Introduction

3 minutes

Imagine you work for a company that does video data processing and pattern analysis. You are building a new prototype platform to process the video from traffic cameras, analyze trends, and provide actionable data for traffic and road improvements.

To improve your algorithms, you have made arrangements with several new cities to collect their traffic camera data. However not all of the video data is in the same format, and many of the formats only have Windows codecs to decode the data. Because of this, you have decided to use Virtual Machines (VMs) to do the initial processing and then push the data onto Azure Functions that will process a standard format. This approach will allow you to bring on new data formats dynamically without stopping the entire system.

Azure provides a robust virtual machine hosting solution that can meet your needs. Let's explore how to create and work with Windows virtual machines in Azure.

Learning objectives

In this module, you will:

- Understand the options that are available for virtual machines in Azure.
- Create a Windows virtual machine using the Azure portal.
- Connect to a running Windows virtual machine using Remote Desktop.
- Install software and change the network configuration on a VM using the Azure portal.

Prerequisites

- Basic understanding of Azure Virtual Machines from **Introduction to Azure Virtual Machines**

- Remote Desktop client

Next unit: Create a Windows virtual machine in Azure

Continue >

Create a Windows virtual machine in Azure

10 minutes

Your company has decided to manage the video data from their traffic cameras in Azure using VMs. In order to run the multiple codecs, we first need to create the VMs. We also need to connect and interact with the VMs. In this unit, you will learn how to create a VM using the Azure portal. You will configure the VM for remote access, select a VM image, and choose the proper storage option.

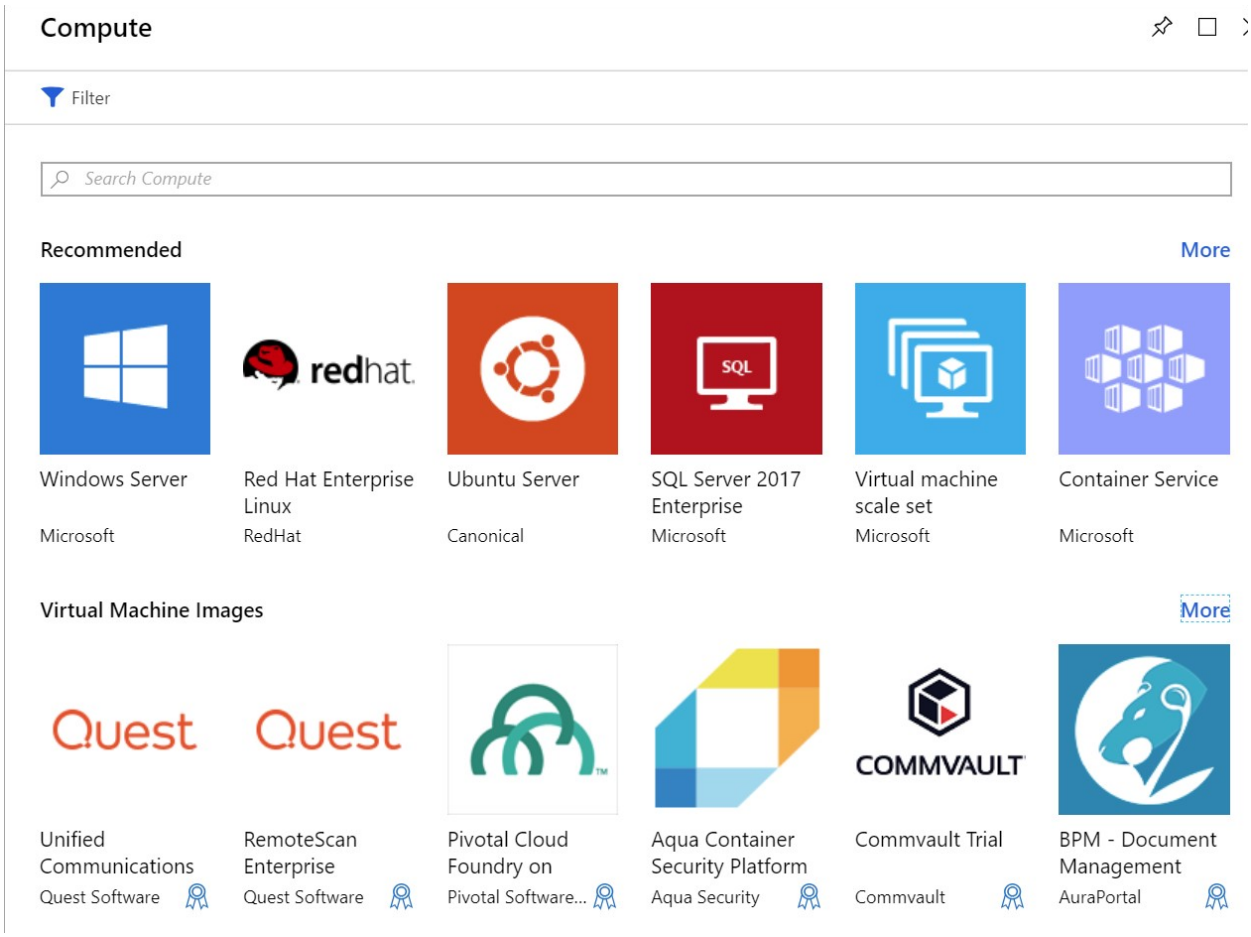
Introduction to Windows virtual machines in Azure

Azure VMs are an on-demand scalable cloud computing resource. They're similar to virtual machines that are hosted in Windows Hyper-V. They include processor, memory, storage, and networking resources. You can start and stop virtual machines at will, just like with Hyper-V, and manage them from the Azure portal or with the Azure CLI. You can also use a Remote Desktop Protocol (RDP) client to connect directly to the Windows desktop user interface (UI) and use the VM as if you were signed in to a local Windows computer.

Creating an Azure VM

VMs can be defined and deployed on Azure in several ways: the Azure portal, a script (using the Azure CLI or Azure PowerShell), or through an Azure Resource Manager template. In all cases, you will need to supply several pieces of information, which we'll cover shortly.

The Azure Marketplace also provides pre-configured images that include both an OS and popular software tools installed for specific scenarios.



Resources used in a Windows VM

When creating a Windows VM in Azure, you also create resources to host the VM. These resources work together to virtualize a computer and run the Windows operating system. These must either exist (and be selected during VM creation), or they will be created with the VM.

- A virtual machine that provides CPU and memory resources.
- An Azure Storage account to hold the virtual hard disks.
- Virtual disks to hold the OS, applications, and data.
- Virtual network (VNet) to connect the VM to other Azure services or your own on-premises hardware.
- A network interface to communicate with the VNet.
- A public IP address so you can access the VM. This is optional.

Like other Azure services, you'll need a **Resource Group** to contain the VM (and optionally group these resources together for administration). When you create a new VM, you can either use an existing resource group or create a new one.

Choose the VM image

Selecting an image is one of the first and most important decisions you'll make when creating a VM. An image is a template that's used to create a VM. These templates include an OS and often other software, such as development tools or web hosting environments.

Any application that can be supported by the computer can be included in the VM image. You can create a VM from an image that's pre-configured to exactly match your requirements, such as hosting an ASP.NET Core app.

> **Tip**
>
> You can also create and upload your own images, check the documentation for more information.

Sizing your VM

Just as a physical machine has a certain amount of memory and CPU power, so does a virtual machine. Azure offers a range of VMs of differing sizes at different price points. The size that you choose will determine the VMs processing power, memory, and max storage capacity.

> **⚠ Warning**
>
> There are quota limits on each subscription that can impact VM creation. By default, you cannot have more than 20 virtual *cores* across all VMs within a region. You can either split up VMs across regions or file an **online request** to increase your limits.

VM sizes are grouped into categories, starting with the B-series for basic testing and running up to the H-series for massive computing tasks. You should select the size of the VM based on the workload you want to perform. It is possible to change the size of a VM after it's been created, but the VM must be stopped first so it's best to size it appropriately from the start if possible.

Here are some guidelines based on the scenario you are targeting.

What are you doing?	Consider these sizes
General use computing / web Testing and development, small to medium databases, or low to medium traffic web servers.	B, Dsv3, Dv3, DSv2, Dv2
Heavy computational tasks Medium traffic web servers, network appliances, batch processes, and application servers.	Fsv2, Fs, F
Large memory usage Relational database servers, medium to large caches, and in-memory analytics.	Esv3, Ev3, M, GS, G, DSv2, Dv2
Data storage and processing Big Data, SQL, and NoSQL databases, which need high disk throughput and IO.	Ls
Heavy graphics rendering or video editing, as well as model training and inferencing (ND) with deep learning.	NV, NC, NCv2, NCv3, ND
High-performance computing (HPC) If you need the fastest and most powerful CPU virtual machines with optional high-throughput network interfaces.	H

Choosing storage options

The next set of decisions revolves around storage. First, you can choose the disk technology. Options include a traditional platter-based hard disk drive (HDD) or a more modern solid-state drive (SSD). Just like the hardware you purchase, SSD storage costs more but provides better performance.

> **Tip**
>
> There are two levels of SSD storage available: standard and premium. Choose Standard SSD disks if you have normal workloads but want better performance. Choose Premium SSD disks if you have I/O intensive workloads or mission-critical systems that need to process data very quickly.

Mapping storage to disks

Azure uses virtual hard disks (VHDs) to represent physical disks for the VM. VHDs replicate the logical format and data of a disk drive but are stored as page blobs in an Azure Storage account. You can choose on a per-disk basis what type of storage it should use (SSD or HDD). This allows you to control the performance of each disk, likely based on the I/O you plan to perform on it.

By default, two virtual hard disks (VHDs) will be created for your Windows VM:

1. The **Operating System disk**. This is your primary or C: drive and has a maximum capacity of 2048 GB.
2. A **Temporary disk**. This provides temporary storage for the OS or any apps. It is configured as the D: drive by default and is sized based on the VM size, making it an ideal location for the Windows paging file.

> **⚠ Warning**
>
> The temporary disk is not persistent. You should only write data to this disk that you are willing to lose at any time.

What about data?

You can store data on the C: drive along with the OS, but a better approach is to create dedicated *data disks*. You can create and attach additional disks to the VM. Each disk can hold up to 4095 GB of data, with the maximum amount of storage determined by the VM size you select.

> **Note**
>
> An interesting capability is to create a VHD image from a real disk. This allows you to easily migrate *existing* information from an on-premises computer to the cloud.

Unmanaged vs. Managed disks

The final storage choice you'll make is whether to use **unmanaged** or **managed** disks.

With unmanaged disks, you are responsible for the storage accounts that are used to hold the VHDs that correspond to your VM disks. You pay the storage account rates for the amount of space you use. A single storage account has a fixed rate limit of 20,000 I/O operations/sec. This means that a single storage account is capable of supporting 40 standard virtual hard disks at full throttle. If you need to scale out, then you need more than one storage account, which can get complicated.

Managed disks are the newer and recommended disk storage model. They elegantly solve this complexity by putting the burden of managing the storage accounts onto Azure. You specify the disk type (Premium or Standard) and the size of the disk and Azure creates and manages both the disk *and* the storage it uses. You don't have to worry about storage account limits, which makes them easier to scale out. They also offer several other benefits:

- **Increased reliability**: Azure ensures that VHDs associated with high-reliability VMs will be placed in different parts of Azure storage to provide similar levels of resilience.
- **Better security**: Managed disks are truly managed resources in the resource group. This means they can use role-based access control to restrict who can work with the VHD data.
- **Snapshot support**: Snapshots can be used to create a read-only copy of a VHD. You have to shut down the owning VM but creating the snapshot only takes a few seconds. Once it's done, you can power on the VM and use the snapshot to create a duplicate VM to troubleshoot a production issue or rollback the VM to the point in time that the snapshot was taken.

- **Backup support**: Managed disks can be automatically backed up to different regions for disaster recovery with Azure Backup all without affecting the service of the VM.

Network communication

Virtual machines communicate with external resources using a virtual network (VNet). The VNet represents a private network in a single region that your resources communicate on. A virtual network is just like the networks you manage on-premises. You can divide them up with subnets to isolate resources, connect them to other networks (including your on-premises networks), and apply traffic rules to govern inbound and outbound connections.

Planning your network

When you create a new VM, you will have the option of creating a new virtual network, or using an existing VNet in your region.

Having Azure create the network together with the VM is simple but it's likely not ideal for most scenarios. It's better to plan your network requirements *up-front* for all the components in your architecture and create the VNet structure you will need separately. Then create the VMs and place them into the already-created VNets.

We'll look more at virtual networks a bit later in this module. Let's apply some of this knowledge and create a VM in Azure.

Next unit: Exercise - Create a Windows virtual machine

Continue >

Exercise - Create a Windows virtual machine

10 minutes

This module requires a sandbox to complete. You have used 2 of 10 sandboxes for today. More sandboxes will be available tomorrow.

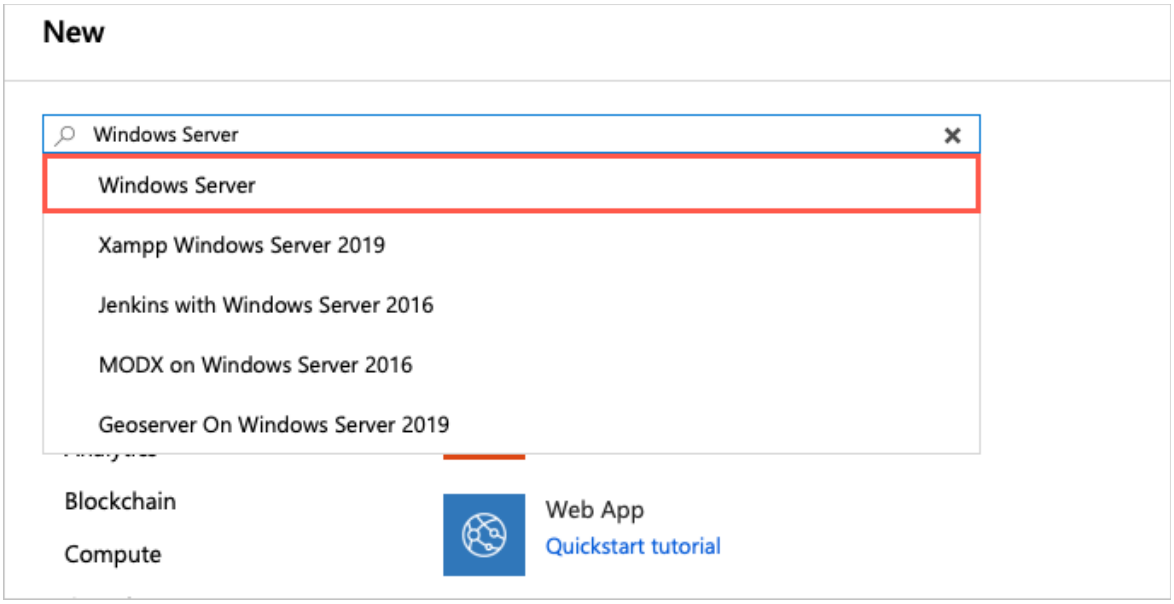
Activate sandbox

Recall that our company processes video content on Windows VMs. A new city has contracted us to process their traffic cameras, but it's a model we've not worked with before. We need to create a new Windows VM and install some proprietary codecs so we can begin processing and analyzing their images.

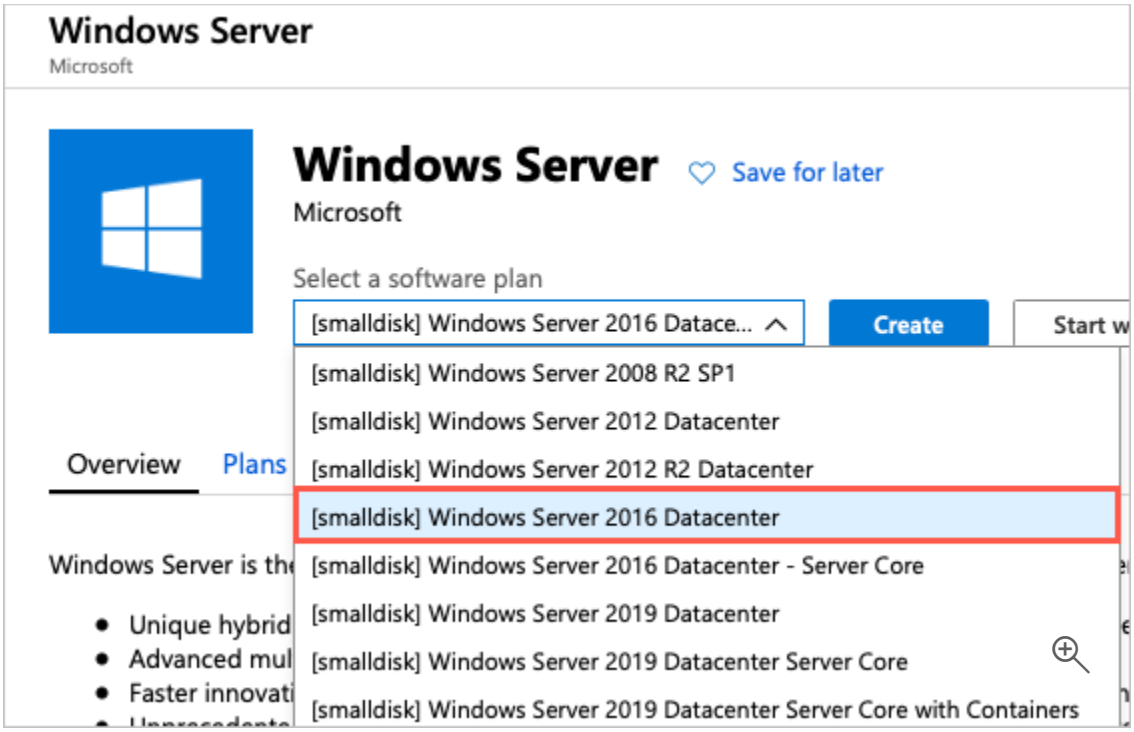
Create a new Windows virtual machine

We can create Windows VMs with the Azure portal, Azure CLI, or Azure PowerShell. The easiest approach is the portal because it walks you through the required information and provides hints and helpful messages during the creation of the VM.

1. Sign into the Azure portal using the same account you activated the sandbox with.
2. On the Azure portal menu or from the **Home** page, select **Create a resource**.
3. In the search box, enter **Windows Server** and then click on the link with the same title in the presented list.



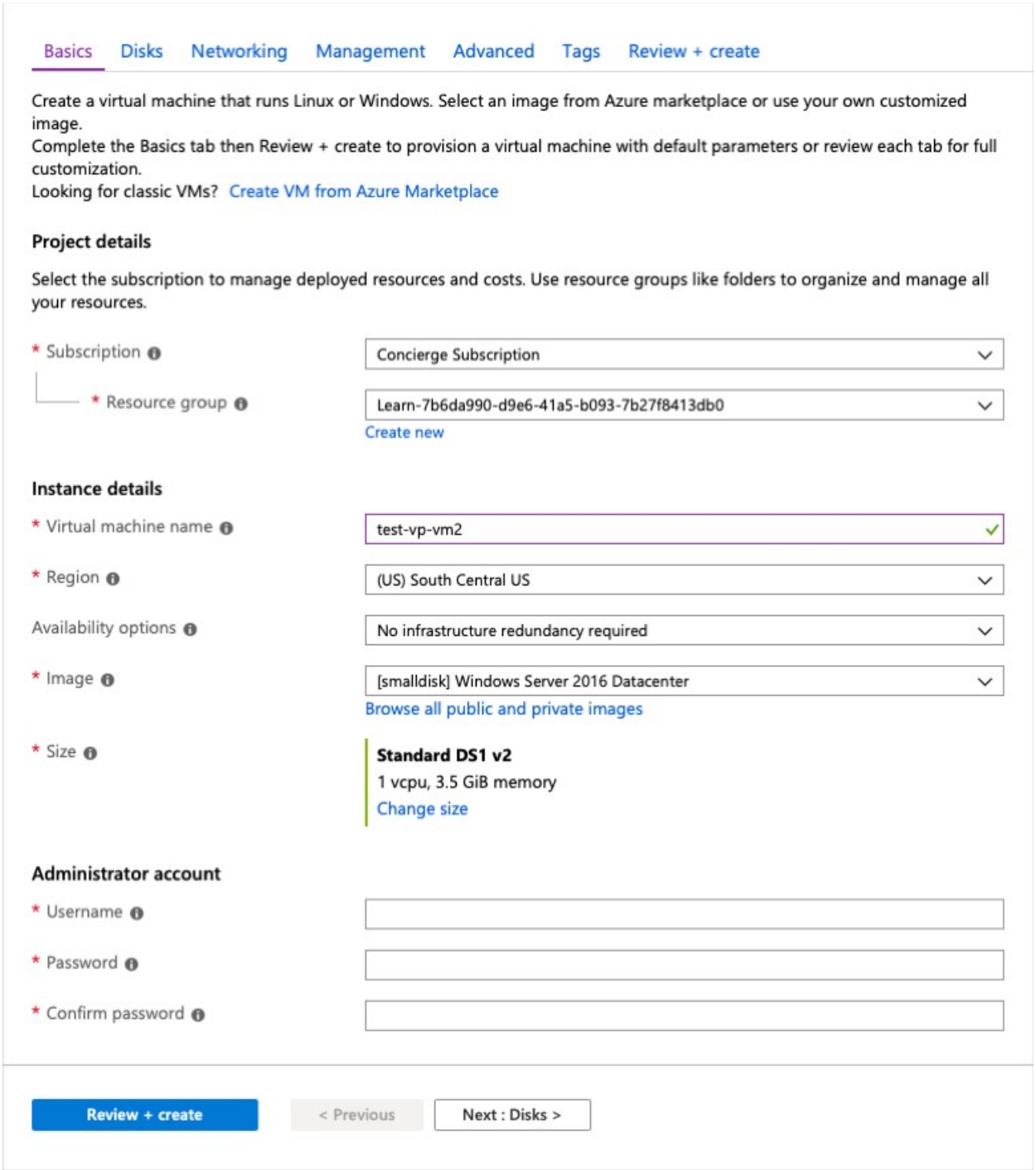
4. There are several Windows Server versions we can select from to create our VM. In the *Windows Server* image overview panel, click on the **Select a software plan** dropdown list and find the **[smalldisk] Windows Server 2016 Datacenter** option.



5. Click the **Create** button to start configuring the VM.

Configure the VM settings

The VM creation experience in the portal is presented in a "wizard" format to walk you through all the configuration areas for the VM. Clicking the "Next" button will take you to the next configurable section. However, you can move between the sections at will with the tabs running across the top that identify each section.



Once you fill in all the required options (identified with red stars), you can skip the remainder of the wizard experience and start creating the VM through the **Review + Create** button at the bottom.

We'll start with the **Basics** section.

Configure basic VM settings

> **Note**
>
> As you change settings and tab out of each free-text field, Azure will validate each value automatically and place a green check mark next to it when it's good. You can hover over error indicators to get more information on issues it discovers.

1. Select the **Subscription** that should be billed for VM hours.
2. For **Resource group**, choose "**[sandbox resource group name]**".
3. In the **Instance Details** section, enter a name for your VM, such as **test-vp-vm2** (for Test Video Processor VM #2).
    - It's best practice to standardize your resource names so you can easily identify their purpose. Windows VM names are a bit limited - they must be between 1 and 15 characters, cannot contain non-ASCII or special characters, and must be unique in the current resource group.
4. Select a region close to you from the locations below.

    The free sandbox allows you to create resources in a subset of the Azure global regions. Select a region from the following list when you create resources:

    - West US 2
    - South Central US
    - Central US
    - East US
    - West Europe
    - Southeast Asia
    - Japan East

5. Leave **Availability options** as "No Infrastructure redundancy required". This option is used to ensure the VM is highly available by grouping multiple VMs together a set to deal with planned or unplanned maintenance events or outages.

6. Ensure the image is set to "[smalldisk] Windows Server 2016 Datacenter". You can open the drop-down list to see all the options available.

7. The **Size** field is not directly editable and has a DS1 default size. Click the **Change size** link to explore other VM sizes. The resulting dialog allows you to filter based on # of CPUs, Name, and Disk Type. Select "Standard DS1 v2" (normally the default) when you are done. That will give the VM 1 CPU and 3.5 GB of memory.

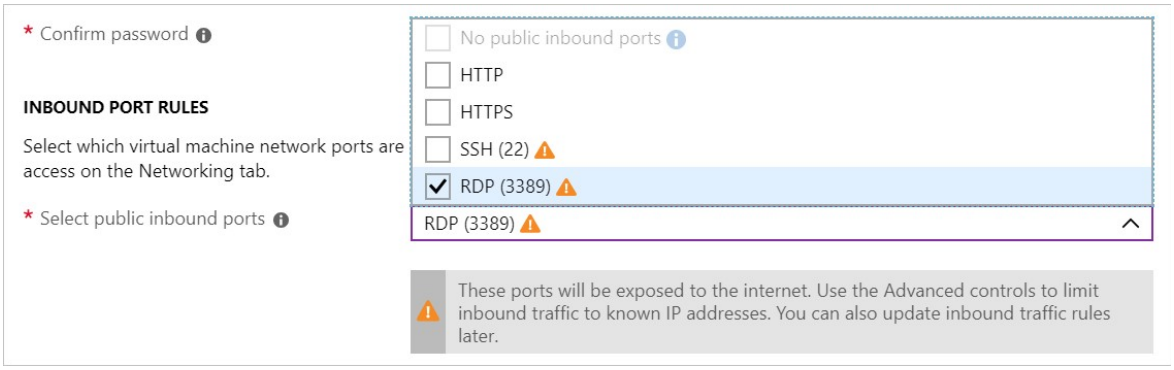
   > **Tip**
   >
   > You can also just slide the view to the left to get back to the VM settings as it opened a new window off to the right and slid the window over to view it.

8. In the **Administrator Account** section, set the **Username** field to a username you will use to sign in to the VM.

9. In the **Password** field, enter a password that's at least 12 characters long. It must have three of the following: one lower case character, one uppercase character, one number, and one special character that is not '\' or '-'. Use something you will remember or write it down, you will need it later.

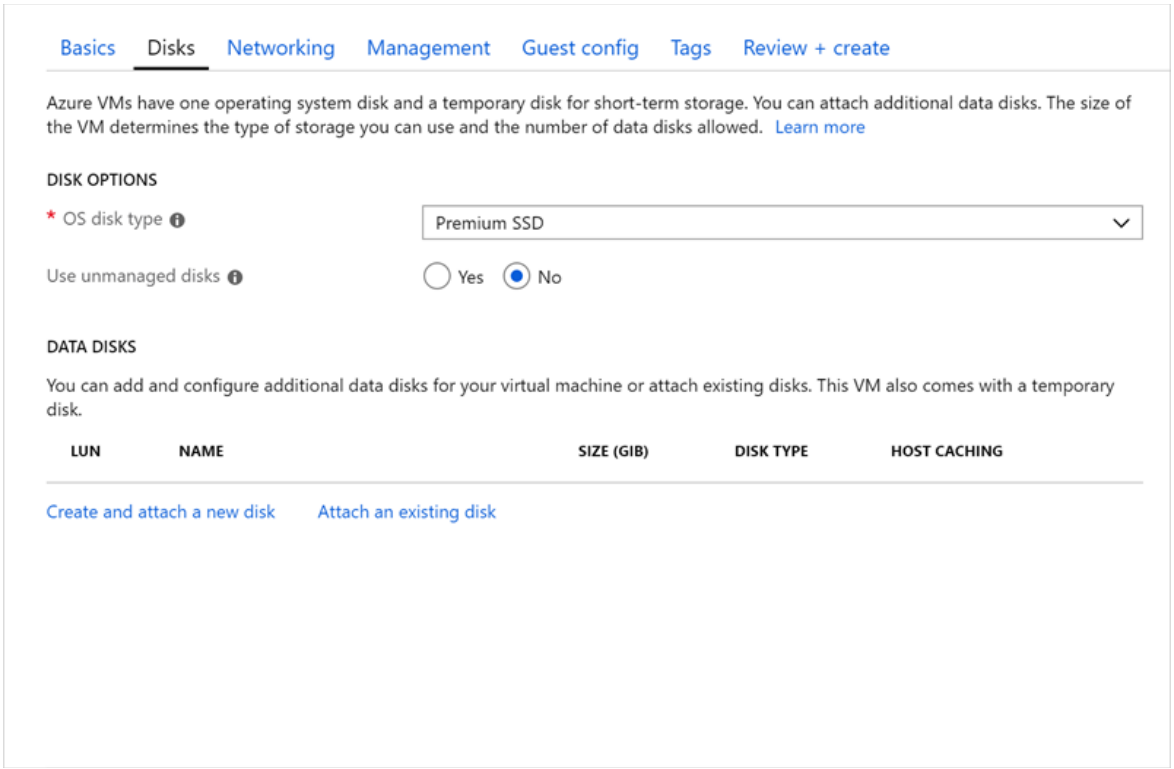
10. Confirm the **password**.

11. In the **Inbound Port Rules** section, open the list and choose *Allow selected ports*. Since this is a Windows VM, we want to be able to access the desktop using RDP. Scroll the list if necessary until you find RDP (3389) and select it. As the note in the UI indicates, we can also adjust the network ports after we create the VM.



Configure Disks for the VM

1. Click **Next** to move to the Disks section.



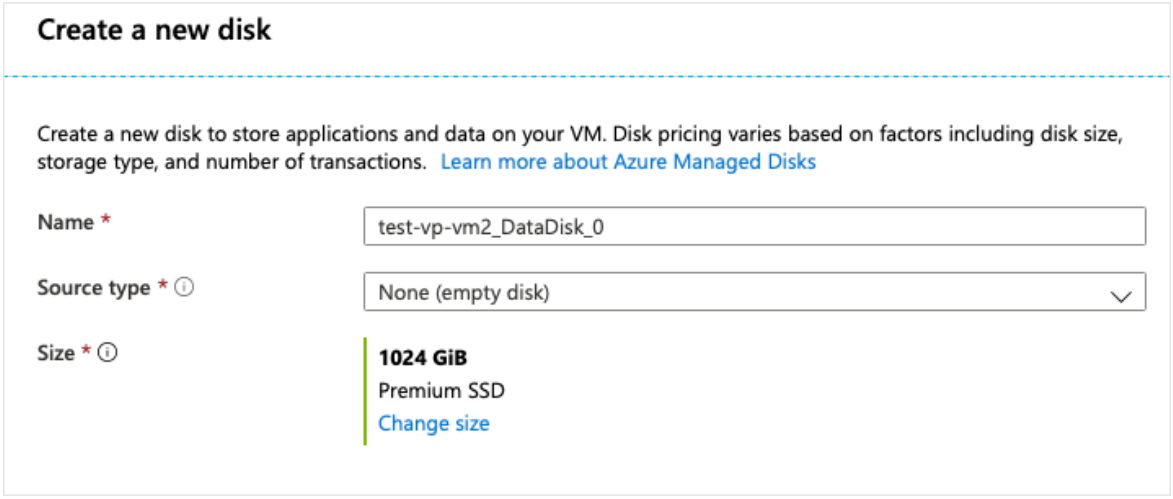
2. Choose "Premium SSD" for the **OS disk type**.

3. Use managed disks so we don't have to work with storage accounts. You can flip the switch in the GUI to see the difference in information that Azure needs if you like.

Create a data disk

Recall we will get an OS disk (C:) and Temporary disk (D:). Let's add a data disk as well.

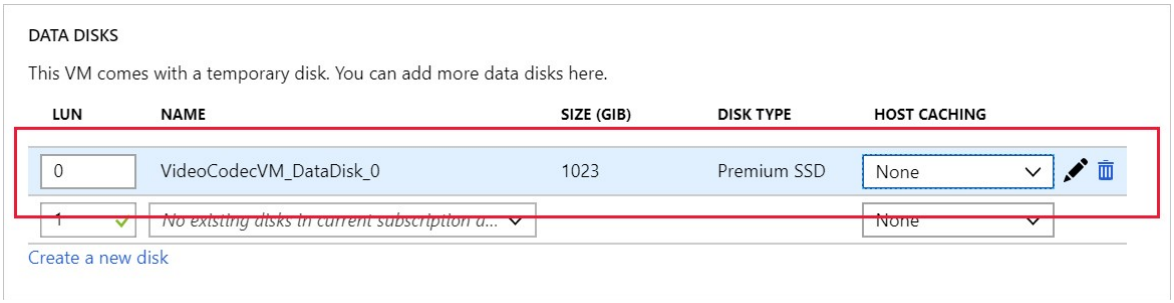
1. Click the **Create and attach a new disk** link in the **DATA DISKS** section.



2. You can take all the defaults: Premium SSD, 1023 GB, and None (empty disk); although notice that here is where we could use a snapshot, or Storage Blob to create a VHD.

3. Click **OK** to create the disk and go back to the **DATA DISKS** section.

4. There should now be a new disk in the first row.



Configure the Network

1. Click **Next** to move to the Networking section.

2. In a production system, where we already have other components, we'd want to utilize an *existing* virtual network. That way our VM can communicate with the

other cloud services in our solution. If there isn't one defined in this location yet, we can create it here and configure the:

- **Address space**: the overall IPV4 space available to this network.
- **Subnet range**: the first subnet to subdivide the address space - it must fit within the defined address space. Once the VNet is created you can add additional subnets.

3. Let's change the default ranges to use the `172.xxx` IP address space. Click **Create New** under Virtual Network.

   - Change the **Address space** field to be `172.16.0.0/16` to give it the full range of addresses
   - Change the **Subnet range** field to be `172.16.1.0/24` to give it 256 IP addresses of the space.

4. Click **OK**.

> **Note**
>
> By default, Azure will create a virtual network, network interface, and public IP for your VM. It's not trivial to change the networking options after the VM has been created so always double-check the network assignments on services you create in Azure.

Finish configuring the VM and create the image

The rest of the options have reasonable defaults and there's no need to change any of them. You can explore the other tabs if you like. The individual options have an `(i)` icon next to them that will show a help bubble to explain the option. This is a great way to learn about the various options you can use to configure the VM.

1. Click the **Review + create** button at the bottom of the panel.

2. The system will validate your options and give you details about the VM being created.

3. Click **Create** to create and deploy the VM. The Azure dashboard will show the VM that's being deployed. This may take several minutes.

While that's deploying, let's look at what we can do with this VM.

Next unit: Use RDP to connect to Windows Azure virtual machines

Continue >

Use RDP to connect to Windows Azure virtual machines

10 minutes

Now that we have a Windows VM in Azure, the next thing you'll do is put your applications and data on those VMs to process our traffic videos.

However, unless you've set up a site-to-site VPN to Azure, your Azure VMs won't be accessible from your local network. If you're just getting started with Azure, it's unlikely that you have a working site-to-site VPN. So how can you transfer files to Azure VMs? One easy way is to use Azure's Remote Desktop Connections feature to share your local drives with your new Azure VMs.

Now that we have a new Windows virtual machine, we need to install our custom software on to it. There are several ways we can do this.

- Remote Desktop Protocol (RDP)
- Custom scripts
- Custom VM images (with the software preinstalled)

Let's look at the simplest approach for Windows VMs: Remote Desktop.

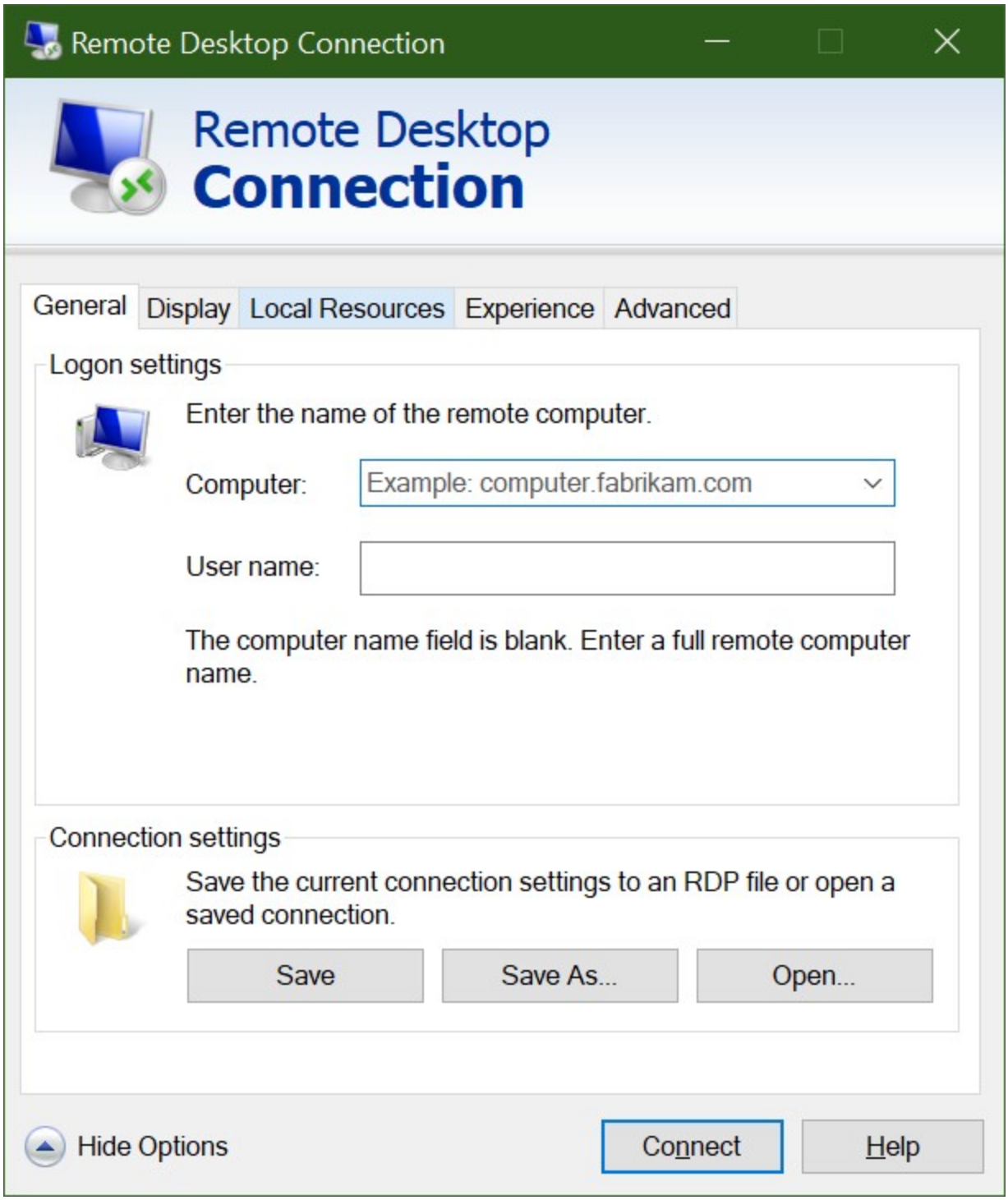
What is the Remote Desktop Protocol?

Remote Desktop (RDP) provides remote connectivity to the UI of Windows-based computers. RDP enables you to sign in to a remote physical or virtual Windows computer and control that computer as if you were seated at the console. An RDP connection enables you to carry out the vast majority of operations that you can do from the console of a physical computer, with the exception of some power and hardware-related functions.

An RDP connection requires an RDP client. Microsoft provides RDP clients for the following operating systems:

- Windows (built-in)
- MacOS
- iOS
- Android

The following screenshot displays the Remote Desktop Protocol client in Windows 10.



There are also open source Linux clients, such as Remmina that enable you to connect to a Windows PC from an Ubuntu distribution.

Connecting to an Azure VM

As we saw a moment ago, Azure VMs communicate on a virtual network. They can also have an optional public IP address assigned to them. With a public IP, we can communicate with the VM over the Internet. Alternatively, we can setup a virtual private network (VPN) that connects our on-premises network to Azure - letting us securely connect to the VM without exposing a public IP. This approach is covered in another module and is fully documented if you are interested in exploring that option.

One thing to be aware of with public IP addresses in Azure is they are often dynamically allocated. That means the IP address can change over time - for VMs this happens when the VM is restarted. You can pay more to assign static addresses if you want to connect directly to an IP address instead of a name and need to ensure that the IP address will not change.

How do you connect to a VM in Azure using RDP?

Connecting to a VM in Azure using RDP is a simple process. In the Azure portal, you go to the properties of your VM, and at the top, click **Connect**. This will show you the IP addresses assigned to the VM and give you the option to download a preconfigured **.rdp** file that Windows then opens in the RDP client. You can choose to connect over the public IP address of the VM in the RDP file. Alternatively, if you're connecting over VPN or ExpressRoute, you can select the internal IP address. You can also select the port number for the connection.

If you're using a static public IP address for the VM, you can save the **.rdp** file to your desktop. If you're using dynamic IP addressing, the **.rdp** file only remains valid while the VM is running. If you stop and restart the VM, you must download another **.rdp** file.

> **Tip**
>
> You can also enter the public IP address of the VM into the Windows RDP client and click **Connect**.

When you connect, you will typically receive two warnings. These are:

-**Publisher warning** - caused by the **.rdp** file not being publicly signed.

- **Certificate warning** - caused by the machine certificate not being trusted.

In test environments, these warnings can be ignored. In production environments, the **.rdp** file can be signed using **RDPSIGN.EXE** and the machine certificate placed in the client's **Trusted Root Certification Authorities** store.

Let's try using RDP to connect to our VM.

Next unit: Exercise - Connect to a Windows virtual machine using RDP

Continue >

100 XP

Exercise - Connect to a Windows virtual machine using RDP

10 minutes

This module requires a sandbox to complete. You have used 2 of 10 sandboxes for today. More sandboxes will be available tomorrow.

Activate sandbox

We have our Windows VM deployed and running, but it's not configured to do any work.

Recall our scenario is a video processing system. Our platform receives files through FTP. The traffic cameras upload video clips to a known URL, which is mapped to a folder on the server. The custom software on each Windows VM runs as a service and watches the folder and processes each uploaded clip. It then passes the normalized video to our algorithms running on other Azure services.

There are a few things we would need to configure to support this scenario:

- Install FTP and open the ports it needs to communicate.
- Install the proprietary video codec unique to the city's camera system.
- Install our transcoding service that processes uploaded videos.

Many of these are typical administrative tasks we won't actually cover here, and we don't have software to install. Instead, we will walk through the steps and show you how you *could* install custom or third-party software using Remote Desktop. Let's start by getting the connection information.

Connect to the VM with Remote Desktop Protocol

To connect to an Azure VM with an RDP client, you will need:

- The public IP address of the VM (or private if the VM is configured to connect to your network).
- The port number.

You can enter this information into the RDP client, or download a pre-configured **RDP** file.

> **Note**
>
> An **RDP** file is a text file that contains a set of name/value pairs that define the connection parameters for an RDP client to connect to a remote computer using the Remote Desktop Protocol.

Download the RDP file

1. In the Azure portal, ensure the **Overview** panel for the virtual machine that you created earlier is open. You can find the VM under **All Resources** if you need to open it. The overview panel has a lot of information about the VM.

    - You can see whether the VM is running.
    - Stop or restart it.
    - Get the public IP address to connect to the VM.
    - See the activity of the CPU, disk, and network.

2. Click the **Connect** button at the top of the pane.

3. In the **Connect to virtual machine** pane, note the **IP address** and **Port number** settings, then click **Download RDP File** and save it to your computer.

4. Before we connect, let's adjust a few settings. On Windows, find the file using Explorer, right-click and select **Edit**. On MacOS you will need to open the file first

with the RDP client and then right-click on the item in the displayed list and select **Edit**.

5. You can adjust a variety of settings to control the experience in connecting to the Azure VM. The settings you will want to examine are:

    - **Display**: By default, it will be full screen. You can change this to a lower resolution, or use all your monitors if you have more than one.
    - **Local Resources**: You can share local drives with the VM - allowing you to copy files from your PC to the VM. Click the **More** button under **Local devices and resources** to select what is shared.
    - **Experience**: Adjust the visual experience based on your network quality.

6. Share your Local C: drive so it will be visible to the VM.

7. Switch back to the **General** tab and click **Save** to save the changes. You can always come back and edit this file later to try other settings.

Connect to the Windows VM

1. Click the **Connect** button to start the connection to the VM.

2. In the **Remote Desktop Connection** dialog box, note the security warning and the remote computer IP address, then click **Connect**.

3. In the **Windows Security** dialog box, enter your username and password that you used in steps 6 and 7.

    > ⓘ **Note**
    >
    > If you are using a Windows client to connect to the VM, it will default to known identities on your machine. You can click the **More choices** option and select "Use a different account" to let you enter a different username/password combination.

4. In the second **Remote Desktop Connection** dialog box, note the certificate errors, then click **Yes**.

Install worker roles

The first time you connect to a Windows server VM, it will launch Server Manager. This allows you to assign a worker role for common web or data tasks. You can also launch the Server Manager through the Start Menu.

This is where we would add the Web Server role to the server. This will install IIS and as part of the configuration you would turn off HTTP requests and enable the FTP server. Or, we could ignore IIS and install a third-party FTP server. We'd then configure the FTP server to allow access to a folder on our big data drive we added to the VM.

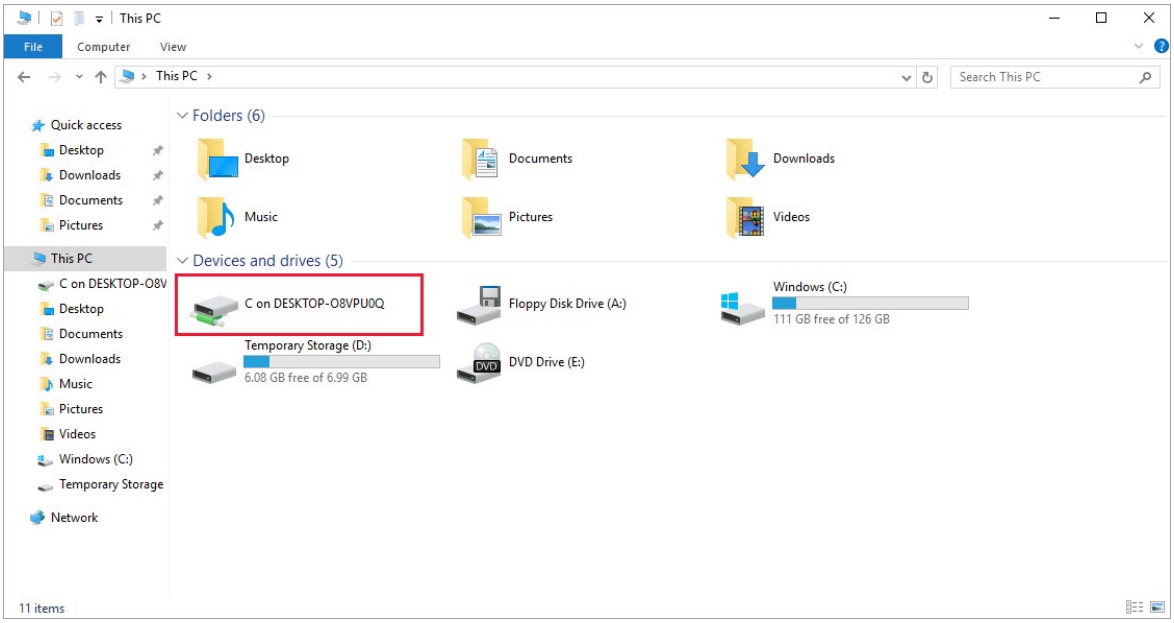
Since we aren't going to actually configure that here, just close Server Manager.

Install custom software

We have two approaches we can use to install software. First, this VM is connected to the Internet. If the software you need has a downloadable installer, you can open a web browser in the RDP session, download the software, and install it. Second, if your software is custom - like our custom service, you can copy it from your local machine over to the VM to install it. Let's look at this latter approach.

1. Open File Explorer. Click on **This PC** in the sidebar. You should see several drives:

    - Windows (C:) drive representing the OS.
    - Temporary Storage (D:) drive.
    - Your local C: drive (it will have a different name than shown below).



With access to your local drive, you can copy the files for the custom software onto the VM and install the software. We won't actually do that since it's just a simulated scenario, but you can imagine how it would work.

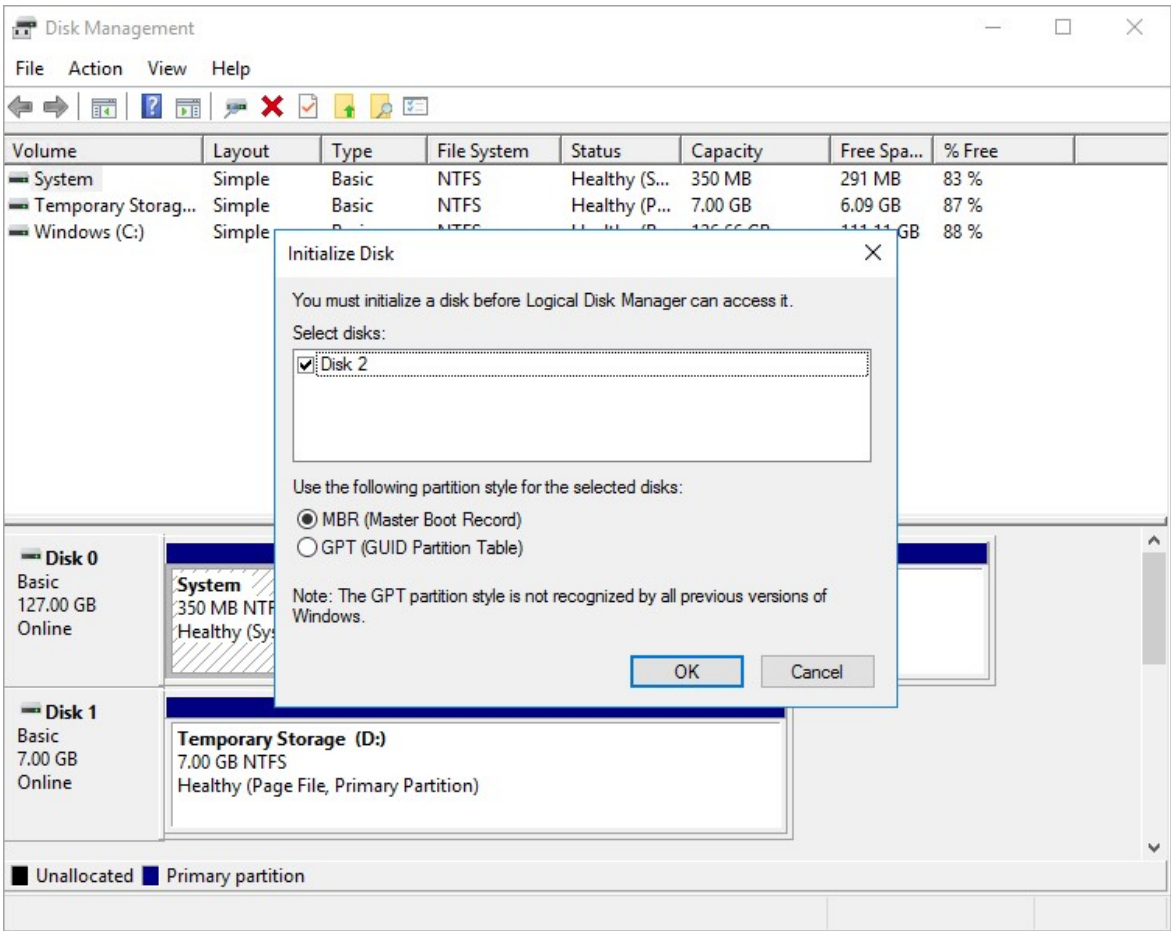
The more interesting thing to observe in the list of drives is what is *missing*. Notice that our **Data** drive is not present. Azure added a VHD but didn't initialize it.

Initialize data disks

Any additional drives you create from scratch will need to be initialized and formatted. The process for doing this is identical to a physical drive.

1. Launch the **Disk Management** tool from the Start Menu. You may have to go to the Computer Management tool first, then Disk Management, or try searching for "Disk Management" in the Start Menu.

2. It will display a warning that it has detected an uninitialized disk.



3. Click **OK** to initialize the disk. It will then show up in the list of volumes where you can format it and assign a drive letter.

4. Open File Explorer and you should now see your data drive.

5. Go ahead and close the RDP client to sign out of the VM. The server will continue to run.

RDP allows you to work with the Azure VM just like a local computer. With Desktop UI access, you can administer this VM as you would any Windows computer: installing software, configuring roles, adjusting features and other common tasks. However, it's a manual process - if we always need to install some software, you might consider automating the process using scripting.

Next unit: Configure Azure virtual machine network settings

Continue >

Configure Azure virtual machine network settings

5 minutes

We've installed our custom software, set up an FTP server, and configured the VM to receive our video files. However, if we try to connect to our public IP address with FTP, we'll find that it's blocked.

Making adjustments to server configuration is commonly performed with equipment in your on-premises environment. In this sense, you can consider Azure VMs to be an extension of that environment. You can make configuration changes, manage networks, open or block traffic, and more through the Azure portal, Azure CLI, or Azure PowerShell tools.

You've already seen some of the basic information and management options in the **Overview** panel for the virtual machine. Let's explore network configuration a bit more.

Opening ports in Azure VMs

By default, new VMs are locked down.

Apps can make outgoing requests, but the only inbound traffic allowed is from the virtual network (e.g. other resources on the same local network), and from Azure's Load Balancer (probe checks).

There are two steps to adjusting the configuration to support FTP. When you create a new VM you have an opportunity to open a few common ports (RDP, HTTP, HTTPS, and SSH). However, if you require other changes to the firewall, you will need to do them yourself.

The process for this involves two steps:

1. Create a Network Security Group.

2. Create an inbound rule allowing traffic on port 20 and 21 for active FTP support.

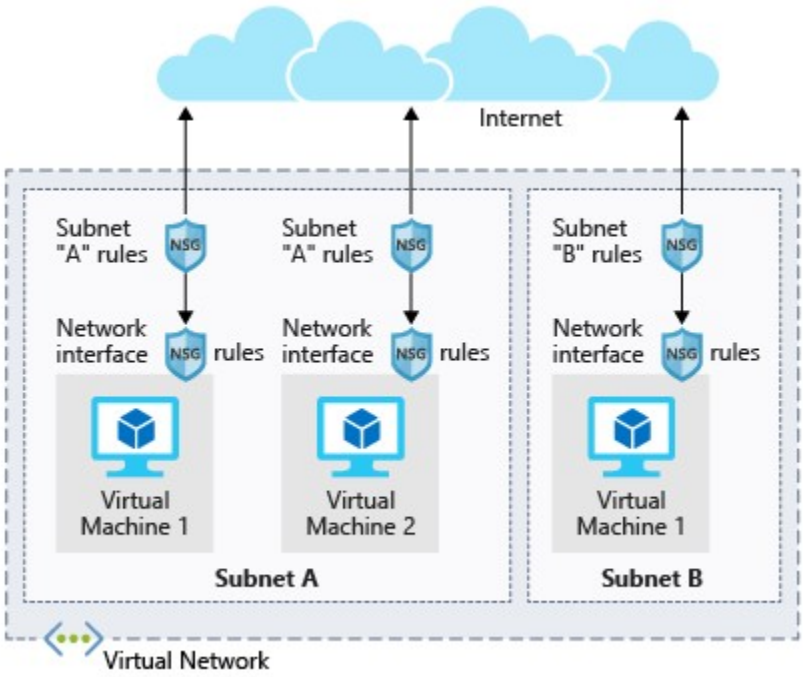
What is a Network Security Group?

Virtual networks (VNets) are the foundation of the Azure networking model and provide isolation and protection. Network Security Groups (NSGs) are the main tool you use to enforce and control network traffic rules at the networking level. NSGs are an optional security layer that provides a software firewall by filtering inbound and outbound traffic on the VNet.

Security groups can be associated to a network interface (for per-host rules), a subnet in the virtual network (to apply to multiple resources), or both levels.

Security group rules

NGSs use *rules* to allow or deny traffic moving through the network. Each rule identifies the source and destination address (or range), protocol, port (or range), direction (inbound or outbound), a numeric priority, and whether to allow or deny the traffic that matches the rule. The following illustration shows NSG rules applied at the subnet and network interface levels.



Each security group has a set of default security rules to apply the default network rules described above. These default rules cannot be modified, but *can* be overridden.

How Azure uses network rules

For inbound traffic, Azure processes the security group associated to the subnet, then the security group applied to the network interface. Outbound traffic is processed in the opposite order (the network interface first, followed by the subnet).

> ⚠ **Warning**
>
> Keep in mind that security groups are optional at both levels. If no security group is applied then **all traffic is allowed** by Azure. If the VM has a public IP, this could be a serious risk particularly if the OS doesn't provide some sort of firewall.

The rules are evaluated in *priority-order*, starting with the **lowest priority** rule. Deny rules always **stop** the evaluation. For example, if an outbound request is blocked by a network interface rule, any rules applied to the subnet will not be checked. In order for traffic to be allowed through the security group, it must pass through *all* applied groups.

The last rule is always a **Deny All** rule. This is a default rule added to every security group for both inbound and outbound traffic with a priority of 65500. That means to have traffic pass through the security group *you must have an allow rule* or it will be blocked by the default final rule.

> ⓘ **Note**
>
> SMTP (port 25) is a special case, depending on your subscription level and when your account was created, outbound SMTP traffic may be blocked. You can make a request to remove this restriction with business justification.

Next unit: Summary

Continue >

Summary

3 minutes

In this module, you learned how to create a Windows VM using the Azure portal. You then connected to the public IP address of the VM and managed it over RDP. You discovered how RDP in Azure provides a similar experience to logging on interactively to a physical computer.

You learned that while RDP allows us to interact with the operating system and software of the virtual machine, the portal allows us to configure the virtual hardware and connectivity. We also could have used PowerShell or the Azure CLI, if a command-line or scriptable environment were preferred.

Clean up

The sandbox automatically cleans up your resources when you're finished with this module.

When you're working in your own subscription, it's a good idea at the end of a project to identify whether you still need the resources you created. Resources left running can cost you money. You can delete resources individually or delete the resource group to delete the entire set of resources.

Check your knowledge

1. When creating a Windows virtual machine in Azure, which port would you open using the INBOUND PORT RULES in order to allow remote-desktop access?

- ○ HTTPS
- ○ SSH (22)
- ◉ RDP (3389) ✓

The Remote Desktop Protocol (RDP) uses port 3389 by default so this port is the standard port you would open if you wanted to use an RDP client to administer your Windows virtual machines.

2. Suppose you have an application running on a Windows virtual machine in Azure. What is the best-practice guidance on where the app should store data files?

- ○ The OS disk (C:)
- ○ The Temporary disk (D:)
- ◉ An attached data disk ✓

Dedicated data disks are generally considered the best place to store application data files. They can be larger than OS disks and you can optimize them for the cost and performance characteristics appropriate for your data.

3. What is the final rule that is applied in every Network Security Group?

- ○ Allow All
- ◉ Deny All ✓

This is a safe choice. It will block all traffic that you don't specifically allow.

- ○ You configure the final rule to your needs

Module complete:

Unlock achievement