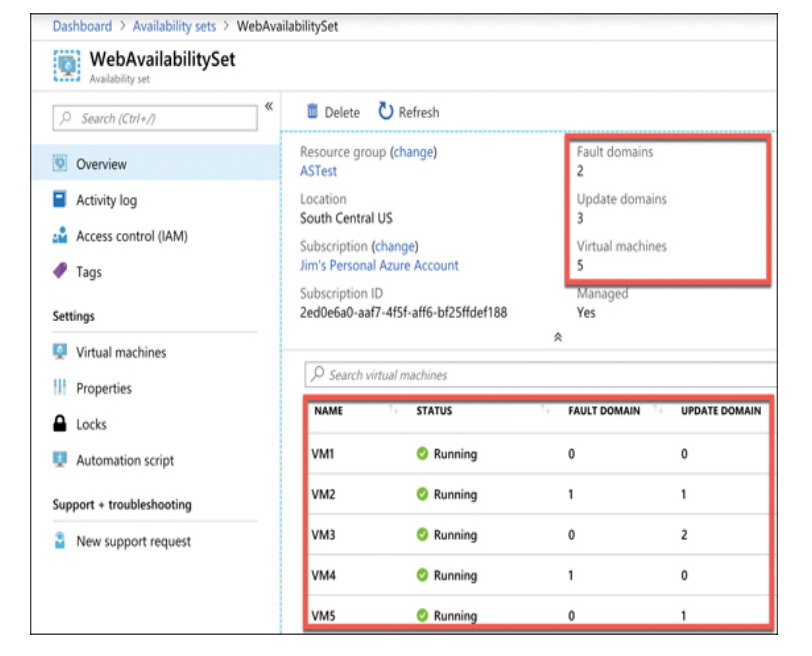
Azure has underlying systems that constantly monitor the health of computer components. If one of these underlying systems detects that a component within the host computer might fail soon, Azure will flag the computer for unplanned maintenance.   
In an **unplanned maintenance** event, Azure will attempt 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.**





*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.*

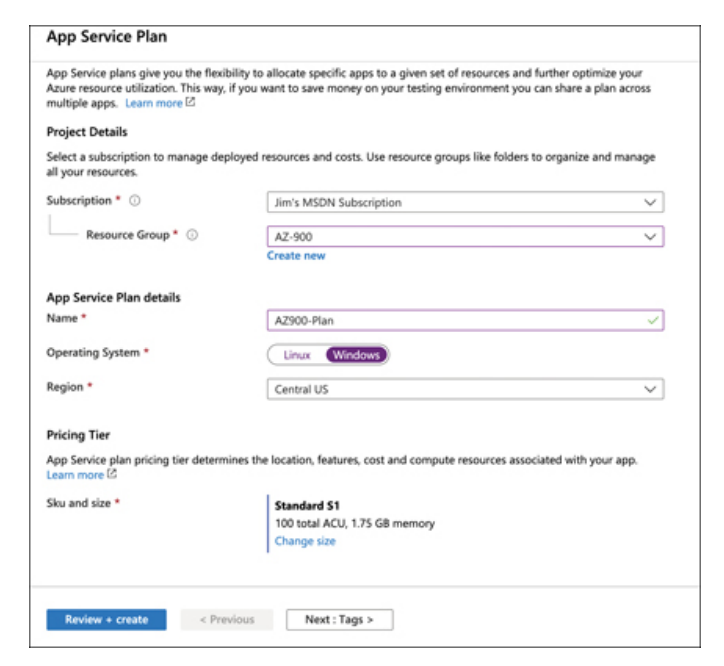
**Scale set:** When you create a scale set, you tell Azure what operating system you want to run and then you tell Azure how many VMs you want in your scale set. You have many other options such as creating a load balancer or gateway and so forth. Azure will create as many VMs as you specify (up to 1,000) in one easy step.

*VMs in a scale set are also compatible with availability zones, so you are protected from problems in an Azure datacenter.*

*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*

**Azure App Service**Azure App Service is a PaaS offering in Azure for hosting websites.  
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.

*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.*



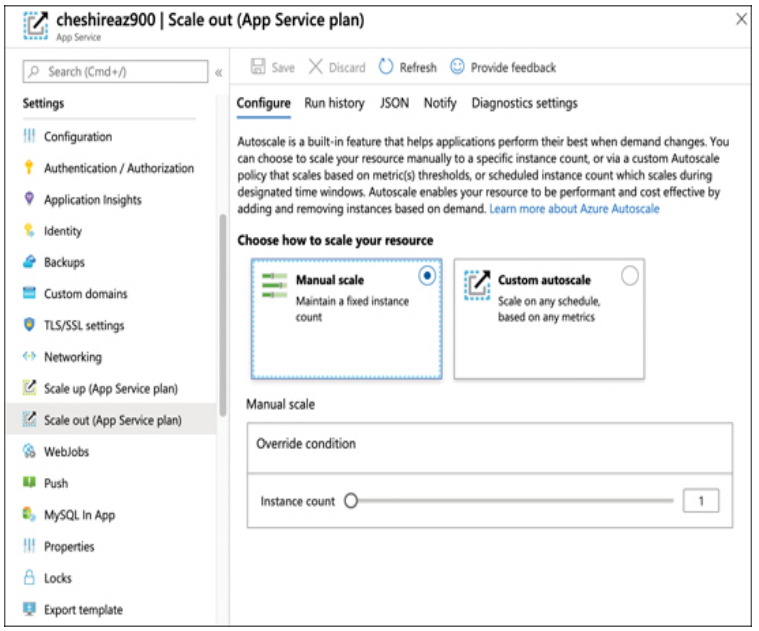
* **Free** A no-cost tier for testing only that runs on VMs shared with other App Service customers.
* **Shared** A low-cost tier for testing only with some additional features not offered in the Free tier. Runs on VMs shared with other App Service customers.
* **Basic, Standard, Premium, and PremiumV2**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

*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.*

*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.*

**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.

You can see many of the features available in App Service, including the ability to quickly and easily scale out when needed.  


**Azure Container Instances (ACI)**

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.

***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.  
  
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.*

**Azure Kubernetes Service (AKS)**

Kubernetes creates containers in a pod. A pod is a group of related containers, and containers within a pod can share resources.  
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.  
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 master node called the Kubernetes master. The entire environment of the master and all its nodes is called a **Kubernetes cluster.**

When you create a Kubernetes cluster in AKS, Azure creates the master 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 in Azure is free. You only pay for the Azure compute resources you use within your cluster*

**Windows Virtual Desktop**

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.

To use Windows Virtual Desktop (WVD), you first create a **WVD 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 WVD.

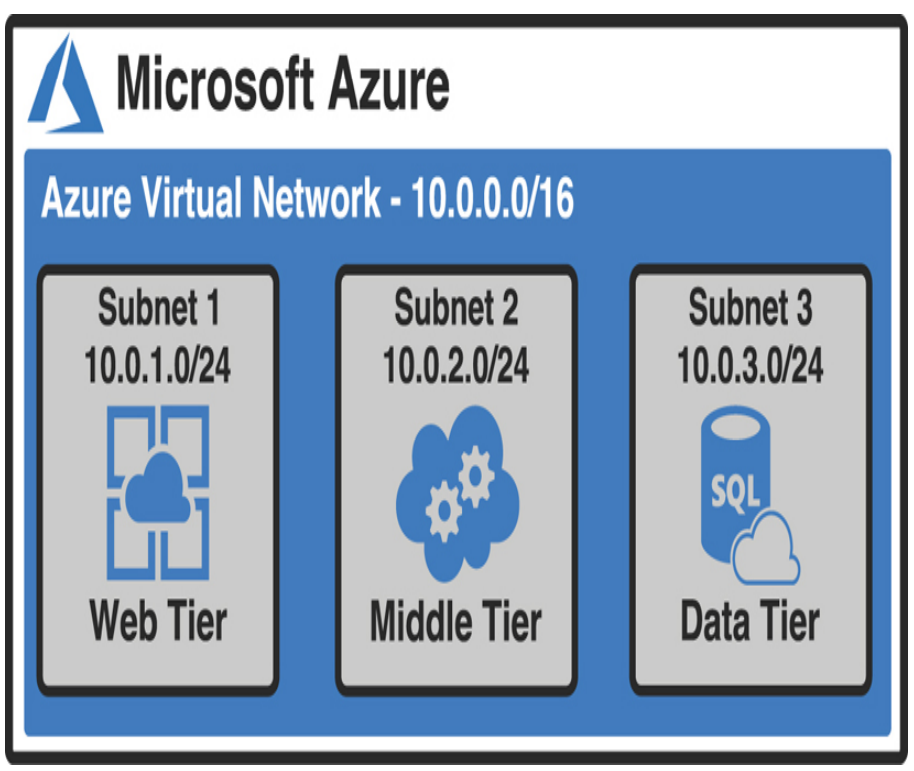
Once you’ve set up the tenant, you can add users from your **Azure Active Directory** so that they can access the OSes and apps in your tenant and assign permissions to them. Those users can then access WVD using the following methods.

* Using the WVD client app for Windows
* Using the WVD client app for MacOS
* Using the WVD client for iOS
* Using the WVD client for Android
* Using the web-based client from any web browser

**Virtual networks -** 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

When you create a VM in Azure, a VNet is created automatically.

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.



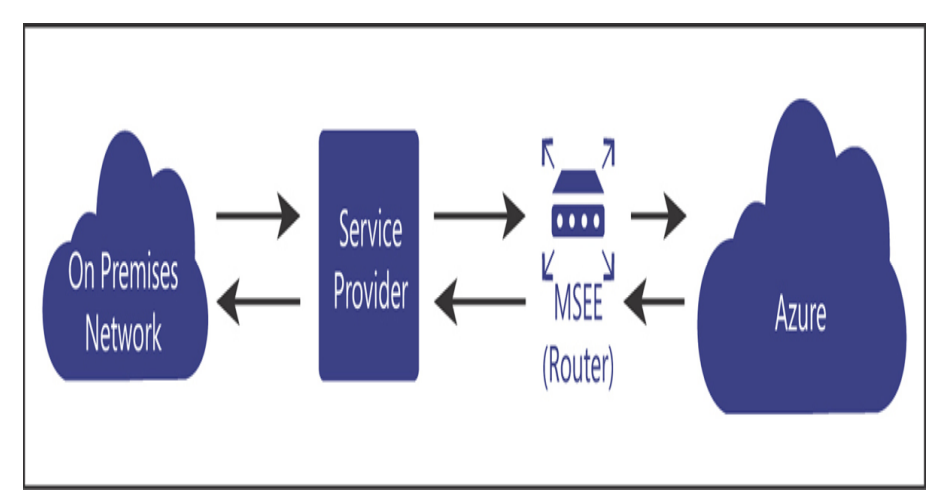
The IP addresses within the VNet at this point are all private IP addresses. They allow resources within the VNet to talk to each other, but you can’t use a private IP address on the Internet. You need a public IP address in order to give the Internet access to your web tier.

**ExpressRoute**

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. For this reason, 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.



*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 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.

**Container (blob) storage**

Azure Blob Storage is designed for storing unstructured data, which has no defined structure. That includes text files, images, videos, documents, and much more.

* **Block blobs** Used to store files used by an application
* **Append blobs** They 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** They are used to store virtual hard disk (.vhd) files that are used in Azure virtual machines. We’ll cover these in Azure Disk Storage later in this chapter.

**Azure Storage Explorer** a free tool available from Microsoft, to upload data.  
**Data Box** if you want to move a large amount of data

**Disk storage**

Disk storage in Azure refers to disks that are used in virtual machines. (Temporary storage when you create a VM)

If you need to store data for a longer period of time that will persist between VM deployments and maintenance events, you can create a disk using an image stored in Azure Storage.

**Unmanaged disks** They use an Azure Storage account in your Azure subscription, and you must manage that account. This is particularly troublesome because there are limitations in Azure Storage, and if you have heavy disk usage, you might end up experiencing downtime because of throttling.

**Managed disks** Microsoft handles the storage account, and all storage limitations are removed. All you need to worry about is your disk. You can leave the storage account in Microsoft’s hands.  
(you avoid a possible single point of failure in your VM)

*Microsoft recommends managed disks for all new VMs. They also recommend that all VMs currently using unmanaged disks be moved to managed disks.*

**Azure Files**

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.

*Azure Files is a completely managed file share that you can mount just like any SMB file share.*

*Because Azure Files shares use SMB, you’ll need to make sure that TCP port 445 is open on your network.*

**Azure File Sync** 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.

**Storage tiers**

Storage that are priced according to how often the data is accessed,

**Hot storage** tier is for data you need to access often. It has the highest cost of storage, but the cost for accessing the data is low.   
**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.***Archive storage** tier 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.

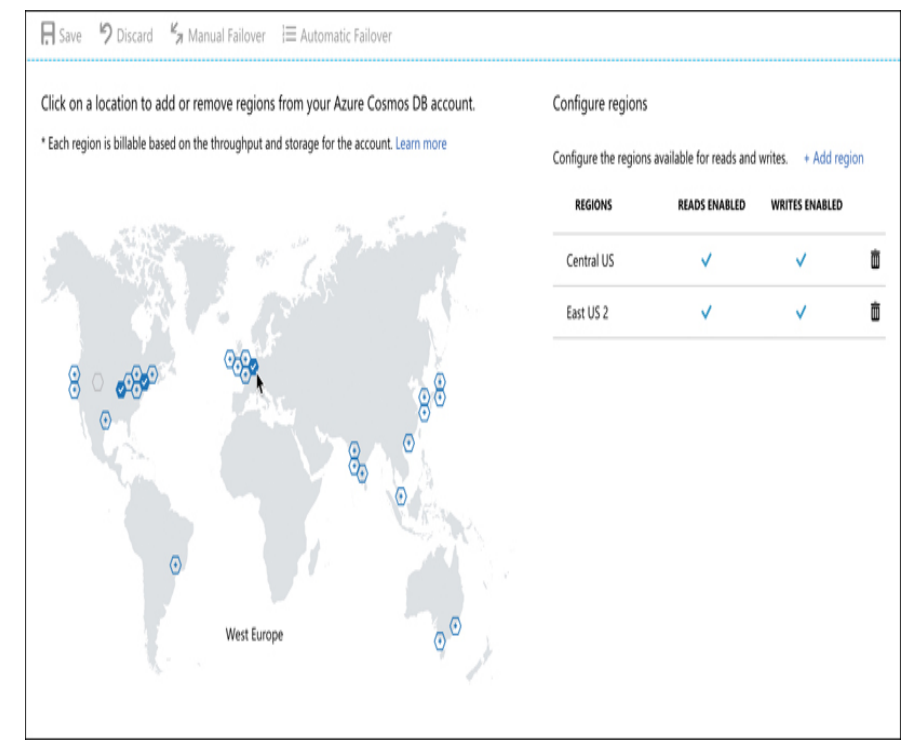
*In fact, while the Hot and Cool access tiers guarantee access to the first byte of data within milliseconds, the Archive tier only guarantees access to the first byte within 15 hours.*

**Cosmos DB**Supports all the NoSQL database types. Microsoft has built some custom code around Cosmos DB so that developers can use their existing skills with other database systems with a Cosmos DB database.

There are four types of NoSQL database systems: key-value, column, document, and graph.

* **Core (SQL)** Creates a document database that you can query using SQL syntax that you might be familiar with from using relational databases.
* **Azure Cosmos DB for MongoDB API** Used for migrating a MongoDB database to Cosmos DB. MongoDB databases are document databases.
* **Cassandra** Used for migrating a Cassandra database to Cosmos DB. Cassandra databases are column databases.
* **Azure Table Used** for migrating data stored in Azure Table Storage to Cosmos DB. This creates a key-value database.
* **Gremlin** Used for migrating Gremlin databases to Cosmos DB. Gremlin databases are graph databases.

*Another huge advantage to Cosmos DB is a feature known as turnkey global distribution. This feature takes advantage of the horizontal scalability of NoSQL databases and allows you to replicate your data globally with a few clicks.*



**Azure SQL Database**

Azure SQL Database is a PaaS offering for SQL Server database hosting. Microsoft manages the platform, so all you must worry about is your database and the data in it.

Azure offers three different deployment options for Azure SQL Database: single database, elastic pool, and managed instance.

**A single database** is simply a database running in a hosted SQL Server instance running in Azure. Microsoft manages the database server, so all you have to worry about is the database itself. Microsoft provides two different purchase models for single databases: Database Transaction Unit (DTU) and virtual core (vCore).

A **DTU** represents a collection of **CPU, memory, and data reads and writes**. There are three tiers in the DTU model: Basic, Standard, and Premium. Each tier offers a higher level of CPU, memory, and data transfer.

The **vCore** model uses a virtual CPU, and it makes it easy to configure the exact hardware configuration you need.  
*You can choose between a provisioned tier (where you choose the CPU, memory, and other resources that are always available) and a serverless tier where you choose a range of resource needs so you can control costs more effectively.*

An **elastic pool** consists of more than one database (and often many databases) all managed by the same SQL Database server. This solution is geared toward SaaS offerings where you might want to have multiple users (or maybe even each user) to be assigned their own database. You can easily move databases in and out of an elastic pool, making it ideal for SaaS.

*In some cases, being able to scale a single database to add additional power is sufficient. However, if your application has wide variations in usage and you find it hard to predict usage (such as with a SaaS service), the ability to add more databases to a pool is much more desirable.*

*While you can scale up and down easily with Azure SQL Database by moving to a higher tier or adding compute, memory, and storage resources, relational databases don’t scale horizontally. There are some options available for scaling out a read-only copy of your database, but in general, relational databases don’t offer the capability of scaling out to provide additional copies of your data in multiple regions.*

**DMS** Database Migration Service works by walking you through a wizard experience to tell Azure which database(s) and table(s) you want to migrate from your source database to Azure SQL Database.

**Azure Database for MySQL**

**Azure Database for PostgreSQL**

**The Azure Marketplace and its usage scenarios**

*All the templates in the Azure Marketplace are ARM templates that deploy one or more Azure services. Remember from our earlier discussion of Azure Resource Manager that all ARM deployments are deployed using ARM templates. The Marketplace is no different.*

**Chapter 3 DESCRIBE CORE SOLUTIONS AND MANAGEMENT TOOLS IN AZURE**

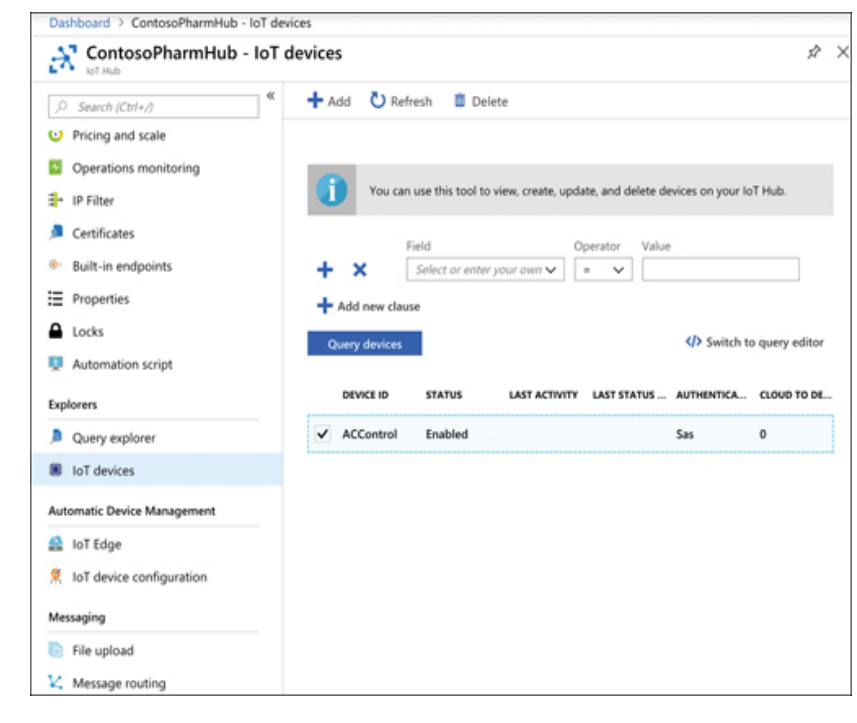
**SKILL 3.1: DESCRIBE CORE SOLUTIONS AVAILABLE IN**

**Azure IoT Hub**

IoT Hub can easily solve all these problems.

* They must update firmware on the IoT devices easily and in a staged way so they aren’t all updated at the same time.
* They must alter the settings on the devices, such as changing alert levels, but these settings are specific to the physical location of the devices in the building
* They must ensure that any connectivity to the devices is completely secure

Shows an IoT device added to the IoT Hub for ContosoPharm.



*From IoT Hub, you can send messages to devices (called cloud-to-device, or C2D messaging) or from your device to IoT Hub (called device-to-cloud, or D2C messaging). You can also intelligently route messages to Event Hub, Azure Storage, and Service Bus based on the content in the message*

In addition to messages, you can also use IoT Hub to send files to your devices. This allows you to easily update the firmware on your devices in a secure way. To update the firmware on an IoT device, you simply copy the firmware to the device. The device will detect the firmware and will reboot and flash the new firmware to the device.

**Device twin:** Every IoT device in IoT Hub has a logical equivalent that’s stored in IoT Hub in JSON format. This JSON representation of the device is called a device twin, and it provides important capabilities.

Each device twin can contain metadata that adds additional categorization for the device. This metadata is stored as tags in the JSON for the device twin, and it’s not known to the actual device. Only IoT Hub can see this metadata.

To help with users who want to add a large number of IoT devices to IoT Hub, Microsoft offers the IoT Hub Device Provisioning Service, or **DPS**

**The DPS** uses enrollment groups to add devices to your IoT Hub. The concept is that once the device wakes up (often, for the first time if it’s a new device), it needs to know that it should connect to your IoT Hub. In order to do that, the DPS needs to uniquely identify the device, and it does that with either a certificate or via a trusted platform module chip.

*As your devices send messages to IoT Hub, you can route those messages to Azure Storage, Event Hub, and various other endpoints.*

*There are two pricing tiers for IoT Hub: Basic and Standard. Each tier offers multiple editions that offer pricing based on the number of messages per day for each IoT Hub unit.*

**IoT Central**

IoT Hub is a great way to manage and provision devices, and it provides a robust means of dealing with messages.

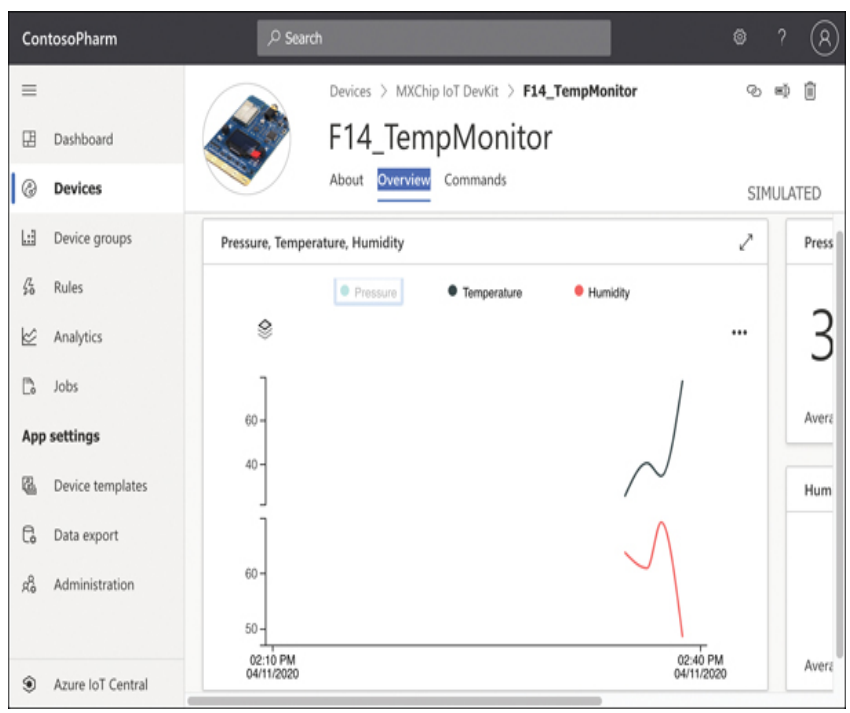
IoT Central is a SaaS offering for IoT devices. Unlike IoT Hub, you don’t have to create any Azure resources to use IoT Central. Instead, you browse to https://apps.azureiotcentral.com and create your app within the web browser interface

*The ability to create a simulated device is specific to IoT Central. IoT Hub doesn’t offer this capability.*

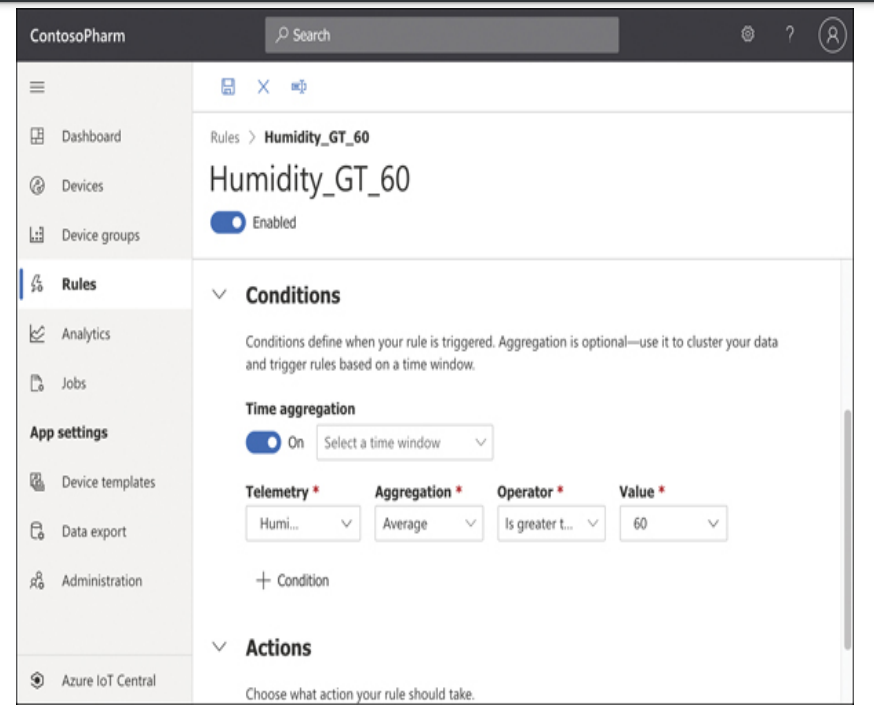
IoT Central gives you control over who can do what using roles. There are three built-in roles to which you can assign a user.

* **Administrator** Users in this role have full access to the application and can edit pages and add new users.
* **Builder** Users in this role can edit pages, but they can’t perform any administrative tasks, such as adding users, changing user roles, changing application settings, and so on.
* **Operator** Users in this role can use the application, but they can’t edit any pages and they can’t perform administrative tasks.

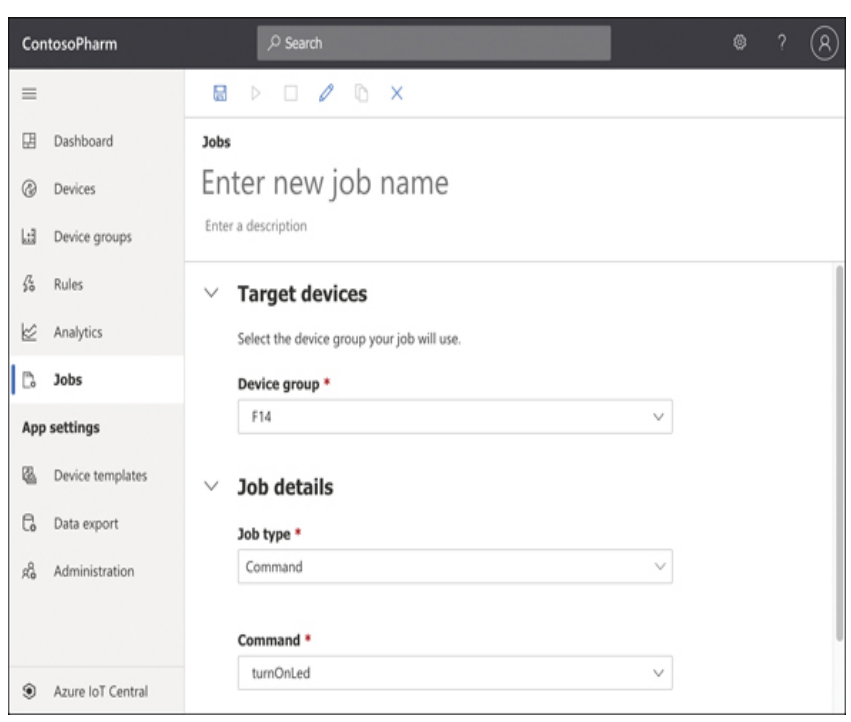
If you click a device, you can look at information coming from the device’s sensors. In Figure 3-11, you can see the Humidity and Temperature sensors on the F14\_TempMonitor device.



IoT Central also allows you to easily configure rules that will monitor your devices and perform an action you choose when your rule is activated. In Figure 3-12, we are configuring a rule that will activate when the Humidity is above 60.



A job can modify properties, change settings, or send commands to devices. In Figure 3-14, we’re creating a job that will turn on the LEDs for all devices in our device set.



**Azure Sphere**

To address these security issues, Microsoft developed Azure Sphere. Azure Sphere is based on Microsoft’s decades of experience and indepth research that Microsoft conducted on securing devices

Azure Sphere is actually an entire ecosystem, and it starts with the chip, or the microprocessor unit (MCU). Microsoft has developed an Azure Sphere MCU that contains security components embedded in the chip.

Third parties can use these MCUs to run code that is specific to their needs, and that code runs on the Azure Sphere operating system, which is a customized version of Linux developed for Azure Sphere.

This ecosystem provides a secure environment for running embedded code, but it also enforces secure communication between devices.

*Pricing for Azure Sphere includes operating system and Azure Sphere Security Service updates through July of 2031.*

**Azure Synapse Analytics**

The problem of actually doing anything with the vast data we collect is common across all businesses, and this is what we mean by big data.

Analyzing big data requires a powerful system for storing data, the ability to query the data in multiple ways, enormous power to execute large queries, assurance that the data is secure, and much more. That’s exactly what **Azure Synapse analytics provides**.

*Azure Synapse is the next evolution of another Azure service called SQL Data Warehouse. While it’s true to say that Azure Synapse is the replacement for SQL Data Warehouse, it’s also important to note that Azure Synapse adds much more functionality. SQL Data Warehouse was focused primarily on storage of big data (called warehousing), but Azure Synapse provides that functionality in addition to powerful analytics features.*

Azure Synapse runs in an Azure Synapse cluster. A cluster is a combination of four different components:

* Synapse SQL
* Apache Spark integration
* Data integration of Spark and Azure Data Lake Storage
* A web-based user interface called Azure Synapse Studio

**Synapse SQL** is the data warehousing portion of Azure Synapse. Using Synapse SQL, you can run powerful queries against your big data.

Each compute node also runs a component called the **Data Movement Service (DMS)** that moves data efficiently between compute nodes.

**Azure Data Lake Storage** is designed for storing large amounts of data that you’d like to analyze, but Data Lake Storage is designed for a wide array of data instead of relational data. In a data lake, data is stored in containers. Each container typically contains related data.

*The terms data lake and data warehouse aren’t specific to Azure. They are generic terms. A data lake refers to a repository of unordered data, and a data warehouse refers to a repository of ordered data.*

**HDInsight**

HDInsight makes it possible to easily create and manage clusters of computers on a common framework designed to perform distributed processing of big data. Essentially, HDInsight is Microsoft’s managed service that provides a cloud-based implementation of a popular data analytics platform called Hadoop (The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.).

HDInsight supported cluster types

**Apache Hadoop:** A framework that uses HDFS, YARN resource management, and a simple MapReduce programming model to process and analyze batch data in parallel.

**Apache Spark:** A parallel processing framework that supports in-memory processing to boost the performance of big-data analysis applications. Spark works for SQL, streaming data, and machine learning. See What is Apache Spark in HDInsight?

**Apache HBase:** A NoSQL database built on Hadoop that provides random access and strong consistency for large amounts of unstructured and semi-structured data–potentially billions of rows times millions of columns. See What is HBase on HDInsight?

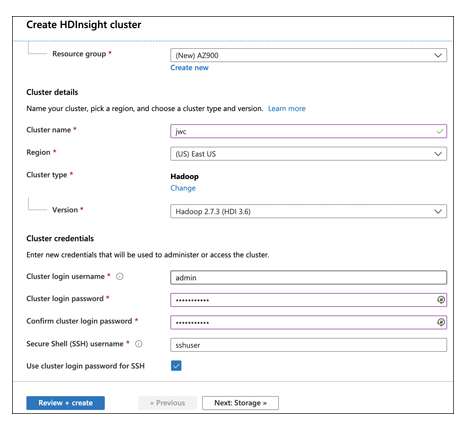
**Microsoft R Server:** A server for hosting and managing parallel, distributed R processes. It provides data scientists, statisticians, and R programmers with on-demand access to scalable, distributed methods of analytics on HDInsight. See Overview of R Server on HDInsight.

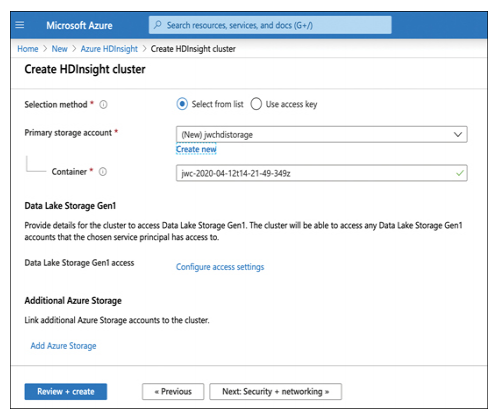
**Apache Storm:** A distributed, real-time computation system for processing large streams of data fast. Storm is offered as a managed cluster in HDInsight. See Analyze real-time sensor data using Storm and Hadoop.

**Apache Interactive Query preview (AKA: Live Long and Process):** In-memory caching for interactive and faster Hive queries. See Use Interactive Query in HDInsight.

**Apache Kafka:** An open-source platform that’s used for building streaming data pipelines and applications. Kafka also provides message-queue functionality that allows you to publish and subscribe to data streams. See Introduction to Apache Kafka on HDInsight.

With HDInsight, Microsoft does all the heavy lifting on its own infrastructure. You benefit from a secure environment—one that is easily scalable to handle huge data processing tasks.



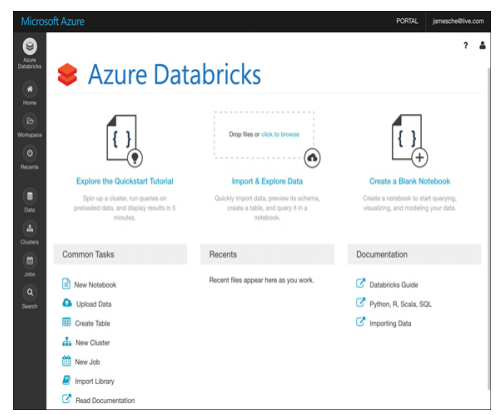


*HD Insight clusters are billed on a per hour basis, and you pay more per hour based on how powerful the machines are* ***in your cluster.***

**Azure Databricks**

Data that gets stored in a data warehouse or data lake is typically raw data that is often unstructured and difficult to consume. Also, you might need data that comes from multiple sources, some of which might even be outside of Azure. Azure Databricks is an ideal solution for accumulating data and for forming the data (called data modeling) so that it’s optimal for machine learning models.

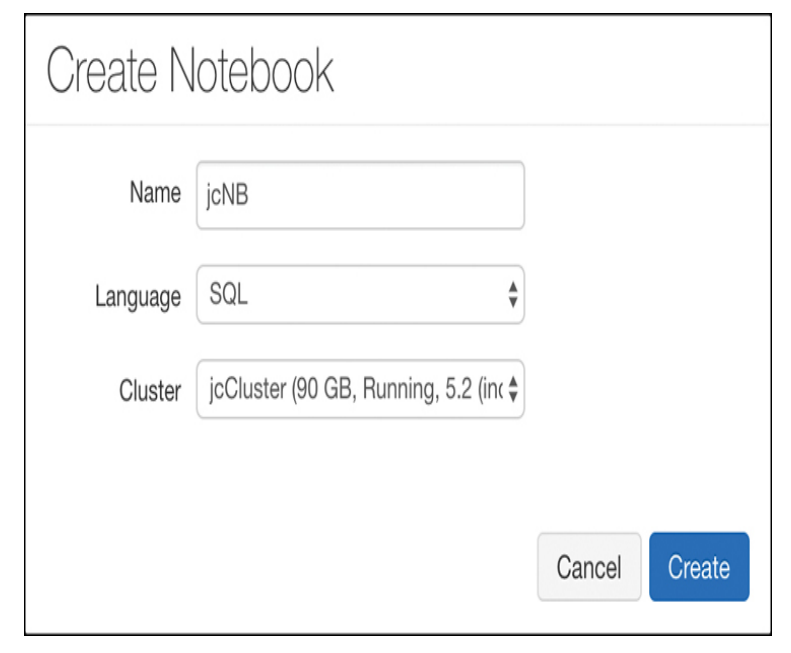
*Databricks is actually the name of a company that originally developed Apache Spark. It now operates a data analytics platform called Databricks. You might be tempted to think of Azure Databricks as the Databricks platform running as a service in Azure, but it’s much more than that. In fact, Microsoft natively built the Databricks Runtime to run in Azure, and Azure Databricks provides many more unique features outside the Databricks platform developed by Databricks*



Databricks does all its work using clusters, which are the compute resources.

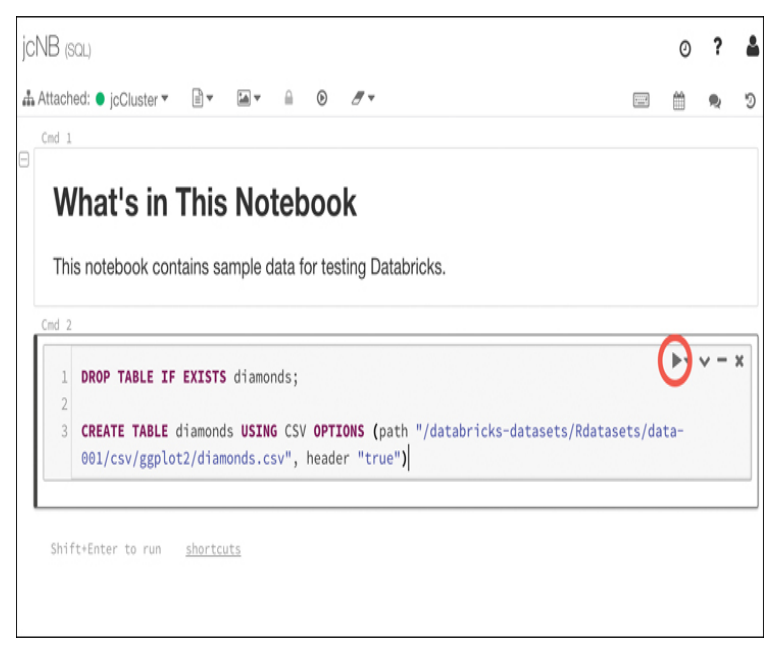


Next, we’ll create a notebook. Notebooks are a powerful way to present and interact with related data. Each notebook contains data as well as visualizations and documentation to help us understand the data. Once your data is in your notebook, you can run commands against machine learning frameworks in order to build your machine learning model right inside your notebook.



You can also choose to specify Python, Scala, or R as the language.

*Documentation in notebooks is entered using Markdown, a language that’s well-suited to writing documentation*



*When you run commands in a cell, Databricks creates a job that runs on the compute resources you allocated to your cluster. Databricks uses a serverless model of computing. That means that when you’re not running any jobs, you don’t have any VMs or compute resources assigned to you.*

*You can use data in Databricks for training machine learning algorithms. The Databricks Runtime ML includes several popular libraries for machine learning, including****: Keras, PyTorch, TensorFlow, and XGBoost.***

***The important point to remember is that Databricks works with third-party machine learning frameworks to allow you to build machine learning models.***

*Databricks is based on Apache Spark.*

Once you’ve built your machine learning model in Databricks, you can export it for use in an external machine learning system. This process is referred to as productionalizing the machinelearning pipeline, and Databricks allows you to productionalize using two different methods: MLeap and Databricks ML Model Export.

**MLeap** is a system that can execute a machine learning model and make predictions based on that model.

**Databricks ML Model Export** is designed to export your machine learning models and pipeline so that they can be used in other machine learning platforms.

**Azure Machine Learning**

All the AI that we’ve developed so far is weak AI. On the other end of the AI spectrum is Artificial General Intelligence, or strong AI. This is the type of AI you see depicted in movies and science fiction books, and we don’t currently have this kind of capability

AI works by creating a digital neural network. Each part of that neural network can communicate and share information with every other part of the network. Just like our brains, a computer neural network takes input, processes it, and provides output. AI can use many methods for processing the input, and each method is a subset of AI. The two most common are natural language understanding and **machine learning.**

**Machine learning (ML)** is similar in that it uses a neural network to accomplish a task, but the task is different from understanding speech. In fact, machine learning can be used in many applications. One of the common uses of machine learning is image recognition.

ML uses a learning algorithm that is the basis for the AI. Once the algorithm is developed, you feed test data to it and examine the result. Based upon that result, you might determine that you need to tweak the algorithm

Azure Machine Learning is designed to make ML approachable to just about anyone. It offers SDKs for both Python and R, and it also offers a drag-and-drop environment and an automated mode for more easily creating and training ML models in a visual way

Azure Machine Learning is available in two editions;   
  
**The Basic edition** offers only access to ML SDKs and notebooks.  
**The Enterprise edition** offers the features of the Basic edition, but it adds many additional features, including the visual designers.

*Azure Machine Learning is priced based upon your usage. You’re billed for a VM where your  
Azure Machine Learning assets run. You are also charged a machine learning surcharge and a small amount per hour for usage.*

**Cognitive Services**

You can think of Cognitive Services as SaaS ML models that you can use directly in your ML solutions without the expense of developing your own.

Computer Vision can do things like

* Recognize objects or recognize a scene
* Tecognize inappropriate content so you
* Moderate images.

**Video Indexer API**

* can analyze video content and extract information from that content.
* You can easily add closed captioning in multiple languages recognize people and objects, and search for videos that contain specific words, people, or even emotions

**Speech Translation**

* language translation in real-time to Speaker Recognition
* analyze speech and identify the speaker.

Pricing for Azure Cognitive Services is

**Transactional** that means that you pay a small amount for transactions that you process through the service.

**Azure Bot Service**

Is a PaaS offering that runs on Azure App Service. That means that it inherits all the features of App Service such as easy scaling and simple configuration.

This includes editing the source code, building the source code, viewing analytics, and so forth. All these tasks are part of the Bot Framework that Microsoft developed to streamline the creation of bots.

*Azure Bot Service can be connected to many popular services such as Slack, Facebook Messenger, Microsoft Teams, and more. Each of these is considered a channel within the Bot Service, and the channels that Microsoft provides are called standard channels. However, you can also use what Microsoft calls Direct Line to connect your bot to your own application or website. Direct Line is considered a premium channel, and it will cost a small amount for the messages it uses.*

Because Bot Service runs on Azure App Service, you are also charged for an App Service plan when you create a Bot Service.

**Serverless computing**

Sometimes, you just need to “borrow” a computer in order to run a computation or perform a quick task. In those situations, a serverless environment is ideal.   
In a serverless situation, you pay only when your code is running on a VM.   
When your code’s not running, you don’t pay anything.

All cloud providers need surplus capacity so they can meet the needs of customers, but when VMs are sitting there waiting for a customer who might want to use them, it means lost revenue for the cloud provider.   
To solve that problem, cloud providers created consumption-based plans that allow you to run your code on these surplus VMs, and you pay only for your use while your code is running.

*Cognitive Services is an example of a serverless service, but Azure has many other serverless services*

**Azure Functions**

Azure Functions is the compute component of Azure’s serverless offerings. That means that you can use Functions to write code without having to worry about deploying that code or creating VMs to run your code.

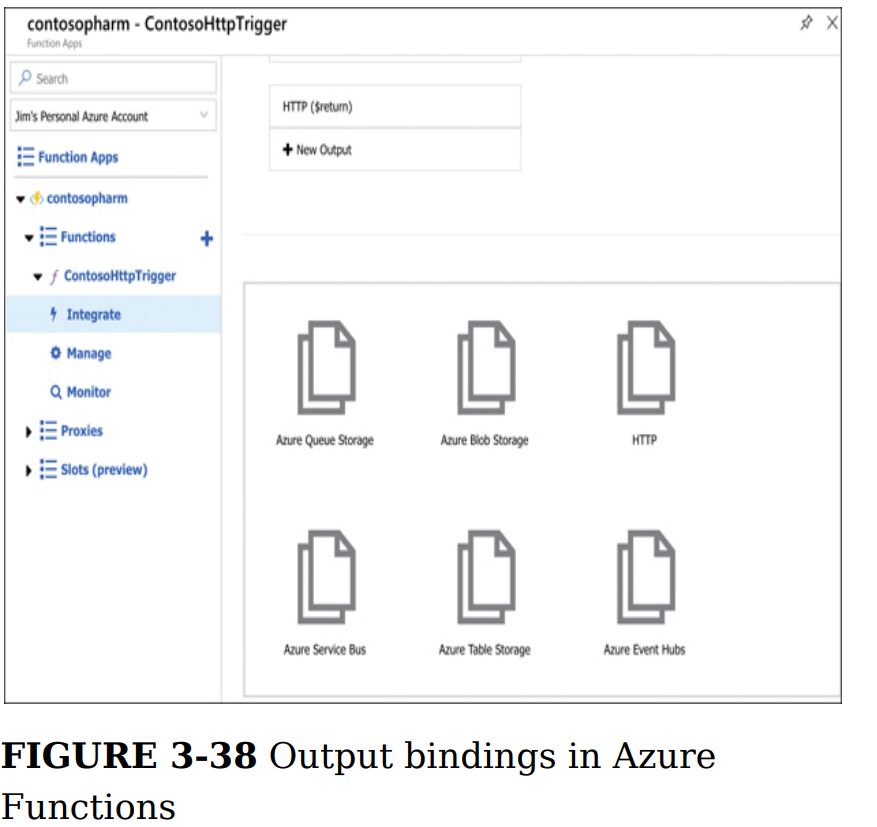
*You can choose to create your Function App in an App Service plan, but if you do, you won’t benefit from the consumption model of paying only when your code runs.*

A **proxy** allows you to configure multiple endpoints for your Function App but expose them all via a single URL

**Slots** allow you to create a copy of your Function App that isn’t public facing.

*You can configure a daily quota for your Function App. Once you reach the quota, Azure will stop the Function App until the next day.*

*If you want to protect your Function App from unauthorized use, you should use authentication features available in App Service to require authentication. You can also use Microsoft API Management to add security requirements to your Function App.*



**Logic Apps**

Logic Apps are similar to Function Apps in that they are kicked off by a trigger, but what happens after that is completely different. Unlike Function Apps, you don’t have to write code to create some powerful workflows with Logic Apps.

A workflow simply means that a Logic App reacts to something happening and responds by performing a series of tasks, such as:

* sending an email
* transferring data to a database

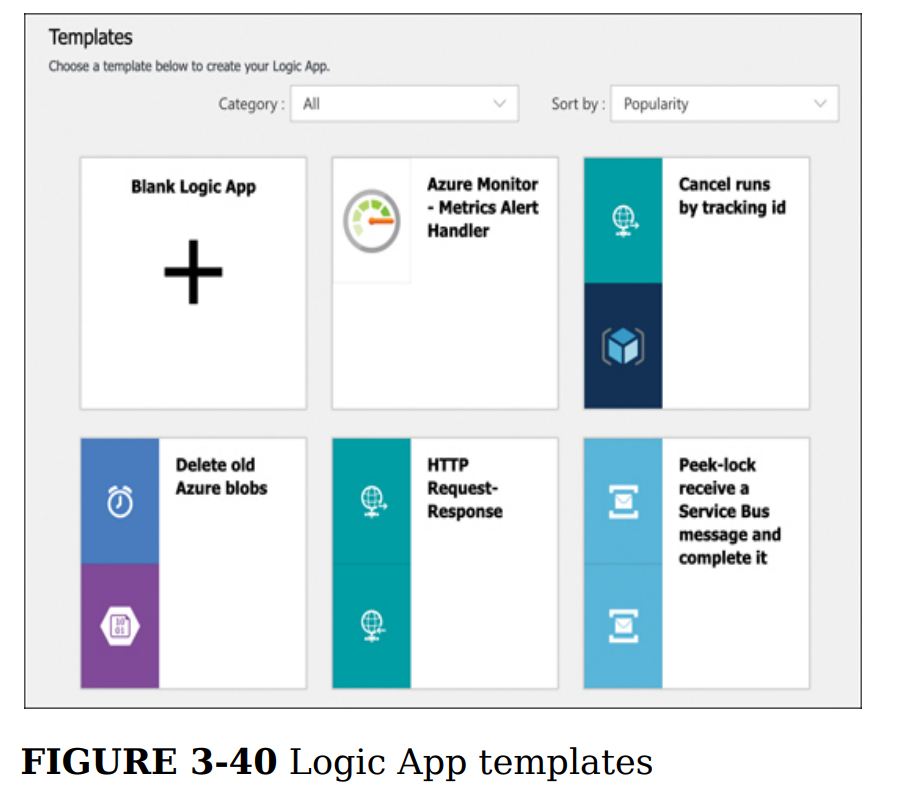
For example, when a customer orders a product from your ecommerce site, you might want to:

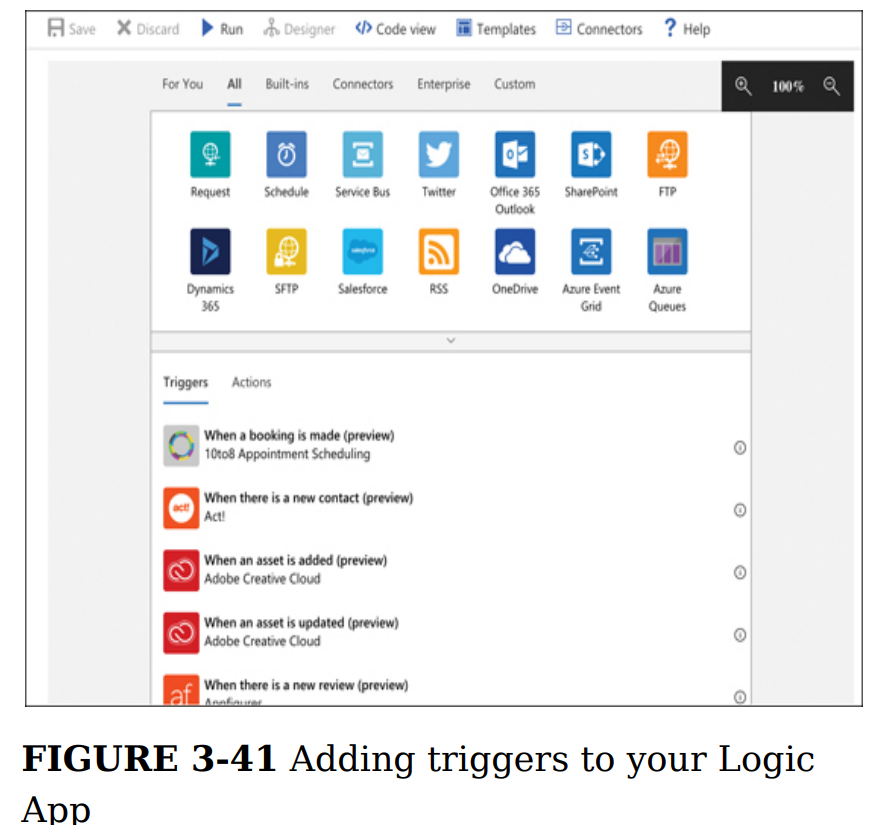
* Update your inventory count of the product
* Generate an invoice for the item
* Email the invoice to the customer
* Sign the customer up for your newsletter
* Generate a shipping label for the item

There are three components in Logic Apps that make workflows possible: connectors, triggers, and actions:

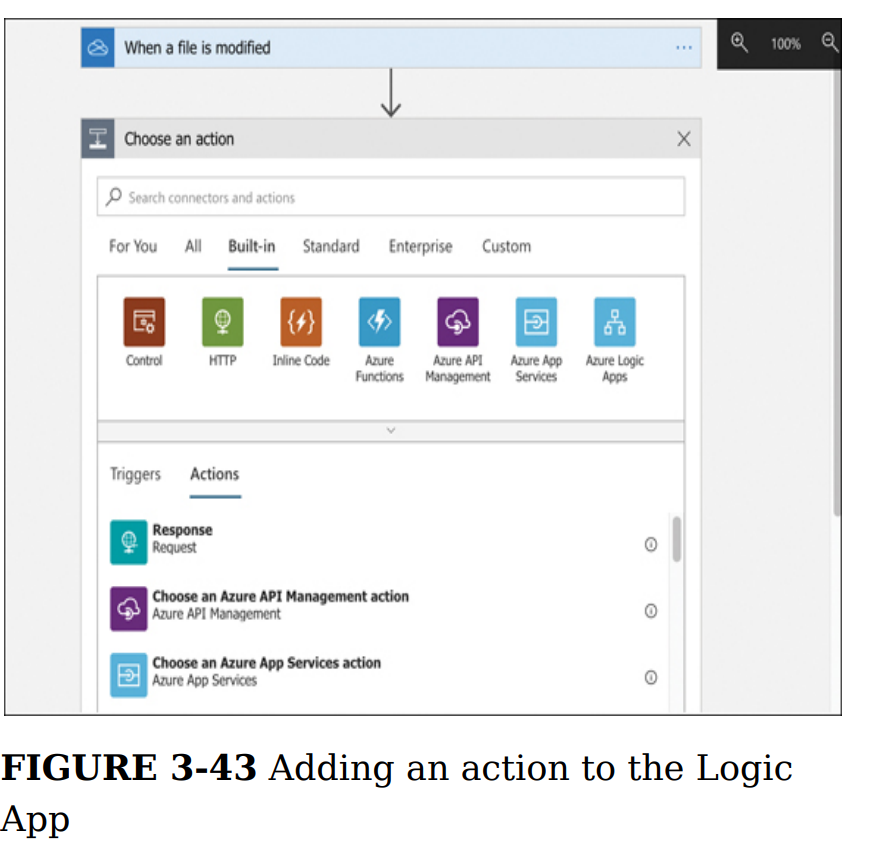
* **A connector** is a component that connects your Logic App to something, such as another Azure service, a third-party service, an FTP server, and so forth. Each connector will have one or more triggers and actions specific to that connector.
* **A trigger** is a specific action that will cause your Logic App workflow to run
* **An action** is what your Logic App will do as an output.





******

*Typically, if a development team wanted to integrate an application with a platform like Salesforce, they would have to spend a great deal of time learning Salesforce and learning how to program an application to use it. In fact, many companies would simply hire developers who already have those skills, usually at a high cost. Using Logic Apps, that same company can integrate with platforms like Salesforce and others without even having any developers at all! It’s hard to overstate the value of that kind of easy integration.*

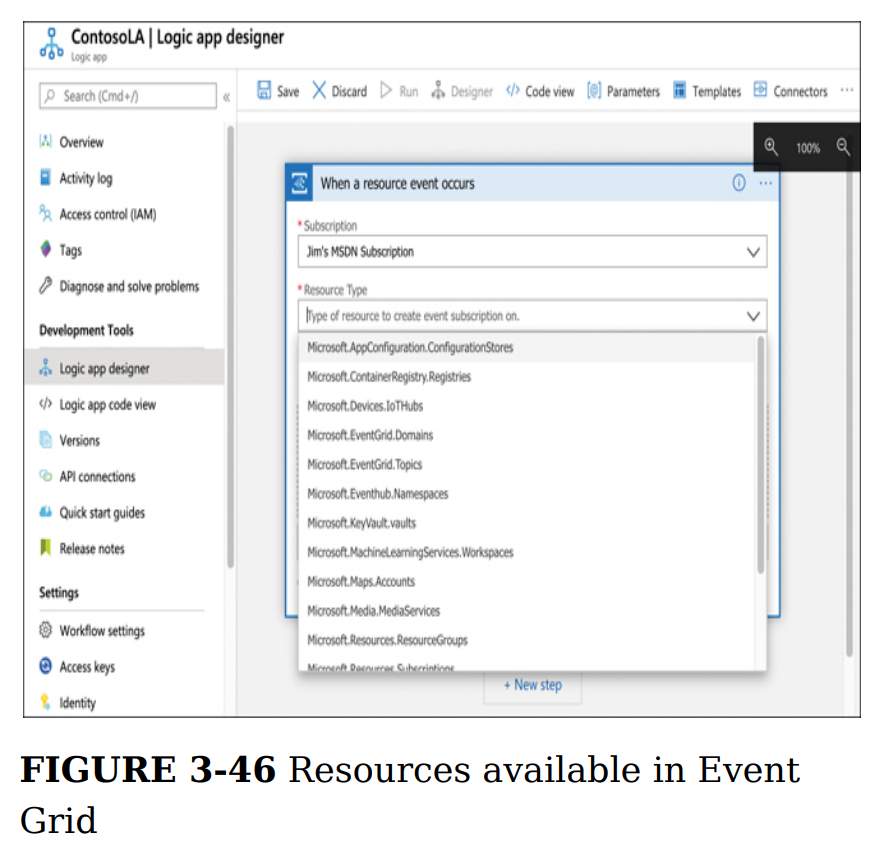
******

*As you’re configuring triggers and actions in the Logic Apps designer, Logic Apps is writing code for you under the hood that will implement your workflow. Logic App workflows are defined using JSON files, and the designer generates this* ***JSON*** *code as you are configuring your app.*

**Event Grid**

Event Grid provides that functionality: Enable an Azure service to trigger an event when something specific happens and configure another Azure service to listen for that event so it can react to it

***Both Azure Functions and Azure Logic Apps*** *are integrated with Event Grid. You can configure a function to run when an Event Grid event occurs.*

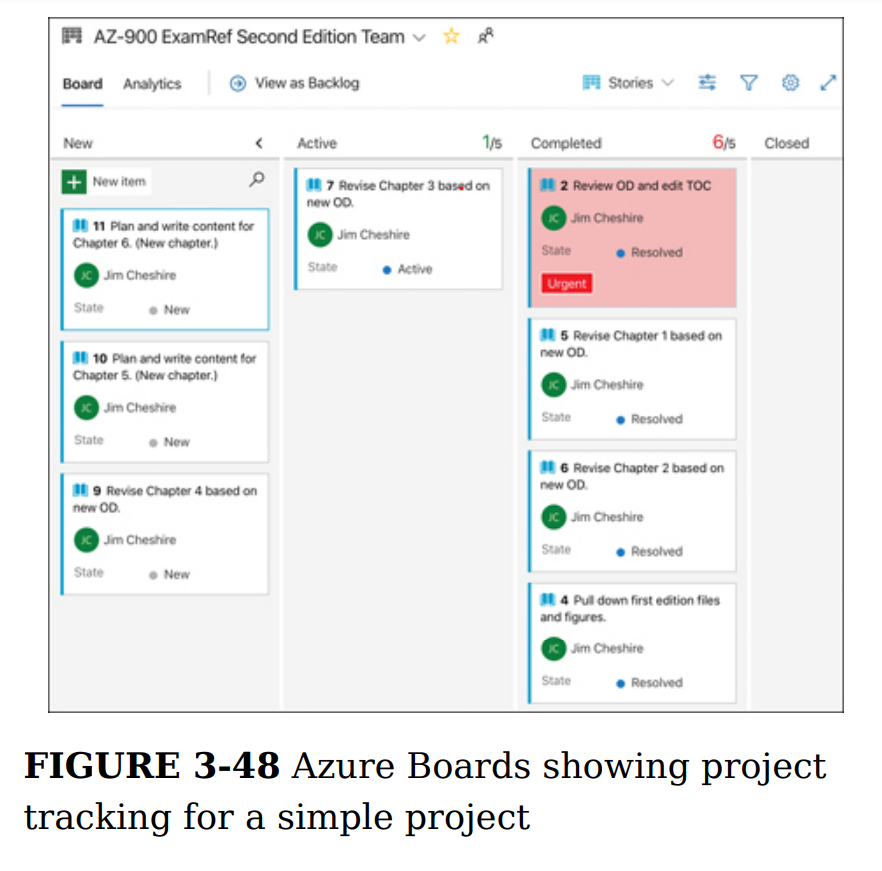
******

The primary benefit of using Event Grid in this way is the rapid development of solutions. You also benefit from Event Grid reliably triggering your events. If an Event Grid event fails to trigger for any reason, Event Grid will continue to retry triggering the event for up to 24 hours.

**Azure DevOps**

Azure DevOps offers a collection of tools that make it much easier to plan, track, and manage such projects.

* **Azure Boards** A visual way to manage and track work for your team using tiles displayed in a drag-and-drop interface
* **Azure Repos** Source and version control using either Team Foundation Version Control or Git
* **Azure Pipelines Manage** software releases through build, test, and release automation
* **Azure Test Plans** Create and track tests to ensure reliable software releases
* **Azure Artifacts** Use popular package feeds from both public and private sources

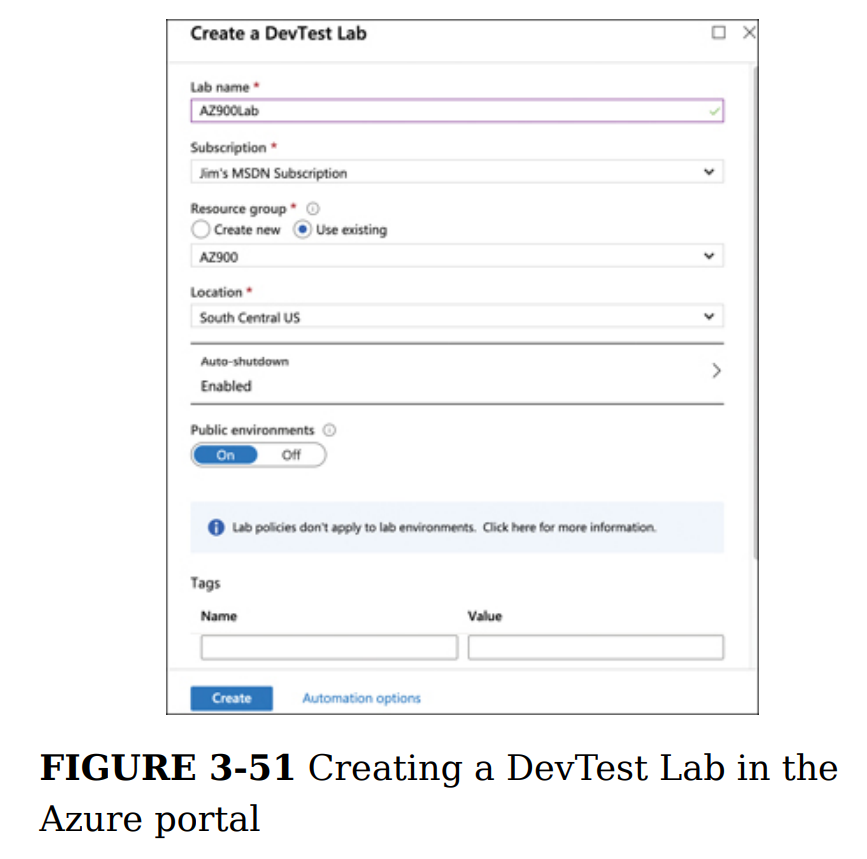
******

******

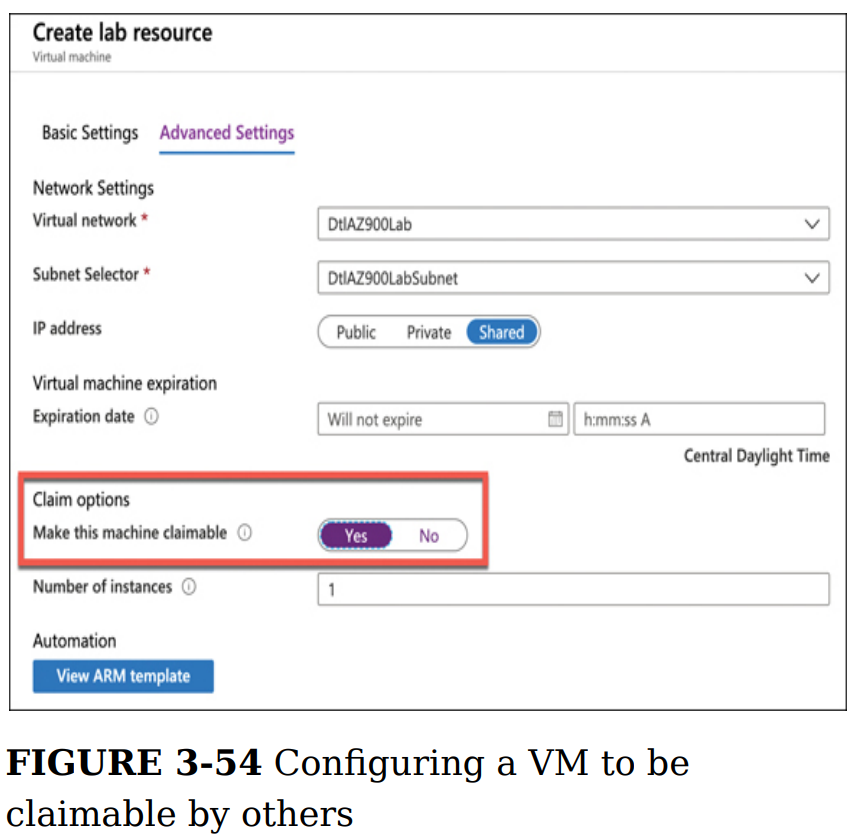
Many software applications use packaged components from third parties or from within the developer’s own organization. These packaged components are frequently referred to as artifacts, and **Azure Artifacts** provides an easy way to keep these artifacts organized. Azure Artifacts also provides the ability to integrate these packages into the build process, including builds that happen within Azure Pipelines.

**Azure DevTest Lab**

*VMs can be created in a DevTest Lab that are preconfigured for specific purposes. When developers need to use a VM for testing, they simply look at a list of available VMs and claim the VM they need. That VM is then allocated to them until they unclaim it. Claiming a VM takes only seconds because the VM doesn’t have to be created again when it’s claimed.*

****

It’s very important to understand that when a VM is created in DevTest Labs, it defaults to being dedicated to a single user and is not claimable. If you are creating a VM that you want others to be able to claim and use, you must click Advanced Settings at the top of the screen shown previously in Figure 3-53 and configure the machine to be claimable, as shown in Figure 3-54.

****

*You can’t easily add one of your existing VMs into a DevTest Lab. In order to do that, you have to jump through some hoops to copy the VM’s VHD image into the Azure Storage container that the DevTest Lab is using and then create a new VM from the custom image.*

Custom images and formulas are similar, but there is one key difference. A custom image is an image that is based on a VHD from an existing VM. A formula is also based on a VHD, but a formula also contains settings that are specific to DevTest Labs, such as VM size, included artifacts, and so forth. A formula, however, often uses a custom image as a base.