Overview of cloud computing

Cloud computing provides a modern alternative to the traditional on-premises datacenter. A public cloud vendor is completely responsible for hardware purchase and maintenance and provides a wide variety of platform services that you can use. You lease whatever hardware and software services you require on an as-needed basis, thereby converting what had been a capital expense for hardware purchase into an operational expense. It also allows you to lease access to hardware and software resources that would be too expensive to purchase. Although you are limited to the hardware provided by the cloud vendor, you only have to pay for it when you use it.

Cloud environments provide an online portal experience, making it easy for users to manage compute, storage, network, and application resources. For example, in the Azure portal, a user can create a virtual machine (VM) configuration specifying the following: the VM size (with regard to CPU, RAM, and local disks), the operating system, any predeployed software, the network configuration, and the location of the VM. The user then can deploy the VM based on that configuration and within a few minutes access the deployed VM. This quick deployment compares favorably with the previous mechanism for deploying a physical machine, which could take weeks just for the procurement cycle.

In addition to the public cloud just described, there are private and hybrid clouds. In a private cloud, you create a cloud environment in your own datacenter and provide self-service access to compute resources to users in your organization. This offers a simulation of a public cloud to your users, but you remain completely responsible for the purchase and maintenance of the hardware and software services you provide. A hybrid cloud integrates public and private clouds, allowing you to host workloads in the most appropriate location. For example, you could host a high-scale website in the public cloud and link it to a highly secure database hosted in your private cloud (or on-premises datacenter).

Microsoft provides support for public, private, and hybrid clouds. Microsoft Azure, the focus of this book, is a public cloud. Microsoft Azure Stack is an add-on to Windows Server 2016 that allows you to deploy many core Azure services in your own datacenter and provides a self-service portal experience to your users. You can integrate these into a hybrid cloud through the use of a virtual private network.

Comparison of on-premises versus Azure

With an on-premises infrastructure, you have complete control over the hardware and software that you deploy. Historically, this has led to hardware procurement decisions focused on scaling up; that is, purchasing a server with more cores to satisfy a performance need. With Azure, you can deploy only the hardware provided by Microsoft. This leads to a focus on scale-out through the deployment of additional compute nodes to satisfy a performance need. Although this has consequences for the design of an appropriate software architecture, there is now ample proof that the scale-out of commodity hardware is significantly more cost-effective than scale-up through expensive hardware.

Microsoft has deployed Azure datacenters in over 22 regions around the globe from Melbourne to Amsterdam and Sao Paulo to Singapore. Additionally, Microsoft has an arrangement with 21Vianet, making Azure available in two regions in China. Microsoft has also announced the deployment of Azure to another eight regions. Only the largest global enterprises are able to deploy datacenters in this manner, so using Azure makes it easy for enterprises of any size to deploy their services close to their customers, wherever they are in the world. And you can do that without ever leaving your office.

For startups, Azure allows you to start with very low cost and scale rapidly as you gain customers. You would not face a large up-front capital investment to create a new VM—or even several new VMs. The use of cloud computing fits well with the scale fast, fail fast model of startup growth.

Azure provides the flexibility to set up development and test configurations quickly. These deployments can be scripted, giving you the ability to spin up a development or test environment, do the testing, and spin it back down. This keeps the cost very low, and maintenance is almost nonexistent.

Another advantage of Azure is that you can try new versions of software without having to upgrade on-premises equipment. For example, if you want to see the ramifications of running your application against Microsoft SQL Server 2016 instead of Microsoft SQL Server 2014, you can create a SQL Server 2016 instance and run a copy of your services against the new database, all without having to allocate hardware and run wires. Or you can run on a VM with Microsoft Windows Server 2012 R2 instead of Microsoft Windows Server 2008 R2.

Cloud offering

Cloud computing usually is classified in three categories: SaaS, PaaS, and IaaS. However, as the cloud matures, the distinction among these is being eroded.

SaaS: Software as a service

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

Microsoft Office 365 is a prototypical model of a SaaS offering. Subscribers pay a monthly or annual subscription fee, and they get Exchange as a Service (online and/or desktop Outlook), Storage as a Service (OneDrive), and the rest of the Microsoft Office Suite (online, the desktop version, or both). Subscribers are always provided the most recent version. This essentially allows you to have a Microsoft Exchange server without having to purchase a server and install and support Exchange—the Exchange server is managed for you, including software patches and updates. Compared to installing and upgrading Office every year, this is much less expensive and requires much less effort to keep updated.

Other examples of SaaS include Dropbox, WordPress, and Amazon Kindle.

PaaS: Platform as a service

With PaaS, you deploy your application into an application-hosting environment provided by the cloud service vendor. The developer provides the application, and the PaaS vendor provides the ability to deploy and run it. This frees developers from infrastructure management, allowing them to focus strictly on development.

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

IaaS: Infrastructure as a service

An IaaS cloud vendor runs and manages server farms running virtualization software, enabling you to create VMs that run on the vendor's infrastructure. Depending on the vendor, you can create a VM running Windows or Linux and install anything you want on it. Azure provides the ability to set up virtual networks, load balancers, and storage and to use many other services that run on its infrastructure. You don't have control over the hardware or virtualization software, but you do have control over almost everything else. In fact, unlike PaaS, you are completely responsible for it.

Azure Virtual Machines, the Azure IaaS offering, is a popular choice when migrating services to Azure because it enables the "lift and shift" model for migration. You can configure a VM similar to the infrastructure currently running your services in your datacenter and migrate your software to the new VM. You might need to make tweaks, such as URLs to other services or storage, but many applications can be migrated in this manner.

Azure VM Scale Sets (VMSS) is built on top of Azure Virtual Machines and provides an easy way to deploy clusters of identical VMs. VMSS also supports autoscaling so that new VMs can be deployed automatically when required. This makes VMSS an ideal platform to host higher-level microservice compute clusters such as for Azure Service Fabric and the Azure Container Service.

Azure services

Azure includes many services in its cloud computing platform. Let's talk about a few of them.

- **Compute services** This includes the Azure Virtual Machines—both Linux and Windows, Cloud Services, App Services (Web Apps, Mobile Apps, Logic Apps, API Apps, and Function Apps), Batch (for large-scale parallel and batch compute jobs), RemoteApp, Service Fabric, and the Azure Container Service.
- **Data services** This includes Microsoft Azure Storage (comprised of the Blob,

Queue, Table, and Azure Files services), Azure SQL Database, DocumentDB, StorSimple, and the Redis Cache.

- **Application services** This includes services that you can use to help build and operate your applications, such as Azure Active Directory (Azure AD), Service Bus for connecting distributed systems, HDInsight for processing big data, Azure Scheduler, and Azure Media Services.
- **Network services** This includes Azure features such as Virtual Networks, ExpressRoute, Azure DNS, Azure Traffic Manager, and the Azure Content Delivery Network.

When migrating an application, it is worthwhile to have some understanding of the different services available in Azure because you might be able to use them to simplify the migration of your application and improve its robustness. It is impossible for us to cover everything in this book, but there are some services we felt you should know about. Chapter 9, “Additional Azure services,” provides a list of these services and a brief description of each of them.

The new world: AzureResource Manager

The Azure Resource Manager is the new methodology for deploying resources.

What is it?

Since it went into public preview, the Azure Service Management (ASM) deployment model has been used to deploy services. In the Azure portal, services managed with ASM are referred to as *classic*. In 2015, Microsoft introduced the Resource Manager deployment model as a modern, more functional replacement for ASM. The Resource Manager deployment model is recommended for all new Azure workloads.

These deployment models are often referred to as *control planes* because they are used to control services, not just to deploy them. This is different from a data plane, which manages the data used by a service.

Typically, your running Azure infrastructure will contain many resources, but some of the resources will be related to one another in some way, such as all being the component services required to run a web application. For example, you might have two VMs running the web application, using a database to store data, and residing in the same virtual network. With Resource Manager, you deploy these assets into the same resource group and manage and monitor them together. You can deploy, update, or delete all of the resources in a resource group in one operation.

In this example, the resource group would contain the following:

- VM1
- VM2
- Virtual network
- Storage account
- Azure SQL Database

You can also create a template that precisely defines all the Resource Manager resources in a deployment. You can then deploy this Resource Manager template into a resource group as a single control-plane operation, with Resource Manager in Azure ensuring that resources are deployed correctly. After deployment, Resource Manager provides security, auditing, and tagging features to help you manage your resources.

Why use Resource Manager?

There are several advantages to using Resource Manager. The deployment is faster because resources can be deployed in parallel rather than sequentially as they are in ASM. The Resource Manager model enables each service to have its own service provider, and they can update it as needed independently of the other services. Azure Storage has its own service provider, VMs have their own service provider, and so on. With the ASM model, all services had to be updated at one time, so if one service was finished and the rest were not, the one that was ready had to wait on the others before it could be released. Here are some of the other major advantages to the Resource Manager model:

- Deployment using templates
    - You can create a reusable (JSON) template that can be used to deploy all of the resources for a specific solution in one fell swoop. You no longer have to create a VM in the portal, wait for it to finish, then create the next VM, and so on.
    - You can use the template to redeploy the same resources repeatedly. For

example, you may set up the resources

    in a test environment and find that it doesn't fit your needs. You can delete the resource group, which removes all of the resources for you, then tweak your template and try again. If you only want to make changes to the resources deployed, you can just change the template and deploy it again, and Resource Manager will change the resources to conform to the new template.

    - You can take that template and easily re-create multiple versions of your infrastructure, such as staging and production. You can parameterize fields such as the VM name, network name, storage account name, etc., and load the template repeatedly, using different parameters.

- Resource Manager can identify dependencies in a template but allows you to specify additional dependencies if necessary. For example, you wouldn't want to deploy a virtual machine before creating the storage account for the VHD files that are used for the OS and data disks.

- Security

  - You can use the new Role-Based Access Control (RBAC) to control access to the resources in the group. For example, you can assign the Owner role to a user, giving that user full administrative privileges to those resources in the group but not to other resources in the subscription. Other roles include Reader (you can read anything except secrets) and Contributor (you can do most anything except add or revoke access).

- Billing

  - To help organize all of the resources in a subscription for billing purposes, you can assign tags to each resource and then retrieve all of the billing information for a specific tag.

    For example, if one department owns a web application and several related components, you can assign the same tag to all of those resources. Then, you can retrieve the billing for that department by retrieving the billing for that tag.

Note If you apply a tag to a resource group, the resources in the group do not inherit that tag. You have to apply the tag to each individual resource.

Maximize the benefits of using Resource Manager

Microsoft has several suggestions to help you maximize the use of the Resource Manager model when working with your applications and components.

- Use templates rather than using scripting like PowerShell or the Azure Command-Line Interface (CLI). Using a template allows resources to be deployed in parallel, making it much faster than using a script executed sequentially.
- Automate as much as possible by leveraging templates. You can include configurations for various extensions like PowerShell DSC and Web Deploy. This way, you don’t need any manual steps to create and configure the resources.
- Use PowerShell or the Azure CLI to manage the resources, such as to start or stop a virtual machine or application.

- Put resources with the same lifecycle in the same resource group. In our example above, what if the database is used by multiple applications? If that's true, or if the database is going to live on even after the application is retired or removed, you don't want to re-create the database every time you redeploy the application and its components. In that case, put the database in its own resource group.

Resource group tips

You can decide how to allocate your resources to resource groups based on what makes sense for you and your organization. A resource group is a logical container to hold related resources for an application or group of applications. These tips should be considered when making decisions about your resource group:

- As noted before, all of the resources in a group should have the same lifecycle.
- A resource can only be assigned to one group at a time.
- A resource can be added to or removed from a resource group at any time. Note that every resource must belong to a resource

group, so if you remove it from one group, you have to add it to another.

- Most types of resource can be moved to a different resource group at any time.
- The resources in a resource group can be in different regions.
- You can use a resource group to control access for the resources therein.

Tips for using Resource Manager templates

Resource Manager templates define the deployment and configuration of your application. They are used to deploy an application and all of its component resources repeatedly.

You can divide the deployments in a set of templates and create a master template that links in all of the required templates.

Templates can be modified and redeployed with updates. For example, you can add a new resource or update configuration information about a resource in a template. When deployed again, Resource Manager will create any new resources it finds and perform updates for any that have been changed. You will see this in Chapter 5, "Azure Virtual Networks," where you deploy a template defining a VNet with two subnets. Then, you add a third subnet and redeploy the template, and you can see the third subnet appear in the Azure portal.

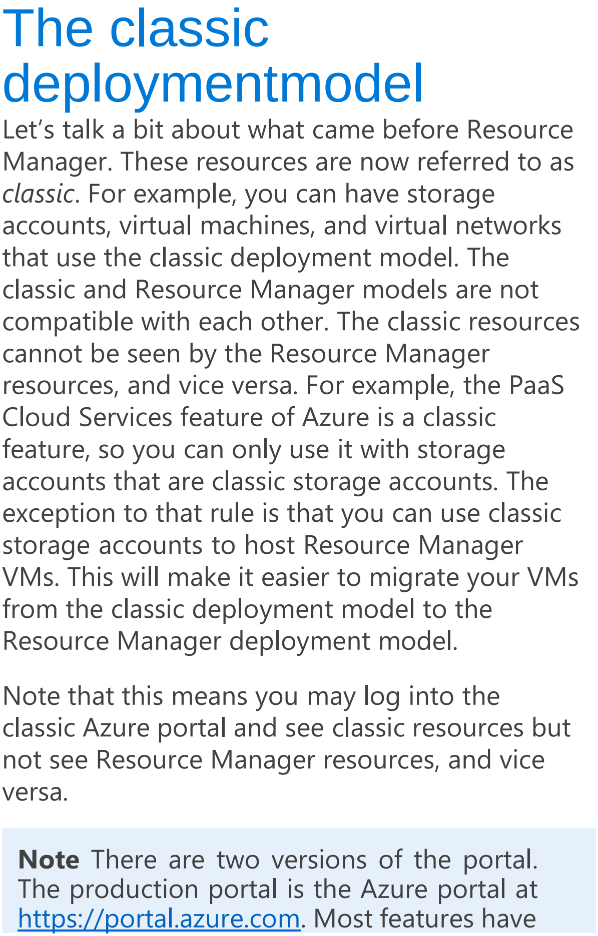
Templates can be parameterized to allow you more flexibility in deployment. This is what allows you to use the same template repeatedly but with different values, such as VM name, virtual network name, storage account name, region, and so on.

You can export the current state of the resources in a resource group to a template. This can then be used as a pattern for other deployments, or it can be edited and redeployed to make changes and additions to the current resource group's resources.

Here is an example of a JSON template. Deploying this template will create a storage account in West US called mystorage. This is parameterized; you can include a parameter file that provides the values for newStorageAccountName and location. Otherwise, it will use the defaults.

```
{

   "$schema":
"http://schema.management.azure.com/schemas/201
5-01-01/deploymentTemplate.json#",

   "contentVersion": "1.0.0.0",

   "parameters": {

    "newStorageAccountName": {

      "type": "string",

      "defaultValue": "mystorage",

      "metadata": {

        "description": "Unique DNS Name for the
Storage Account where the Virtual Machine's
disks will be placed."

      }

    },



    "location": {

      "type": "string",

      "defaultValue": "West US",

      "allowedValues": [

        "West US",

        "East US"

      ],
        "metadata": {
```

```
        "description": "Restricts choices to
where premium storage is located in the US."
      }
    }
   },

   "resources": [
     {
      "type":
"Microsoft.Storage/storageAccounts",
      "name":
"[parameters('newStorageAccountName')]",
      "apiVersion":  "2015-06-15",
      "location": "[parameters('location')]",
      "properties": {
        "accountType": "Standard_LRS"
      }
     }
  ]
}
```



The classic deploymentmodel

Let’s talk a bit about what came before Resource Manager. These resources are now referred to as *classic*. For example, you can have storage accounts, virtual machines, and virtual networks that use the classic deployment model. The classic and Resource Manager models are not compatible with each other. The classic resources cannot be seen by the Resource Manager resources, and vice versa. For example, the PaaS Cloud Services feature of Azure is a classic feature, so you can only use it with storage accounts that are classic storage accounts. The exception to that rule is that you can use classic storage accounts to host Resource Manager VMs. This will make it easier to migrate your VMs from the classic deployment model to the Resource Manager deployment model.

Note that this means you may log into the classic Azure portal and see classic resources but not see Resource Manager resources, and vice versa.

Note There are two versions of the portal. The production portal is the Azure portal at https://portal.azure.com. Most features have

> been moved to the Azure portal, with some exceptions such as Azure Active Directory (Azure AD). The previous portal is called the classic Azure portal (https://manage.windowsazure.com), and it can still be used to manage Azure AD and to configure and scale classic resources such as Cloud Services.

You can migrate your assets from the classic to the Resource Manager deployment model.

- For storage accounts, you can use AzCopy to copy blobs, files, and tables to a new Resource Manager storage account. Note that tables must be exported from the classic account and then imported into the Resource Manager account.
- For virtual machines, you can shut them down and copy their VHD file to a new Resource Manager storage account and then use the VHD file to re-create the VM.
- For virtual networks, you can re-create them as Resource Manager VNets.
- There is also a migration service that is in public preview. Microsoft recommends using this only for nonproduction workloads at this time. For more information, check out this

article:

https://azure.microsoft.com/documentation/articles/virtual-machines-windows-migration-classic-resource-manager/

PowerShell changes for the Resource Manager and classic deploymentmodels

Chapter 8, “Management tools,” talks about some of the tools available to use with Azure, including the Azure PowerShell cmdlets and the Azure CLI.

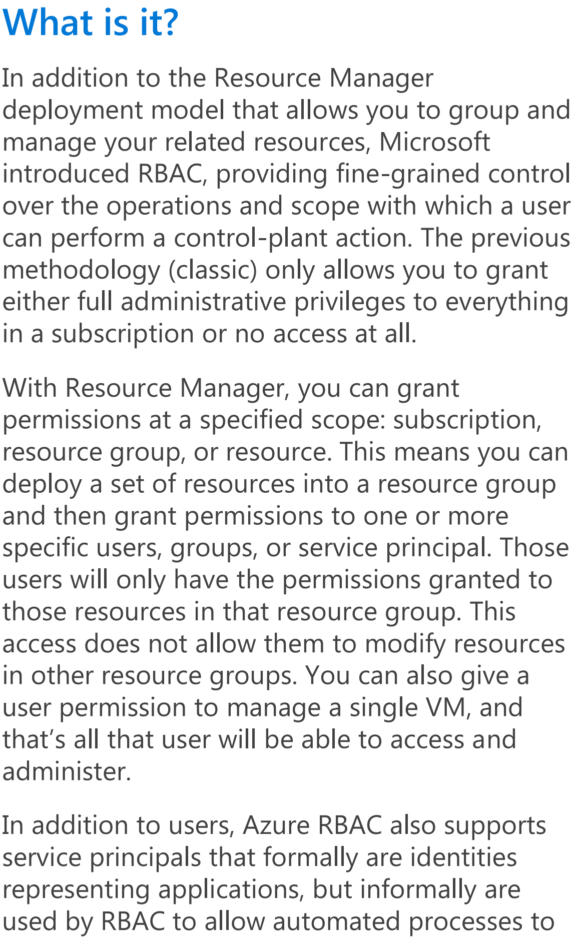
One of the other changes made when the Azure team created the Resource Manager model was to create PowerShell cmdlets that work just for the Resource Manager model. They did this by appending “Rm” to “Azure” in the name of the cmdlets. For example, to create a classic storage account, you would use the *New-AzureStorageAccount* cmdlet. To create a Resource Manager storage account, you would use the *New-AzureRmStorageAccount* cmdlet.

Microsoft did this so you could easily tell which kind of resource you were creating. Also, this ensures that scripts that are currently being used will continue to work. Each time you deploy a Resource Manager resource, you have to specify the resource group into which it should be placed. Also, some of the cmdlets for Resource Manager (such as creating a VM) have more details than their counterparts in the classic model.

One last note: for storage accounts, the only PowerShell cmdlets impacted are on the control plane, such as those for creating a storage account, listing storage accounts, removing a storage account, and so on. All of the PowerShell cmdlets used to access the actual objects in storage—blobs, tables, queues, and files—remain unchanged. So once you are pointed to the right storage account, you're good to go.

Role-Based AccessControl

In this section, we'll take a look at Role-Based Access Control (RBAC) to understand how you can use it to manage the security for your Resource Manager resources.



What is it?

In addition to the Resource Manager deployment model that allows you to group and manage your related resources, Microsoft introduced RBAC, providing fine-grained control over the operations and scope with which a user can perform a control-plant action. The previous methodology (classic) only allows you to grant either full administrative privileges to everything in a subscription or no access at all.

With Resource Manager, you can grant permissions at a specified scope: subscription, resource group, or resource. This means you can deploy a set of resources into a resource group and then grant permissions to one or more specific users, groups, or service principal. Those users will only have the permissions granted to those resources in that resource group. This access does not allow them to modify resources in other resource groups. You can also give a user permission to manage a single VM, and that’s all that user will be able to access and administer.

In addition to users, Azure RBAC also supports service principals that formally are identities representing applications, but informally are used by RBAC to allow automated processes to manage Resource Manager resources. To grant access, you assign a role to the user, group, or service principal. There are many predefined roles, and you can also define your own custom roles.

Roles

Each role has a list of Actions and Not Actions. The Actions are allowed, and the Not Actions are excluded. See https://azure.microsoft.com/documentation/articles/role-based-access-built-in-roles/ for the full list of roles and their Actions and Not Actions.

For example, there is a role called Contributor. With this role, a user can manage everything except access. This role has the following Actions and Not Actions:

- Actions: * → Can create and manage resources of all types
- Not Action: Microsoft.Authorization/*/Write → Can't create roles or assign roles
- Not Action: Microsoft.Authorization/*/Delete → Can't delete roles or role assignments

Let's take a look at some of the most common roles.

- **Owner** A user with this role can manage everything, including access. This role has no Not Actions. This is synonymous with Co-Administrator in the classic deployment model.
- **Reader** A user with this role can read resources of all types (except secrets) but can't make changes. This role will allow someone to look at the properties of a storage account, but it won't let that person retrieve the access keys.
- **SQL DB Contributor** A user with this role can manage SQL databases but not their security-related policies.
- **SQL Security Manager** A user with this role can manage the security-related policies of SQL Servers and databases.
- **Storage Account Contributor** A user with this role can manage storage accounts but cannot manage access to the storage accounts. This means the user with this role can't assign any roles to any users for the storage account. Note that the user with this role *can* retrieve the access keys for the storage account, which means they have full access to the data in the storage account.

- **Virtual Machine Contributor** A user with this role can manage virtual machines but can't manage the VNet to which they are connected or the storage account where the VHD file resides. Note that this role *does* include access to the storage account keys, which is needed to create the container for the VHD files as well as the VHD files themselves.

These are only a few of the many roles that can be assigned to a user, a group of users, or an application.

Custom roles

If none of the built-in roles and no combination of the built-in roles provides exactly what you need, you can create a custom role. You can do this using PowerShell, the Azure CLI, or the REST APIs. Once you create a custom role, you can assign it to a user, group, or application for a subscription, resource group, or resource. Custom roles are stored in the Azure AD and can be shared across all subscriptions that use the same Active Directory.

For example, you could create a custom role for monitoring and restarting virtual machines. Here are the Actions you would assign to that role:

- Microsoft.Storage/*/read
- Microsoft.Network/*/read
- Microsoft.Compute/*/read
- Microsoft.Compute/virtualMachines;/start/action
- Microsoft.Compute/virtualMachines/restart/action
- Microsoft.Authorization/*/read
- Microsoft.Resources/subscriptions/resourceGroups/read
- Microsoft.Insights/alertRules/*
- Microsoft.Insights/diagnosticSettings/*
- Microsoft.Support/*

Note that as requested, this role can only start and restart virtual machines. It can't create them or delete them.

A convenient way to create a custom role is to download the definition of an existing role and use that as a starting point. When you create a custom role, you also need to specify in which subscriptions it can be used—at least one must be specified.

In the next section, we'll see how to assign roles to users for a resource group and how to give full administrative privileges for a subscription to a user.

The Azure portal

An online management portal provides the easiest way to manage the resources you deploy into Azure. You can use this to create virtual networks, set up Web Apps, create VMs, define storage accounts, and so on, as listed in the previous section.

As noted earlier in this chapter, there are currently two versions of the portal. The production portal is the Azure portal at https://portal.azure.com. Most features have been moved to the Azure portal, with some exceptions such as Azure AD. The previous portal is called the classic Azure portal (https://manage.windowsazure.com), and it can still be used to manage Azure AD and to configure and scale classic resources such as Cloud Services.

In most cases, you will be using the Azure portal, so that's what we're going to focus on in this book. All of the resources that use the Resource

Manager deployment model can only be accessed in the Azure portal.

Let’s take a look at the Azure portal and how you navigate through it.

Dashboard and hub

The Azure portal is located at https://portal.azure.com. When you open this the first time, it will look similar to Figure 1-1.

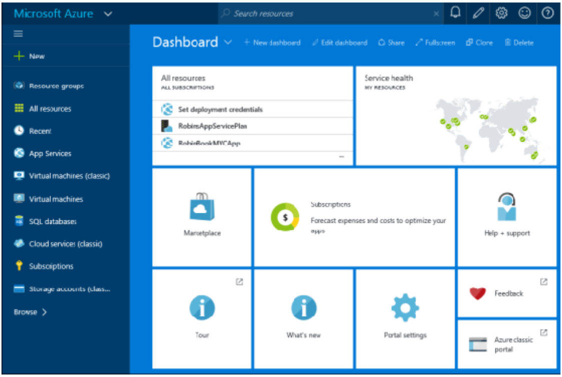


Figure 1-1 Azure portal.

This is called your Dashboard. The column on the left is called a hub; it shows you a core set of options such as Resource Groups, All Resources, and Recent. The other items on this hub are resources you have selected and/or used before.

For example, I have recently created some App Services and VMs. You can click any of these, and it will show the resources you have for that type. For example, if you click SQL Databases, it will show a list of your SQL Databases.

You can customize the list of resources that show up in that left hub. If you click Browse, you will see a selection screen showing all of the options, and you can select which ones you want to appear, as displayed in Figure 1-2.

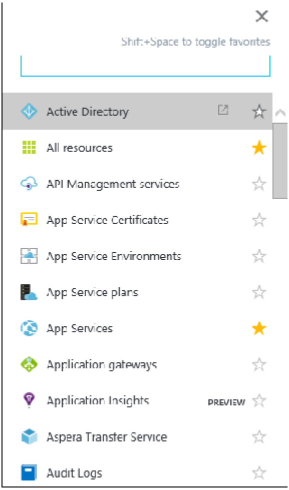


Figure 1-2 Configure default hub in the Azure portal.

The area on the right with the tiles is called your

Dashboard. You can customize this by adding tiles, removing tiles, resizing tiles, and so on by selecting Edit Dashboard, as shown in Figure 1-3.

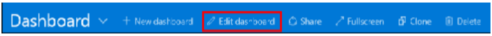


Figure 1-3 How to edit the Dashboard in the Azure portal.

As you create resources, you can choose to pin them to the Dashboard, and it will add them to this section.

There are a couple of default tiles on the Dashboard that are of interest.

- **All Resources** Clicking this will bring up a list of all of your resources.
- **Service Health** This shows the health of the regions around the world. If you click this, it will show a list of the regions, and you can select one to get more detailed information.
- **Marketplace** This will take you directly to the Marketplace blade where you can search for and add resources.
- **Subscriptions** This shows the subscriptions that can be managed by the account you are using. You can select a subscription and see the billing information for the current month. If you have a starting credit, this will

show the amount of credit left. Accounts having starting credit include MSDN accounts and BizSpark accounts.

- **Help + Support** This takes you to the blade where you can submit a new support request and manage the requests you have already put in. It also provides links to the MSDN forums and StackOverflow where you can post questions.

Now, let's look at the icons in the upper-right corner of the Azure portal, as shown in Figure 1-4.

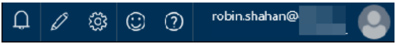


Figure 1-4 Notifications, settings, etc. in the Azure portal.

From left to right, here's what these icons mean:

- Clicking the bell shows notifications from this session. For example, if you create a new VM, when it's finished, it will put a notification here.

- Clicking the pencil puts the Dashboard into edit mode, just like clicking Edit Dashboard above.

- Clicking the gear icon brings up the Settings screen for the portal, where you can do things like enable or disable toast notifications, set the default language, and so on.

- Clicking the smiley face will show a dialog you can use to send feedback to the portal team.
- Clicking the question mark will show a drop-down menu allowing you to create a new support request, view your current support requests, and so on.
- The last field shows the account you have used to log into the portal. If you administer more than one subscription, this will show the list of Azure ADs to which the user belongs. You can click this to sign out, change your password, or submit an idea.

Creating and viewing resources

As you make selections, the portal scrolls to the right. The separate sections that get opened are called blades.

Click New in the main hub. You see a categorized list of the resources available, as shown in Figure 1-5. This is a new blade.

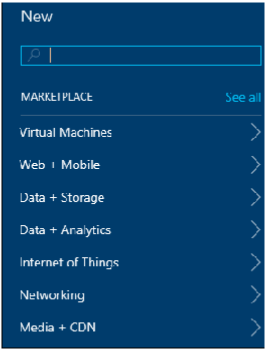


Figure 1-5 Creating a new resource in the Azure portal.

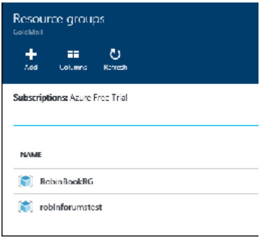
If you click See All, it will take you to the Azure Marketplace. The Marketplace contains all of the resources that you can use in Azure. This includes everything from VM images, which are certified before being made available, all of the SQL Server options, and Web Apps. It also includes applications such as Drupal and WordPress. To add any resource, you can search for it, then select it to add it to your Azure subscription.

You can also select a category on this blade. It will show the list of resources valid for that category, and you can then select which one you want to create. For example, to create a VM, you would click the Virtual Machines category; to create a storage account or a SQL Server, you would click Data + Storage.

Once you have created some resources, there are several ways to view them. Let’s look back in the main hub (Figure 1-1), which has two helpful options—Resource Groups and All Resources.

View by resource group

Use this option to see all of your resources by resource group. Click Resource Groups, and you see a blade like Figure 1-6 showing all of your resource groups.

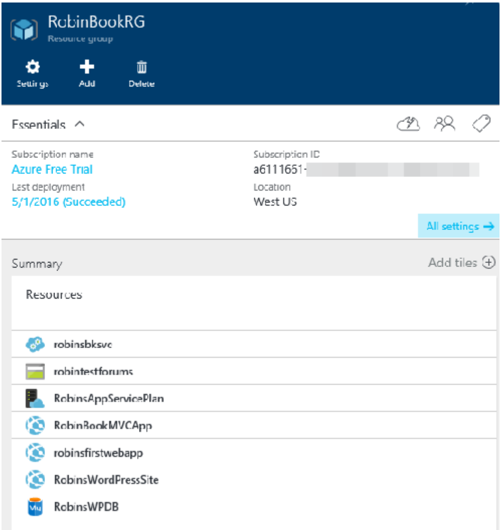


Subscriptions: Azure Free Trial

NAME
RobinBookRG
robInforumstest

Figure 1-6 Screenshot showing all of your resource groups in the Azure portal.

Next, select one of the resource groups, and it shows all of the resources deployed to that group (Figure 1-7).



Subscription name
Azure Free Trial

Last deployment
5/1/2016 (Succeeded)

Subscription ID
a6111651-[redacted]

Location
West US

robinsbksvc
robintestforums
RobinsAppServicePlan
RobinBookMVCApp
robinsfirstwebapp
RobinsWordPressSite
RobinsWPDB

Figure 1-7 List of resources in the selected resourcegroup.

You can click any of the resources here, and they will be displayed in a new blade.

Click All Settings to show the Settings blade (Figure 1-8). From there, you can look at the costs by resource, view the deployment history of the resources, set tags and locks, and manage what users have access to this resource group.

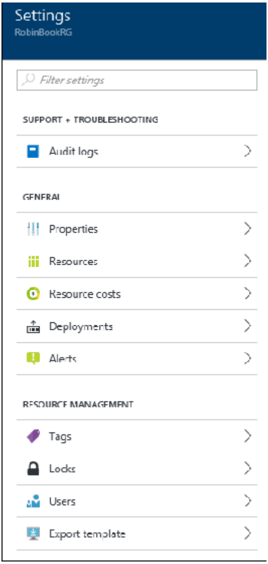
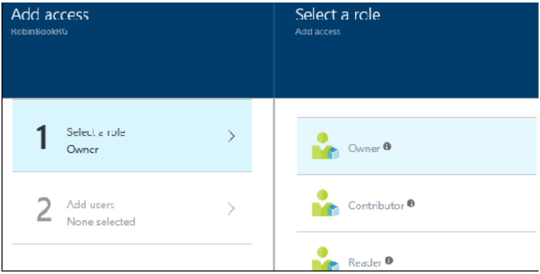


Figure 1-8 Settings blade when looking at resources ina resource group.

This is where you can use RBAC to control access to all of the resources in the same resource group at one time by assigning roles to users. The user has to be set up in the Azure AD, which is done in the classic Azure portal (https://manage.windowsazure.com).

Let's give VM Contributor access to another user account. This is granting the ability to manage the VMs but not the ability to manage the access to the VMs. So this new user could not grant access to anybody else. If you want someone to have full administrative privileges of all the resources in the resource group, you can grant that user the Owner role.

In the Users blade, click Add. You are prompted to select the role you want the user to have (Figure 1-9).

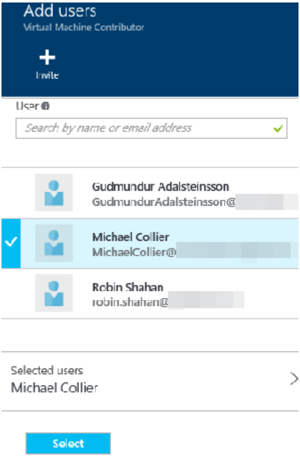


1 Select a role
Owner

2 Add users
None selected

Figure 1-9 Select a role to assign to a new user.

Look through the list and find the Virtual Machine Contributor role and select it. The Add Access blade highlights Add Users and shows a list of users to the right from which to select (Figure 1-10). Select an account and then click Select at the bottom of the blade.



User

Search by name or email address

Gudmundur Adalsteinsson
GudmundurAdalsteinsson@[redacted]

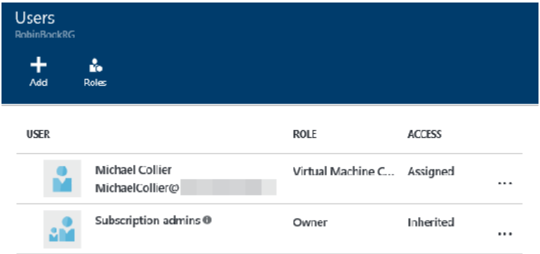
✓ Michael Collier
MichaelCollier@[redacted]

Robin Shahan
robin.shahan@[redacted]

Selected users
Michael Collier

Figure 1-10 Select a user to add.

Next, click OK on the Add Access blade. It returns to the Users screen, which now reflects the user(s) added and their roles (Figure 1-11).



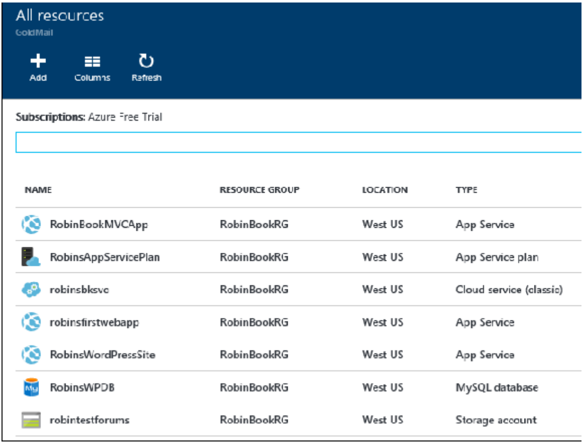
USER	ROLE	ACCESS	
Michael Collier MichaelCollier@[redacted]	Virtual Machine C...	Assigned	...
Subscription admins	Owner	Inherited	...

Figure 1-11 List of users and their assigned roles.

I added the Virtual Machine Contributor role for Michael Collier. This means that Michael Collier now has the ability to manage the VMs in that resource group.

View by resource

Back in the main hub (Figure 1-1), let's look at the other view of our resources. Click All Resources. This shows exactly what you expect—a list of all your resources (Figure 1-12). You can edit the columns by selecting Columns. I've added the Type column because I can never remember what all of the icons mean.



All resources

GoldMail

Subscriptions: Azure Free Trial

NAME	RESOURCE GROUP	LOCATION	TYPE
RobinBookMVCApp	RobinBookRG	West US	App Service
RobinsAppServicePlan	RobinBookRG	West US	App Service plan
robinsblksvc	RobinBookRG	West US	Cloud service (classic)
robinsfirstwebapp	RobinBookRG	West US	App Service
RobinsWordPressSite	RobinBookRG	West US	App Service
RobinsWPDB	RobinBookRG	West US	MySQL database
robintestforums	RobinBookRG	West US	Storage account

Figure 1-12 List of resources in the subscription.

Clicking any resource brings up a blade for that specific resource.