**Module 3: Integrating with Azure Compute Services**

**Lab B: Moving containers between on-premises Hyper-V virtual machines and Azure virtual machines**

**Exercise 1: Creating a Docker host by using Docker Machine**

**Task 1: Install Docker Toolbox for Windows**

1. On the host computer, start Internet Explorer and browse to <https://docs.docker.com/docker-for-windows/install>
2. On the **Install Docker for Windows** page, click **Download from Docker Store**.
3. On the **Docker Community Edition for Windows** page, click **Please Login to Download**. If you do not have a **Docker account**, click **Create account**. On the **Welcome to Docker Store** page, create a new Docker ID. Note that you will need to provide a valid email address. Once you receive the verification email, click the link included in the email and click **Please Login to Download** again.
4. Log in using your Docker ID.
5. On the **Docker Community Edition for Windows** page, click **Get Docker**.
6. When prompted whether to run or save **Docker for Windows Installer.exe**, click **Run**.
7. In the **Installing Docker for Windows** window, accept the default setings and click **OK**
8. Wait for the installation to complete and, in the **Installation succeeded** window, click **Close**.
9. In the Internet Explorer, browse to <https://github.com/docker/machine/releases/tag/v0.15.0>
10. Click **docker-machine-Windows-x86\_64.exe** and, when prompted, click **Save**.
11. Once the download completes, start File Explorer, browse to the Downloads folder, right-click **docker-machine-Windows-x86\_64.exe** and click **Properties**.
12. In the **Properties** dialog box, enable the **Unblock** checkbox and then click **OK**.
13. In File Explorer, rename **docker-machine-Windows-x86\_64.exe** to **docker-machine.exe**.
14. In File Explorer, copy **docker-machine.exe** to **C:\Program Files\Docker\Docker\resources\bin**, overwriting the existing file.

**Task 2: Create a Docker host in a Hyper-V virtual machine by using Docker Machine**

1. On the host computer, close the **Welcome Docker is now up and running** window.
2. On the host computer, click **Start**, in the Start menu, expand the **Windows System** folder, right-click **Command Prompt**, in the right-click menu, click **More** and then click **Run as administrator**.
3. In the **Administrator: Command Prompt** window, type the following command, and then press Enter.

docker-machine create --driver hyperv --hyperv-virtual-switch "Microsoft Hyper-V Network Adapter - Virtual Switch" localdockervm

1. Wait for the new Microsoft Hyper-V virtual machine to be provisioned. This might take a few minutes. If the boot2docker ISO image is out-of-date, Docker Machine will initiate download of the latest version.

**Task 3: Create a Docker host in an Azure virtual machine by using Docker Machine**

1. On the host computer, in the **Administrator: Command Prompt** window, type the following command, and then press Enter:

az cloud set --name AzureUSGovernment

az login

**Note:** New procedure—you’ll just get a Windows Security Sign in screen.

1. In the sign-in windows that appears, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
2. Note the message stating that you have signed in to the Microsoft Azure Cross-platform Command Line Interface application on your device. Close the Internet Explorer window.
3. If you have multiple subscriptions associated with your Microsoft account, to identify the subscription in which you are going to create a new Azure VM, in the **Administrator: Command Prompt** window, type the following command, and then press Enter:

az account show

1. Note the value of the id property for each subscription in the output of the previous command. To specify the subscription in which you are going to create a virtual network, in the **Administrator: Command Prompt** window, type the following command and then press Enter (replace <subscription\_id> with the value of the id property of that subscription):

az account set --subscription "<subscription\_id>"

1. To identify Ubuntu images that you can use in this lab, in the **Administrator: Command Prompt** window, type the following command, and then press Enter (make sure to replace <location> with the name of an Azure region where you provisioned an Azure VM in the first lab of this module):

az vm image list-skus --location "<location>" --publisher Canonical --offer UbuntuServer --output table

1. Identify the values appearing in the **Name** column in the output. Verify that **16.04.0-LTS** is available. If that is not the case, then, in step 12, replace **16.04.0-LTS** with the name of one of an available images.
2. To create a new resource group, in the **Administrator: Command Prompt** window, type the following command and press Enter (make sure to replace <location> with the name of the Azure region you used in step 8):

az group create --name wingovm3bXX--LabRG --location "USGov Virginia"

1. To change the current directory to the local user profile, in the **Administrator: Command Prompt** window, type the following, and then press Enter :

cd %USERPROFILE%

1. To provision a Docker host in an Azure VM, in the **Administrator: Command Prompt** window, type the following (replacing <subscription\_id> with the value of your Azure subscription id you specified earlier in this task, replacing the <location> with the name of the Azure region where you provisioned the Azure VM in the first lab of this module, and <vm\_size> with the size of that Azure VM), and then press Enter (make sure to use the lower case characters when specifying the virtual machine name):

docker-machine create --driver azure --azure-ssh-user student --azure-subscription-id "<subscrption\_id>" --azure-open-port 80 --azure-open-port 8080 --azure-image "Canonical:UbuntuServer:16.04.0-LTS:latest" --azure-location "<location>" --azure-resource-group "wingovm3bXX-LabRG" --azure-size "<vm\_size>" --azure-static-public-ip "wingovm3bXX-vm1"

1. Note: New procedure—you’ll just get a Windows Security Sign in screen.  ~~will be presented with the message instructing you to open a browser at the page~~ [~~https://aka.ms/devicelogin~~](https://aka.ms/devicelogin) ~~and provide the code included in the message to authenticate. Start Internet Explorer and browse to~~ [~~https://aka.ms/devicelogin~~](https://aka.ms/devicelogin)~~.~~
2. ~~On the~~ **~~Device Login~~** ~~page, type the code included in the message. This will identify Docker Machine for Azure as the application attempting to authenticate. Click~~ **~~Continue~~**~~.~~
3. In the sign-in windows that appears, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
4. Note the message stating that the Docker Machine for Azure would like to sign you in and read your profile and access Azure Service Management as you. Click **Accept**.
5. Note the message stating that you signed in to the Docker Machine for Azure application on your device. Close the Internet Explorer window.
6. In the **Administrator: Command Prompt** window, monitor the progress of provisioning the Azure VM.
7. Wait for the new Docker host Azure VM to be provisioned.
8. in the **Administrator: Command Prompt** window, type the following command, and press then Enter.

docker-machine ls

1. Verify that the output includes both the Hyper-V virtual machine and the Azure VM.
2. in the **Administrator: Command Prompt** window, type the following command, and then press Enter.

docker-machine env wingovm3bXX-vm1

1. Review the output returned by the **docker-machine env** command.
2. To configure the local shell environment, in the **Administrator: Command Prompt** window, type the following command, and then press Enter.

@FOR /f "tokens=\*" %i IN ('docker-machine env wingovm4XX-vm1') DO @%i

**Task 4: Run a container in a Docker host running in a Hyper-V virtual machine**

1. On **the host computer**, click **Start** and then click **Windows Administrative Tools**.
2. In the **Administrative Tools** window, click **Hyper-V Manager**.
3. In the **Hyper-V Manager** window, in the list of virtual machines, right-click **localdockervm**, and then click **Connect**.
4. At the command prompt in the console of the **localdockervm** virtual machine, type the following command, and then press Enter:

docker run -d -p 80:80 --restart=always nginx

1. Wait until the container for nginx is downloaded and started on the Docker host virtual machine. This might take a few minutes depending on the available bandwidth.

**Note:** The ability to download the nginx image and connect to the nginx-based container depends on the availability of DHCP in the lab environment.

1. To obtain the IP address of the target virtual machine hosting the containerized application, type the following command at the command prompt in the console session to **localdockervm**, and then press Enter.

ifconfig eth0

1. Review the second line of the output and note the IP address following the **inet addr:** string.
2. On the host computer, start Internet Explorer, and then browse to the IP address you obtained in the previous step. Verify that Internet Explorer displays the **Welcome to nginx** page.

**Task 5: Run a container in a Docker host running in an Azure virtual machine**

1. On the host computer, in the **Administrator: Command Prompt** window, type the following command, and then press Enter.

docker run -d -p 80:80 --restart=always nginx

1. To verify the successful outcome, in the **Administrator: Command Prompt** window, type the following command, and then press Enter.

docker ps

1. To obtain the IP address of the target Azure VM hosting the containerized application, in the **Administrator: Command Prompt** window, type the following command, and then press Enter.

docker-machine ip wingovm3bXX-vm1

1. On the host computer, start Internet Explorer, and then browse to the IP address you obtained in the previous step. Verify that Internet Explorer displays the **Welcome to nginx** page.
2. Leave the Internet Explorer window open.

**Result**: After you completed this exercise, you should have successfully installed Docker Toolbox for Windows, created a Docker host in a Hyper-V virtual machine by using Docker Machine, created a Docker host in an Azure virtual machine by using Docker Machine, and run a sample containerized web server nginx on both Docker host virtual machines.

**Exercise 2: Deploying a private Docker Registry in Azure**

**Task 1: Create an Azure Container Registry**

1. On the host computer, start Internet Explorer.
2. In Internet Explorer, browse to the Azure portal at <http://portal.azure.us>
3. When prompted, sign in by using the Microsoft account that is the Service Administrator of your Azure subscription.
4. In the Azure portal, in the hub menu on the left hand side, click **+ Create a resource**.
5. On the **New** blade, click **Containers**, and then click **Azure Container Registry**.
6. On the **Create container registry** blade, specify the following settings and click **Create**:
   1. Registry name: a unique name consisting of between 5 and 50 alphanumeric characters
   2. Subscription: the name of the Azure subscription you are using in this lab
   3. Resource group: click **Use existing** and, in the drop-down box, select **wingovm3bXX-LabRG**
   4. Location: select any Azure location where you can create an Azure Container Registry, preferably the same one hosting the Azure VM you deployed earlier in this exercise
   5. Admin user: **Enable** (this allows you to use the registry name as username and admin user access key as password to access the registry)
   6. SKU: **Basic**
7. Wait for the operation to complete.

**Task 2: Identify Azure Container Registry authentication settings.**

1. On the host computer, in the Azure portal, click **All services**, in the **Filter** text box, type **Container registries**, and then, in the service menu, click **Container registries**.
2. On the **Container registries** blade, click the Azure container registry you created in the previous task.
3. On the container registry blade, click **Access keys**.
4. Click the **Click to copy** icon next to the **password** entry. If prompted whether to allow the webpage to acces your Clipboard, click **Allow access**.
5. Note the values of the **Username** and the **Login server** entries. The username should match the registry name and the login server name should consist of the registry name followed by the **.azurecr.io** suffix.

**Task 3: Push an image to Azure Container Registry.**

1. On the host computer, in the **Administrator: Command Prompt** window, to log in to the Azure Container registry you created in the first task, type the following, replacing the <user-name>, <password>, and <login-server> entries with the values you identified in the previous task, and then press Enter:

docker login --username <user-name> --password <password> <login-server>

1. Ensure that you receive the **Login succeeded** message. Next, to pull an existing image from Docker Hub to the Azure Docker VM, in the **Administrator: Command Prompt** window, type the following, and then press Enter:

docker pull microsoft/aci-helloworld

1. Wait for the image to be downloaded to the Docker Azure VM. Next, to tag the image with the Azure Container registry name, in the **Administrator: Command Prompt** window, type the following, replacing the <login-server> entry with the value you identified in the previous task, and then press Enter:

docker tag microsoft/aci-helloworld <login-server>/aci-helloworld:v1

1. To push the tagged image to the Azure Container registry, in the **Administrator: Command Prompt** window, type the following, replacing the <login-server> entry with the value you identified in the previous task, and then press Enter:

docker push <login-server>/aci-helloworld:v1

1. Wait for the image to be pushed to the registry. Next, to view the images stored in the Azure Container registry, switch to the Internet Explorer window displaying the Azure portal, on the container registry blade, click **Repositories** and note that the list includes the **aci-helloworld** repository.

**Task 4: Download and deploy images from the Azure Container Registry**

1. On the host computer, in the **Administrator: Command Prompt** window, to pull an image from the Azure Container registry, type the following, replacing the <login-server> entry with the value you identified earlier in this exercise, and then press Enter:

docker pull <login-server>/aci-helloworld:v1

1. Note that, in this case, the image does not need to be downloaded, since its up-to-date version is already present on the target Docker Azure VM.
2. Next, to deploy the image downloaded from the Azure Container registry, in the **Administrator: Command Prompt** window, type the following, replacing the <login-server> entry with the value you identified earlier in this exercise, and then press Enter:

docker run -d --restart=always -p 8080:80 <login-server>/aci-helloworld:v1

1. To verify that the image has been successfully deployed, in the **Administrator: Command Prompt** window, type the following, and then press Enter:

docker ps

1. Note that the output includes the aci-helloworld:v1 image.
2. To access the running container, switch to the Internet Explorer window displaying the **Welcome to nginx!** page, append **:8080** directly after the IP address of the Azure VM appearing in the address bar, and then press Enter. Verify that you see the **Welcome to Azure Container Instances** page.

**Task 5: Identify and delete all lab Azure resources**

1. In the Azure portal, in the hub menu, click **Resource groups**.
2. On the **Resource groups** blade, click **wingovm3bXX-LabRG**.
3. On the **wingovm3bXX-LabRG** blade, click **Delete**.
4. On the **Are you sure you want to delete "wingovm3bXX-LabRG"** blade, in the **TYPE THE RESOURCE GROUP NAME** box, type the name of the resource group, and then click **Delete**.
5. Close all open windows.

**Result**: After you completed this exercise, you should have successfully deployed a private Docker registry in Azure.