# Overview

The purpose of this guided practice is to familiarize you with container automation using the docker build command and a Dockerfile.

# Objectives

* Create containers.
* Manage containers with docker.
* Manage apps in containers.

## Skills Reviewed

* Creating, running, using, and deleting containers.
* Creating folders.
* Downloading and installing applications.

## New Skills

* Creating an app with a Dockerfile.
* Using the docker build image.
* Viewing a docker image history.

## References

* Dockerfile on Windows - <https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-docker/manage-windows-dockerfile>
* Dockerfile reference - <https://docs.docker.com/engine/reference/builder/>
* Docker build command reference - <https://docs.docker.com/engine/reference/commandline/build/>

# Initial Conditions

Your virtual machine should be in this state prior to beginning this guided practice:

* Docker Enterprise installed on a Windows Server 2019 virtual machine.
* The latest Windows Server Core docker images have been download and ready for use.

# Final Conditions

At the end of this exercise, you will have:

* A docker build file that creates a web server with a custom site.
* A docker container of an IIS web server with a custom web site.

# Instructions

A Dockerfile is a text file that can be used to build a container. It contains a description that includes the image to be used, any commands to be run, ports to publish, the application to run and other settings and metadata.

In this guided practice, you will use a Dockerfile to automate the build of a container with a web site.

## Setup

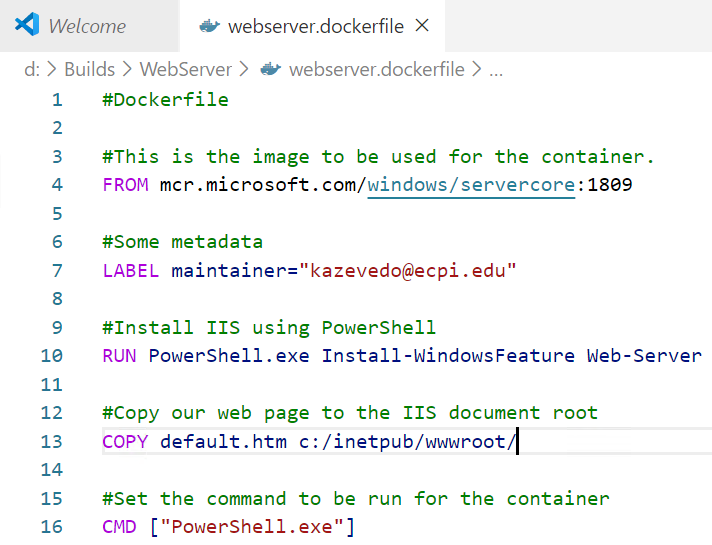
* Remove any containers that might still exist on your system.
* Create a folder named **Builds** in the root of the **D:\** drive.
* Create a folder named **WebServer** in the **Builds** folder.
* Download and install **Visual Studio Code** on the lab host machine. You can download it from here. <https://code.visualstudio.com/Download>

## Creating a Dockerfile

When building containers, the commands in the Dockerfile will use absolute references to files or relative references. Creating a folder and saving all the files to this location can make referencing the files inside the Dockerfile easier. For this reason, you should save all the files you create to the **D:\Builds\WebServer** folder.

A Dockerfile is a text file and so it can be created using the text editor of your choice. In this example, you will use the Visual Studio Code.

To create a docker file to create a container that uses the Windows Server Core base image, installs IIS, and creates a simple web page, perform the following:

1. Login to the Host virtual machine.
2. Copy the **default.htm** file from the **D:\WebData** **to** the **D:\Builds\WebServer,** which you will need to create.
3. Downloadandinstall **Visual Studio Code.**
4. Open **Visual Studio Code**, and type the text shown in the image below. Change the maintainer email address to your own.
5. Save the file in the **D:\Builds\WebServer** folder. Name the file **WebServer.dockerfile**.
6. The entries in the Dockerfile have the following function:
   1. **FROM** 🡪 specifies the image to use
   2. **LABEL** 🡪 adds a label named maintainer to the image and sets the value to [kazevedo@ecpi.edu](mailto:kazevedo@ecpi.edu)
   3. **RUN** 🡪 runs a command in the container. This command installs the Web Server role
   4. **COPY** 🡪 copies the default.htm file from the working directory to the c:\inetpub\wwwroot directory. Note that you must use forward slashes for the path separator.
   5. **CMD** 🡪 shows the command to be run in the container.

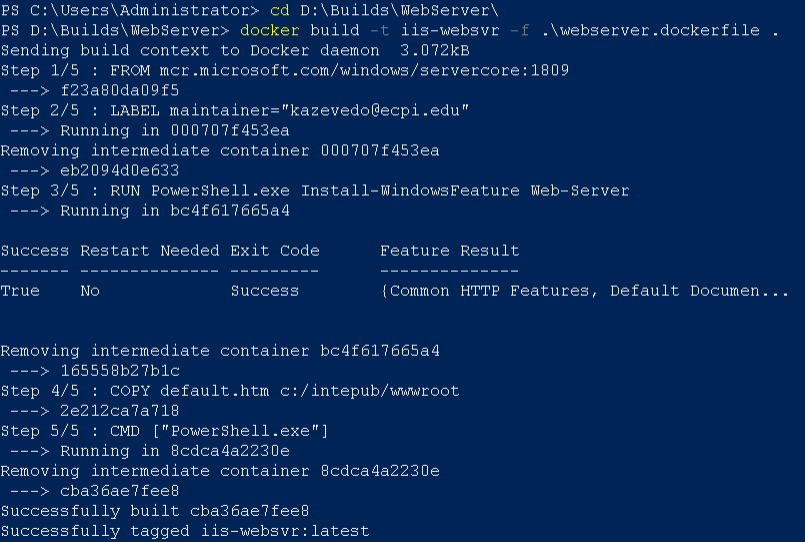
## Building an application with a Dockerfile

Once you have created the Dockerfile, you can now build your application.

To build an application using a Dockerfile, perform the following:

1. Open an **elevated** **PowerShell** session.
2. Change the directory to the **D:\Builds\WebServer** directory.
3. Type the following command:

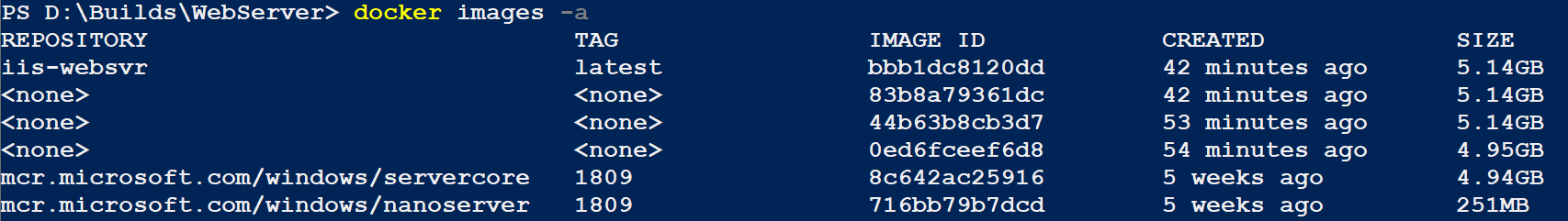
docker build -t iis-websvr -f webserver.dockerfile .

1. You should see the output like the figure below.
2. There are a few things to note in the output and the command usage:
   1. If you name the file Dockerfile, you do not have to specify a file name with the -f parameter.
3. You can verify that the image was created, using the following command:

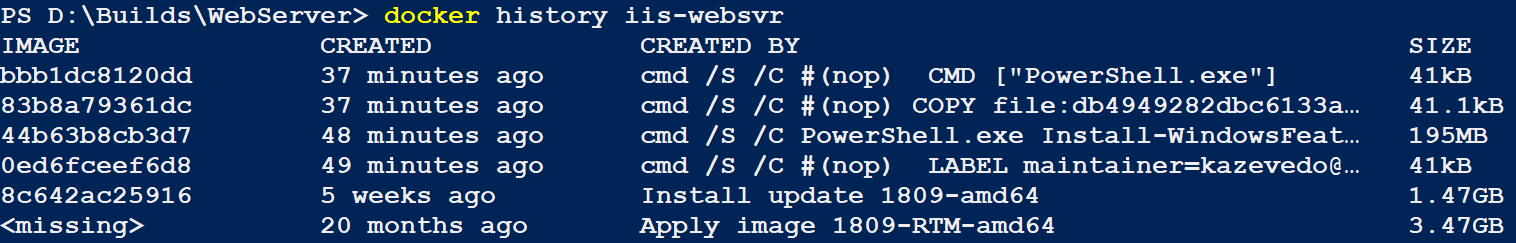
docker images

1. You should output like the image below.
2. To see all the images on your system, type the following command:

docker images -a

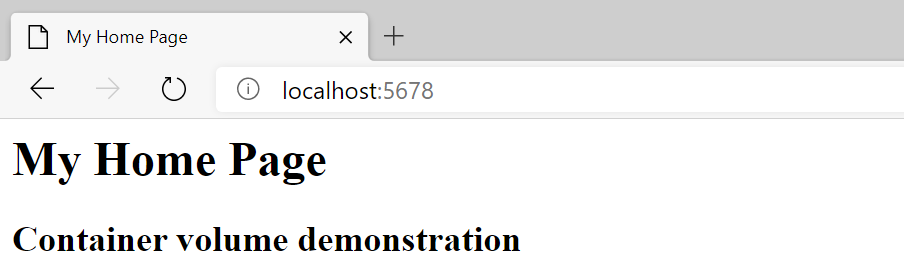
1. Your output should look like the image below.
2. Note the extra images. These were created during the build process.
3. You can verify this by looking at the **iis-websvr** image with the following command:

docker history iis-websvr

1. You should output like the image below.
2. Notice that the image IDs match the images with the <none> repository. These are intermediate images and are needed for the final image. These work like differencing disks and each image only consists of the added files. If you do not want these images to be present, you can specify the **--rm** option when building the image.

## Testing a Dockerfile

To test the image and Dockerfile you need use the image to create a container and test its operation. To do this perform the following:

1. Open an **elevated** **PowerShell** session.
2. **Create** a container using the following settings:
   1. **Name:** DF-Test
   2. **Image:** iis-websvr
   3. **Command:** PowerShell
   4. **Options**
      1. Publish port 80 to port 5678
      2. Interactive
      3. TTY
      4. The container removes itself when it stops
3. Browse to the website using a web browser. You should see the output shown in the figure.

## Modifying a Dockerfile

One of the great features of Dockerfiles is that they allow you to quickly make changes to an image and redeploy the container.

You have decided that you want your web site to show your network documentation instead of the previous page. To make these changes, perform the following:

1. Downloadthe **netdocs.zip** filefrom the assignment page and extractthefilestothe **D:\Builds\WebServer** folder.
2. Delete the **default.htm** file in the **D:\Builds\WebServer** folder.
3. Rename the **netdocs.htm** file **to** **default.htm**
4. Open the **webserver.dockerfile** and add the following line after the **COPY** **default.htm**…command
5. **COPY netdocs\_files c:/inetpub/wwwroot/**
6. The changed section in the Dockerfile should look like the image below.
7. Save your **webserver.docker** file.

## Redeploying a Dockerfile

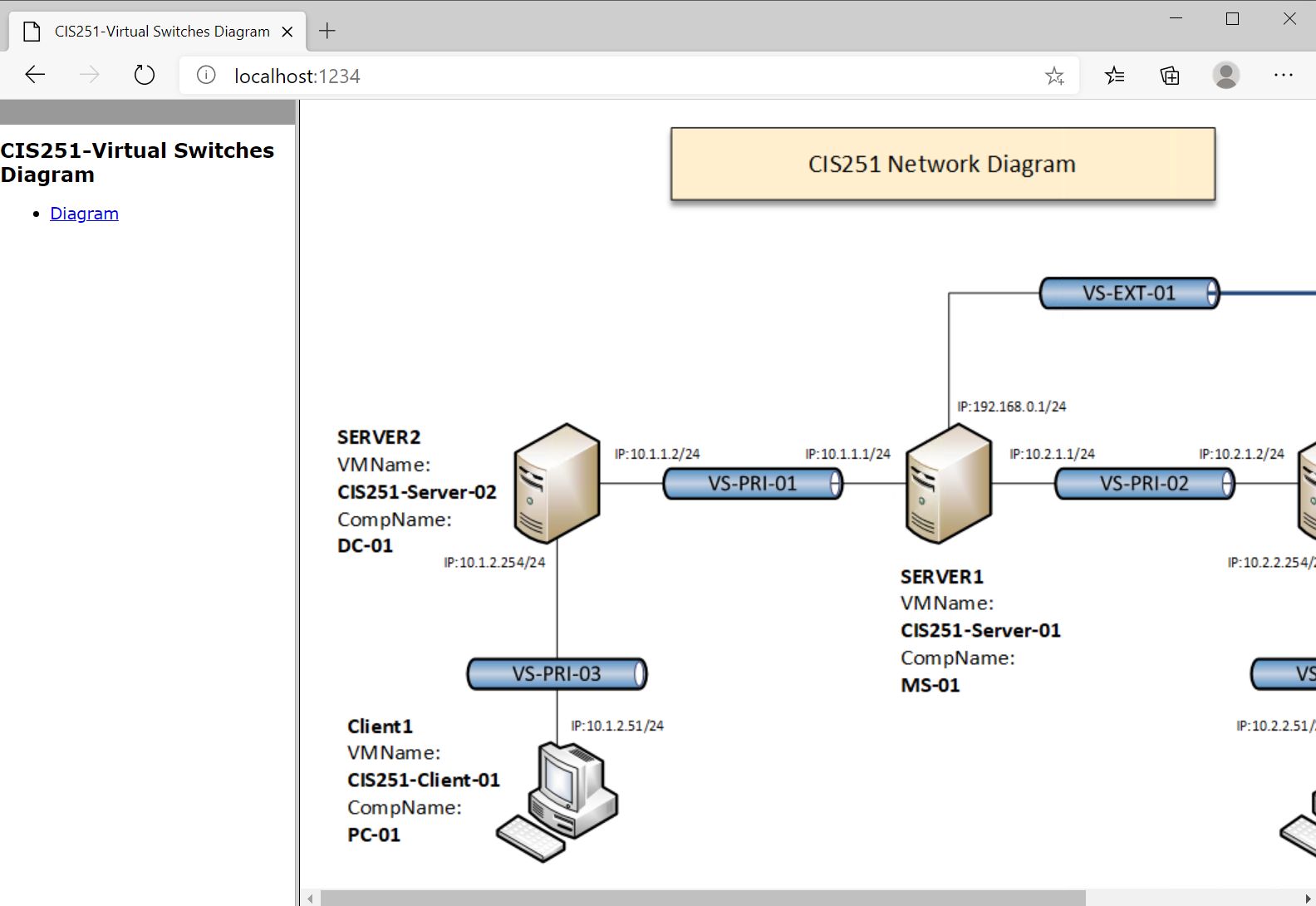
You are now ready to redeploy your web application. To rebuild your image, type the following command in the WebServer directory:

docker build -t iis-websvr:v2 -f .\webserver.dockerfile .

The process should be very quick, and you should now have a new docker image named iis-websvr:v2 as shown below.

Create a new container with the following settings:

* **Name**: V2-Test
* Uses the image you just created
* **Command**: PowerShell
* **Options**
  + Publish port 80 to port 1234
  + Interactive
  + TTY

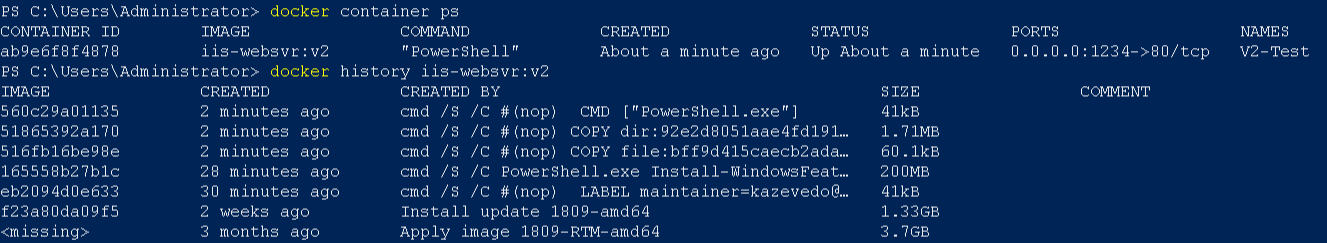
Your website should look something like the screen below. **Leave** your **container** **running** until you run the commands in the submission requirements section.

# Submission Requirements

1. **Open** a **new Word document** and **paste** a **screenshot** **of** the **output** of the following commands:

***docker container ps***

***docker history iis-websvr:v2***

1. The output of the command should look like the image below.
2. Paste a screen shot of your complete docker file.
3. **Fill** **in** the **information** in the following table. Copy the following table to the document and fill in the information about all the **new** commands used in this lab (the example provided is not a new command and should be deleted):

|  |  |  |
| --- | --- | --- |
| Docker Build Commands | | |
| Command | Example | Description |
| docker run | docker run -it --rm iis-core-template PowerShell | Creates a container using the iis-core-template …. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Upload** the **document** in the submission area of the assignment.