# Lab: Deploying compute workloads by using images and containers

## Microsoft Azure user interface

Given the dynamic nature of Microsoft cloud tools, you might experience Azure user interface (UI) changes after the development of this training content. These changes might cause the lab instructions and lab steps to not match up.

Microsoft updates this training course as soon as the community brings needed changes to our attention. However, because cloud updates occur frequently, you might encounter UI changes before this training content is updated. **If this occurs, adapt to the changes and work through them in the labs as needed.**

## Instructions

### Before you start

#### Sign in to the lab virtual machine

* Sign in to your **Windows 10** virtual machine using the following credentials:
  + **Username**: Admin
  + **Password**: Pa55w.rd
* **Note**: Lab virtual machine sign in instructions will be provided to you by your instructor.

#### Review installed applications

* Observe the taskbar located at the bottom of your **Windows 10** desktop. The taskbar contains the icons for the applications you will use in this lab:
  + Microsoft Edge
  + File Explorer

### Exercise 1: Create a virtual machine (VM) by using the Azure portal

#### Task 1: Open the Azure portal

1. On the taskbar, select the **Microsoft Edge** icon.
2. In the open browser window, navigate to the **Azure portal** ([portal.azure.com](https://portal.azure.com)).
3. Enter the **email address** for your Microsoft account.
4. Select **Next**.
5. Enter the **password** for your Microsoft account.
6. Select **Sign in**.

* **Note**: If this is your first time signing in to the **Azure Portal**, a dialog box will appear offering a tour of the portal. Select **Get Started** to begin using the portal.

#### Task 2: Create a resource group

1. On the navigation menu located on the left side of the portal, select the **+ Create a resource** link.

* **Note**: If you cannot find the **Create a resource** link, the “Create a resource” icon is a plus-sign character located on the left side of the portal.

1. At the top of the **New** blade, locate the **Search the Marketplace** text box above the list of featured services.
2. In the search text box, enter the text **Resource Group** and press Enter.
3. In the **Everything** search results blade, select the **Resource group** result.
4. In the **Resource group** blade, select **Create**.
5. In the additional **Resource group** blade, observe the tabs at the top of the blade, such as **Basics**.

* **Note**: Each tab represents a step in the workflow to create a new **resource group**. At any time, you can select **Review + create** to skip the remaining tabs.

1. On the **Basics** tab, perform the following actions:
   1. Leave the **Subscription** text box set to its default value.
   2. In the **Resource group** text box, enter the value **ContainerCompute**.
   3. In the **Region** drop-down list, select the **(US) East US** location.
   4. Select **Review +** **Create**.
2. In the **Review + Create** tab, review the options that you selected during the previous steps.
3. Select **Create** to create the resource group by using your specified configuration.
4. Wait for the creation task to complete before moving forward with this lab.

#### Task 3: Create a Linux virtual machine resource

1. On the navigation menu located on the left side of the portal, select the **+ Create a resource** link.
2. At the top of the **New** blade, locate the **Search the Marketplace** text box above the list of featured services.
3. In the search text box, enter **Ubuntu Server 18** and press Enter.
4. In the **Everything** search results blade, select the **Ubuntu Server 18.04 LTS** result.
5. In the **Ubuntu Server 18.04 LTS** blade, select **Create**.
6. In the **Create a virtual machine** blade, observe the tabs at the top of the blade, such as **Basics** and **Disks**.

* **Note**: Each tab represents a step in the workflow to create a new **virtual machine**. At any time, you can select **Review + create** to skip the remaining tabs.

1. In the **Basics** tab, perform the following actions:
   1. Leave the **Subscription** text box set to its default value.
   2. In the **Resource group** drop-down list, select the existing **ContainerCompute** option.
   3. In the **Virtual machine name** text box, enter **simplevm**.
   4. In the **Region** drop-down list, select the **(US) East US** location.
   5. In the **Availability options** drop-down list, ensure **No infrastructure redundancy required** is selected.
   6. In the **Image** text box, make sure that the **Ubuntu Server 18.04 LTS** option is selected.
   7. In the **Size** text box, select the **Change size** link.
2. In the **Select a VM size** blade, perform the following actions:
   1. Select the **B1s** option from the list of sizes.
   2. Press **Select**.
3. Go back to the **Basics** tab and perform the following actions:
   1. In the **Authentication type** section, select **Password**.
   2. In the **Username** text box, enter **Student**.
   3. In the **Password** and **Confirm password** fields, enter **StudentPa55w.rd**.
   4. In the **Login with Azure Active Directory (Preview)** section, select **Off**.
   5. In the **Public inbound ports** section, select **Allow selected ports**.
   6. In the **Select inbound ports** drop-down list, select only **SSH (22)**.
   7. Select **Next : Disks >**.
4. In the **Disks** tab, perform the following actions:
   1. In the **OS disk type** section, select **Standard SSD**.
   2. Select **Review + create**.
5. In the **Review + Create** tab, review the options that you selected during the previous steps.
6. Select **Create** to create the VM by using your specified configuration.
7. Wait for the creation task to complete before moving forward with this lab.

#### Task 4: Validate the virtual machine

1. On the navigation menu located on the left side of the portal, select the **Resource groups** link.
2. In the **Resource groups** blade, locate and select the **ContainerCompute** resource group that you created earlier in this lab.
3. In the **ContainerCompute** blade, select the **simplevm** VM that you created earlier in this lab.
4. In the **Virtual Machine** blade, select **Connect**.
5. In the **Connect to virtual machine** pop-up that appears, perform the following actions:
   1. In the **IP address** text box, select **Public IP address**.
   2. In the **Port number** text box, enter **22**.
   3. **Copy** the text in the **Login using VM local account** text box.
   * **Note**: The command that you copied will connect to the VM by using SSH from a remote computer. You will use this command later in the lab.
6. At the top of the portal, select the **Cloud Shell** icon to open a new shell instance.

* **Note**: The **Cloud Shell** icon is represented by a greater than symbol and underscore character.

1. If this is your first time opening the **Cloud Shell** by using your subscription, a **Welcome to Azure Cloud Shell Wizard** will appear that allows you to configure **Cloud Shell** for first-time usage. Perform the following actions in the wizard:
   1. A dialog box will appear that prompts you to create a new Storage Account to begin using the shell. Accept the default settings and select **Create storage**.
   2. Wait for the **Cloud Shell** to finish its first-time setup procedures before moving forward with the lab.

* **Note**: If you do not see the configuration options for the **Cloud Shell**, this is most likely because you are using an existing subscription with this course’s labs. The labs are written from the presumption that you are using a new subscription.

1. In the **Cloud Shell** command prompt at the bottom of the portal, **paste** the command you copied earlier in this lab and press Enter to connect to your new VM by using SSH.

* **Note**: This command will be dependent on your username and IP address. For example, if the username is **Student** and the IP address is **40.125.245.5**, the command would be **ssh Student@40.125.245.5**.

1. The SSH tool will first inform you that the authenticity of the host can’t be established and then ask if you want to continue connecting. Enter **yes** in the prompt and then press Enter to continue connecting to the VM.
2. The SSH tool will then ask you for a password. Enter **StudentPa55w.rd** and then press Enter to authenticate with the VM.

* **Note**: Characters do not show when typing password. Please be advised.

1. After you are connected to the VM by using SSH, you will see a prompt for the Bash shell in the VM. In the prompt, type in the following command and press Enter to view the computer name of the Linux VM:

* uname -mnipo

1. In the prompt, type in the following command and press Enter to view information about the distribution and operating system of the Linux VM.

* uname -srv

1. Close the **Cloud Shell** pane at the bottom of the portal.

#### Review

In this exercise, you created a new VM manually by using the Azure portal interface and connected to the VM by using the Cloud Shell and SSH.

### Exercise 2: Create a virtual machine by using Azure CLI

#### Task 1: Open Cloud Shell

1. In the top navigation bar in the Azure Portal, select the **Cloud Shell** icon to open a new shell instance.
2. Wait for the **Cloud Shell** to finish connecting to an instance before moving forward with the lab.
3. In the **Cloud Shell** command prompt at the bottom of the portal, type the following command and press Enter to view the version of the Azure CLI tool:

* az --version

#### Task 2: Use the Azure CLI commands

1. Type the following command and press Enter to view a list of subgroups and commands at the root level of the CLI:

* az --help

1. Type the following command and press Enter to view a list of subgroups and commands for **virtual machines**:

* az vm --help

1. Type the following command and press Enter to view a list of arguments and examples for the **Create Virtual Machine** command:

* az vm create --help

1. Type the following command and press Enter to create a new **virtual machine** with the following settings:
   * **Resource group**: ContainerCompute
   * **Name**: quickvm
   * **Image**: Debian
   * **Username**: Student
   * **Password**: StudentPa55w.rd

* az vm create --resource-group ContainerCompute --name quickvm --image Debian --admin-username student --admin-password StudentPa55w.rd
* **Note**: If this command fails with an error indicating Resource group 'ContainerCompute' could not be found, it likely means you either made an error naming your resource group at the start of this lab, or you have more than one subscription associated with your login name. If you named the resource group wrong, substitute the correct name in the command. If you have more than one subscription, you can use the az account set --subscription command to select the proper subscription.

1. Wait for the VM creation process to complete. After the process completes, the command will return a JSON file containing details about the machine.
2. Type the following command and press Enter to view a more detailed JSON file that contains various metadata about the newly created VM:

* az vm show --resource-group ContainerCompute --name quickvm

1. Type the following command and press Enter to list all the IP addresses associated with the VM:

* az vm list-ip-addresses --resource-group ContainerCompute --name quickvm

1. Type the following command and press Enter to filter the output to only return the first IP address value:

* az vm list-ip-addresses --resource-group ContainerCompute --name quickvm --query '[].{ip:virtualMachine.network.publicIpAddresses[0].ipAddress}' --output tsv

1. Type the following command and press Enter to store the results of the previous command in a new Bash shell variable named *ipAddress*:

* ipAddress=$(az vm list-ip-addresses --resource-group ContainerCompute --name quickvm --query '[].{ip:virtualMachine.network.publicIpAddresses[0].ipAddress}' --output tsv)

1. Type the following command and press Enter to print the value of the Bash shell variable *ipAddress*:

* echo $ipAddress

1. Type the following command and press Enter to connect to the VM that you created earlier in this lab by using the SSH tool and the IP address stored in the Bash shell variable *ipAddress*:

* ssh student@$ipAddress

1. The **SSH** tool will first inform you that the authenticity of the host can’t be established and then ask if you want to continue connecting. Enter **yes** and then press Enter to continue connecting to the VM.
2. The **SSH** tool will then ask you for a password. Enter **StudentPa55w.rd** and then press Enter to authenticate with the VM.
3. After you connect to the VM using SSH, type the following command and press Enter to view metadata describing the Linux VM:

* uname -a

1. Close the **Cloud Shell** pane at the bottom of the portal.

#### Review

In this exercise, you used the Azure Cloud Shell to create a VM as part of an automated script.

### Exercise 3: Create a Docker container image and deploy it to Azure Container Registry

#### Task 1: Open Cloud Shell and editor

1. In the top navigation bar in the Azure Portal, select the **Cloud Shell** icon to open a new shell instance.
2. Wait for the **Cloud Shell** to finish connecting to an instance before moving on with the lab.
3. In the **Cloud Shell** command prompt at the bottom of the portal, type the following command and press **Enter** to move from the root directory to the **~/clouddrive** directory:

* cd ~/clouddrive

1. Type the following command and press Enter to create a new directory named **ipcheck** within the **~/clouddrive** directory:

* mkdir ipcheck

1. Type the following command and press Enter to change the active directory from **~/clouddrive** to **~/clouddrive/ipcheck**:

* cd ~/clouddrive/ipcheck

1. Type the following command and press Enter to create a new .NET Core console application in the current directory:

* dotnet new console --output . --name ipcheck

1. Type the following command and press Enter to create a new file in the **~/clouddrive/ipcheck** directory named **Dockerfile**:

* touch Dockerfile

1. Type the following command and press Enter to open the embedded graphical editor in the context of the current directory:

* code .

#### Task 2: Create and test a .NET Core application

1. Within the graphical editor, locate the **FILES** pane and double-select the **Program.cs** file to open that file in the editor.
2. Delete the entire contents of the **Program.cs** file.
3. Copy and paste the following code into the **Program.cs** file:

* public class Program  
  {  
   public static void Main(string[] args)  
   {   
   if (System.Net.NetworkInformation.NetworkInterface.GetIsNetworkAvailable())  
   {  
   System.Console.WriteLine("Current IP Addresses:");  
   string hostname = System.Net.Dns.GetHostName();  
   System.Net.IPHostEntry host = System.Net.Dns.GetHostEntry(hostname);  
   foreach(System.Net.IPAddress address in host.AddressList)  
   {  
   System.Console.WriteLine($"\t{address}");  
   }  
   }  
   else  
   {  
   System.Console.WriteLine("No Network Connection");  
   }  
   }  
  }

1. **Save** the **Program.cs** file by using either the menu in the the graphical editor, or the **Ctrl+S** keyboard shortcut. Do not close the graphical editor.
2. Back in the command prompt, type the following command and press Enter to execute the application:

* dotnet run

1. Observe the results of the execution. There should be at least one IP address listed for the Cloud Shell instance.
2. Within the graphical editor, locate the **FILES** pane on the left side of the editor and double-select the **Dockerfile** file to open that file in the editor.
3. Copy and paste the following code into the **Dockerfile** file:

* FROM mcr.microsoft.com/dotnet/core/sdk:2.2-alpine AS build  
  WORKDIR /app  
    
  COPY \*.csproj ./  
  RUN dotnet restore  
    
  COPY . ./  
  RUN dotnet publish --configuration Release --output out  
    
  FROM mcr.microsoft.com/dotnet/core/runtime:2.2-alpine  
  WORKDIR /app  
    
  COPY --from=build /app/out .  
    
  ENTRYPOINT ["dotnet", "ipcheck.dll"]

1. **Save** the **Dockerfile** file by using either the menu in the graphical editor or the **Ctrl+S** keyboard shortcut.
2. Close the **Cloud Shell** pane at the bottom of the portal.

#### Task 3: Create an Azure Container Registry resource

1. On the navigation menu located on the left side of the portal, select the **+ Create a resource** link.
2. At the top of the **New** blade, locate the **Search the Marketplace** text box above the list of featured services.
3. In the search text box, enter **Container Registry** and press **Enter**.
4. In the **Everything** search results blade, select the **Container Registry** result.
5. In the **Container Registry** blade, select **Create**.
6. In the **Create container registry** blade, perform the following actions:
   1. In the **Registry name** text box, give your registry a globally unique name.
   * Note: The blade will automatically check the name for uniqueness and inform you if you are required to choose a different name.
   1. Leave the **Subscription** text box set to its default value.
   2. In the **Resource group** drop-down list, select the existing **ContainerCompute** option.
   3. In the **Location** text box, select **East US**.
   4. In the **Admin user** section, select **Disable**.
   5. In the **SKU** drop-down list, select **Basic**.
   6. Select **Create**.
7. Wait for the creation task to complete before moving forward with this lab.

#### Task 4: Open Cloud Shell and store Azure Container Registry metadata

1. At the top of the portal, select the **Cloud Shell** icon to open a new shell instance.
2. Wait for the **Cloud Shell** to finish connecting to an instance before moving forward with the lab.
3. In the **Cloud Shell** command prompt at the bottom of the portal, type the following command and press Enter to view a list of all container registries in your subscription:

* az acr list

1. Type the following command and press Enter:

* az acr list --query "max\_by([], &creationDate).name" --output tsv

1. Type the following command and press Enter:

* acrName=$(az acr list --query "max\_by([], &creationDate).name" --output tsv)

1. Type the following command and press Enter:

* echo $acrName

#### Task 5: Deploy a Docker container image to the Azure Container Registry

1. Type the following command and press Enter to change the active directory from **~/** to **~/clouddrive/ipcheck**:

* cd ~/clouddrive/ipcheck

1. Type the following command and press Enter to view the contents of the current directory:

* dir

1. Type the following command and press Enter to upload the source code to your **Container Registry** and build the container image as an **Azure Container Registry Task**:

* az acr build --registry $acrName --image ipcheck:latest .

1. Wait for the build task to complete before moving forward with this lab.
2. Close the **Cloud Shell** pane at the bottom of the portal.

#### Task 6: Validate your container image in the Azure Container Registry

1. On the navigation menu located on the left side of the portal, select the **Resource groups** link.
2. In the **Resource groups** blade, locate and select the **ContainerCompute** resource group that you created earlier in this lab.
3. In the **ContainerCompute** blade, select the container registry that you created earlier in this lab.
4. In the **Container Registry** blade, locate the **Services** section and select the **Repositories** link.
5. In the **Repositories** section, select the **ipcheck** container image repository.
6. In the **Repository** blade, select the **latest** tag.
7. Observe the metadata for the version of your container image with the **latest** tag.

* **Note**: You can also select the **Run ID** hyperlink to view metadata about the build task.

#### Review

In this exercise, you created a .NET Core console application to display a machine’s current IP address. You then added the Dockerfile file to the application so that it could be converted into a Docker container image. Finally, you deployed the container image to Azure Container Registry.

### Exercise 4: Deploy an Azure container instance

#### Task 1: Enable Admin User in Azure Container Registry

1. On the navigation menu located on the left side of the portal, select the **Resource groups** link.
2. In the **Resource groups** blade, locate and select the **ContainerCompute** resource group that you created earlier in this lab.
3. In the **ContainerCompute** blade, select the container registry that you created earlier in this lab.
4. In the **Container Registry** blade, select **Update** from the top of the blade.
5. In the **Update container registry** blade, perform the following actions:
   1. In the **Admin user** section, select **Enable**.
   2. Select **Save**.
6. Close the **Update container registry** blade.

#### Task 2: Deploy a container image automatically to an Azure Container instance

1. In the **Container Registry** blade, locate the **Services** section and select the **Repositories** link.
2. In the **Repositories** section, select the **ipcheck** container image repository.
3. In the **Repository** blade, select the ellipsis menu located immediately to the right of the **latest** tag entry.
4. In the pop-up menu that appears, select the **Run instance** link.
5. In the **Create container instance** blade that appears, perform the following actions:
   1. In the **Container name** text boxtext box, enter **managedcompute**.
   2. Leave the **Container image** text box set to its default value.
   3. In the **OS type** section, select **Linux**.
   4. Leave the **Subscription** text box set to its default value.
   5. In the **Resource group** drop-down list, select **ContainerCompute**.
   6. In the **Location** drop-down list, select **East US**.
   7. In the **Number of cores** drop-down list, select **2**.
   8. In the **Memory (GB)** text box, enter **4**.
   9. In the **Public IP address** section, select **No**.
   10. Select **OK**.
6. Wait for the creation task to complete before moving forward with this lab.

#### Task 3: Deploy a container image manually to an Azure Container instance

1. On the navigation menu located on the left side of the portal, select the **Resource groups** link.
2. In the **Resource groups** blade, locate and select the **ContainerCompute** resource group that you created earlier in this lab.
3. In the **ContainerCompute** blade, select the container registry that you created earlier in this lab.
4. In the **Container Registry** blade, locate the **Settings** section and select the **Access keys** link.
5. In the **Access Keys** section, record the values for the following fields:
   1. **Login server**
   2. **Username**
   3. **Password**
   * **Note**: You will use these values later in this lab when you create another **container instance**.
6. On the navigation menu located on the left side of the portal, select the **+ Create a resource** link.
7. At the top of the **New** blade, locate the **Search the Marketplace** text box above the list of featured services.
8. In the search text box, enter **Container instance** and press **Enter**.
9. In the **Everything** search results blade, select the **Container Instances** result.
10. In the **Container Instances** blade, select **Create**.
11. In the **Create Container Instances** blade, observe the tabs on the left of the blade, such as **Basics** and **Advanced**.

* **Note**: Each tab represents a step in the workflow to create a new **container instance**.

1. In the **Basics** tab, perform the following actions:
   1. Leave the **Subscription** text box set to its default value.
   2. In the **Resource group** drop-down list, select **ContainerCompute**.
   3. In the **Container name** text box, enter **manualcompute**.
   4. In the **Region** drop-down list, select **(US) East US**.
   5. In the **Image type** section, select **Private**.
   6. In the **Image name** text box, enter the **Login server** value that you recorded earlier and then add the suffix /**ipcheck:latest**.
   * **Note**: For example, if your **Login server** value is **azadmin.azurecr.io**, then your container image name would be **azadmin.azurecr.io/ipcheck:latest**
   1. In the **Image registry login server** text box, enter the **Login server** value that you recorded earlier in this lab.
   2. In the **Image registry user name** text box, enter the **Username** value that you recorded earlier in this lab.
   3. In the **Image registry password** text box, enter the **Password** value that you recorded earlier in this lab.
   4. In the **OS type** section, select **Linux**.
   5. In the **Size** section, select the **Change size** link.
   6. In the **Change container size** blade, perform the following actions:
      1. In the **Number of CPU cores** textbox, enter **1**.
      2. In the **Memory (GiB)** textbox, enter **1.5**.
      3. Leave the default value for the **GPU type** drop-down list.
      4. Select **Ok**.
   7. Select **Next: Networking**
2. In the **Networking** tab, perform the following actions:
   1. In the **Include Public IP address** section, select **Yes**.
   2. Ensure in the **Ports** section, the port **80** is there, with the port protocol **TCP**.
   3. Leave the **DNS name label** text box empty.
   4. Select **Next: Advanced**.
3. In the **Advanced** tab, perform the following actions:
   1. In the **Restart policy** drop-down list, select **On failure**.
   2. Leave the **Environment variable** text box empty.
   3. Leave the **Command override** text box empty.
   4. Select **Review + create**.
4. In the **Review + create** tab, review the selected options.
5. Select **Create** to create the container instance by using your specified configuration.
6. Wait for the creation task to complete before moving forward with this lab.

#### Task 4: Validate that the container instance ran successfully

1. On the navigation menu located on the left side of the portal, select the **Resource groups** link.
2. In the **Resource groups** blade, locate and select the **ContainerCompute** resource group that you created earlier in this lab.
3. In the **ContainerCompute** blade, select the **manualcompute** container instance that you created earlier in this lab.
4. In the **Container Instance** blade, locate the **Settings** section and select the **Containers** link.
5. In the **Containers** section, observe the list of **Events**.
6. Select the **Logs** tab and observe the text logs from the container instance. > **Note**: You can also optionally view the **Events** and **Logs** from the **managedcompute** container instance.

**Note**: After the application finished executing, the container was terminated because it had completed its work. For the manually created container instance, you indicated that you indicated that a successful exit was acceptable, so the container shows a single execution run. The automatically created instance did not offer you this option, and assumes the container should always be running, so you will see repeated restarts of the container.

#### Review

In this exercise, you used multiple methods to deploy a container image to an Azure container instance. By using the manual method, you were also able to customize the deployment further and execute task-based applications as part of a container run.

### Exercise 5: Clean up subscription

#### Task 1: Open Cloud Shell and list resource groups

1. At the top of the portal, select the **Cloud Shell** icon to open a new shell instance.
2. In the **Cloud Shell** command prompt at the bottom of the portal, type in the following command and press Enter to list all resource groups in the subscription:

* az group list

1. Type the following command and press Enter to view a list of possible commands to delete a resource group:

* az group delete --help

#### Task 2: Delete resource groups

1. Type the following command and press Enter to delete the **ContainerCompute** resource group:

* az group delete --name ContainerCompute --no-wait --yes

1. Close the **Cloud Shell** pane at the bottom of the portal.

#### Task 3: Close active applications

1. Close the currently running **Microsoft Edge** application.

#### Review

In this exercise, you cleaned up your subscription by removing the **resource groups** used in this lab.