Deploying a Web Server using Chef: Module 7, Lesson 4   
Getting Started with Chef Hands-On Lab

## Overview

In this lab, you will deploy an Apache webserver on a CentOS virtual machine and an Internet Information Server (IIS) on a Windows virtual machine.

## Objectives

In this hands-on lab you will learn how to:

* Launch a virtual CentOS 7.x and Windows Server 2012 R2 Datacenter machine
* Install the Chef Development Kit on each VM
* Create a web server cookbook and recipe to install, configure and start a web server on each VM
* Verify that the web server is running from the command line and from a web browser

## Prerequisites

The following are required to complete this hands-on lab:

* A Microsoft Azure subscription
* A web browser
* A terminal application (if you are using a Mac or Linux workstation) or PuTTY (for a Windows workstation) to connect to the Linux VM
* An RDP client to connect to the Windows VM, such as Remote Desktop Connection from a Windows machine or Microsoft Remote Desktop for a Mac.

**Note:** The Azure portal is continually improved and changed. The steps in this exercise reflect the user interface of the Microsoft Azure portal at the time of writing, but may not match the latest design of portal.

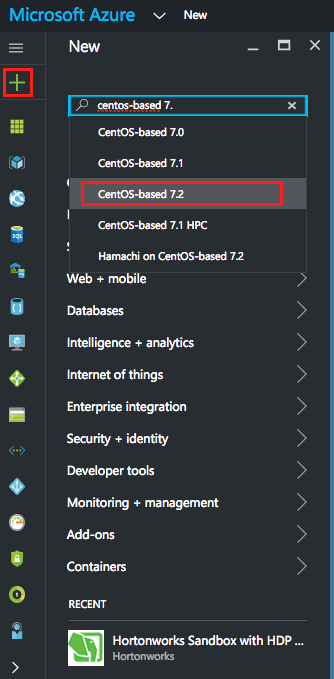
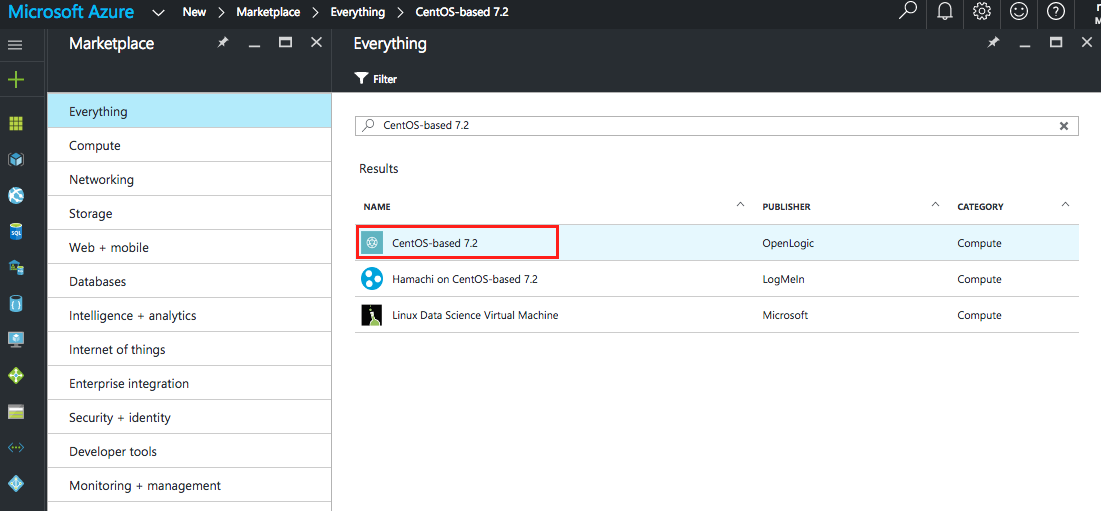
## Exercises

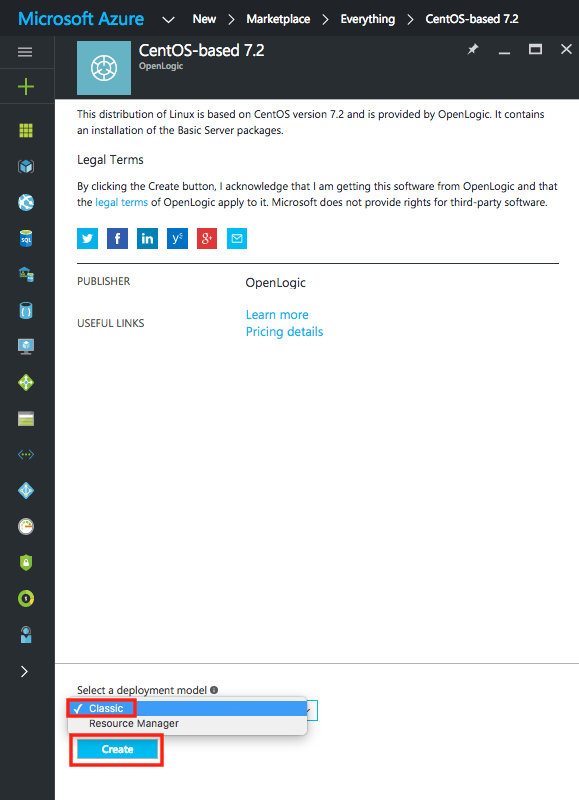
This hands-on lab includes the following exercises:

* Exercise 1: Launch a CentOS 7.x VM
* Exercise 2: Launch a Windows Server 2012 R2 Datacenter VM
* Exercise 3: Install the Chef Development Kit (ChefDK) on each VM
* Exercise 4: Create a cookbook and recipe for CentOS, converge the node and verify your functioning web server
* Exercise 5: Create a cookbook and recipe for Windows Server 2012, converge the node and verify your functioning web server

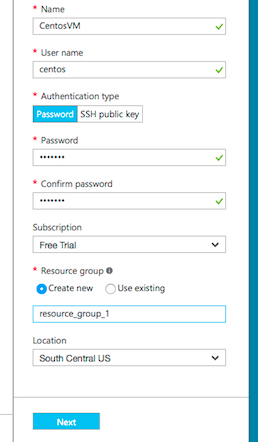
## Exercise 1: Launch a CentOS 7.x VM

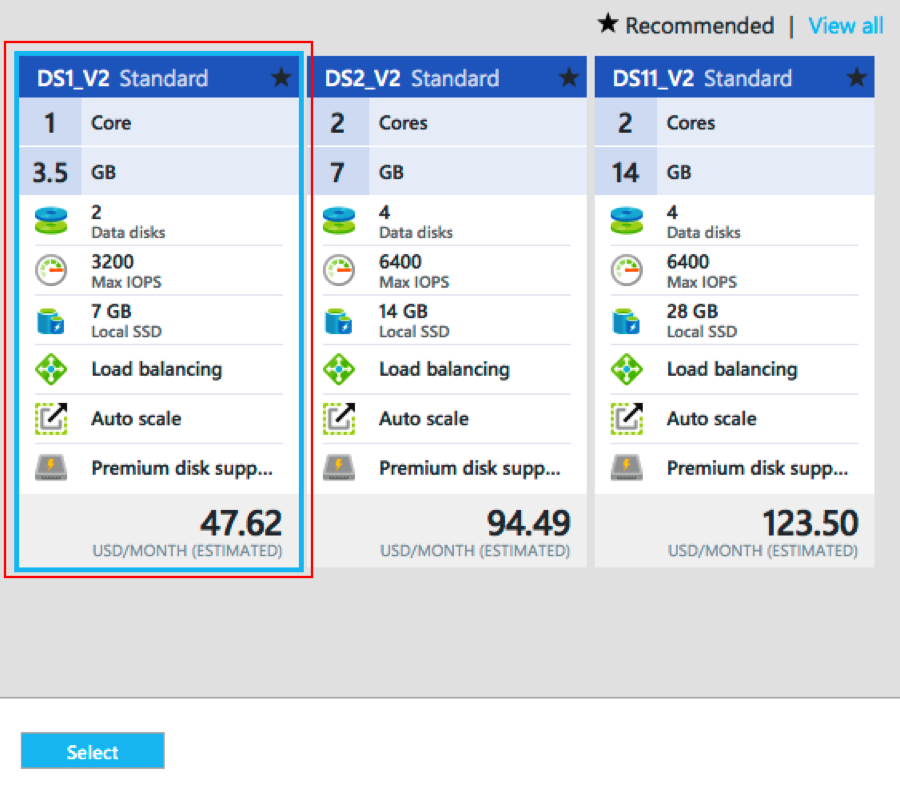
The first task you have to perform is to launch virtual machines

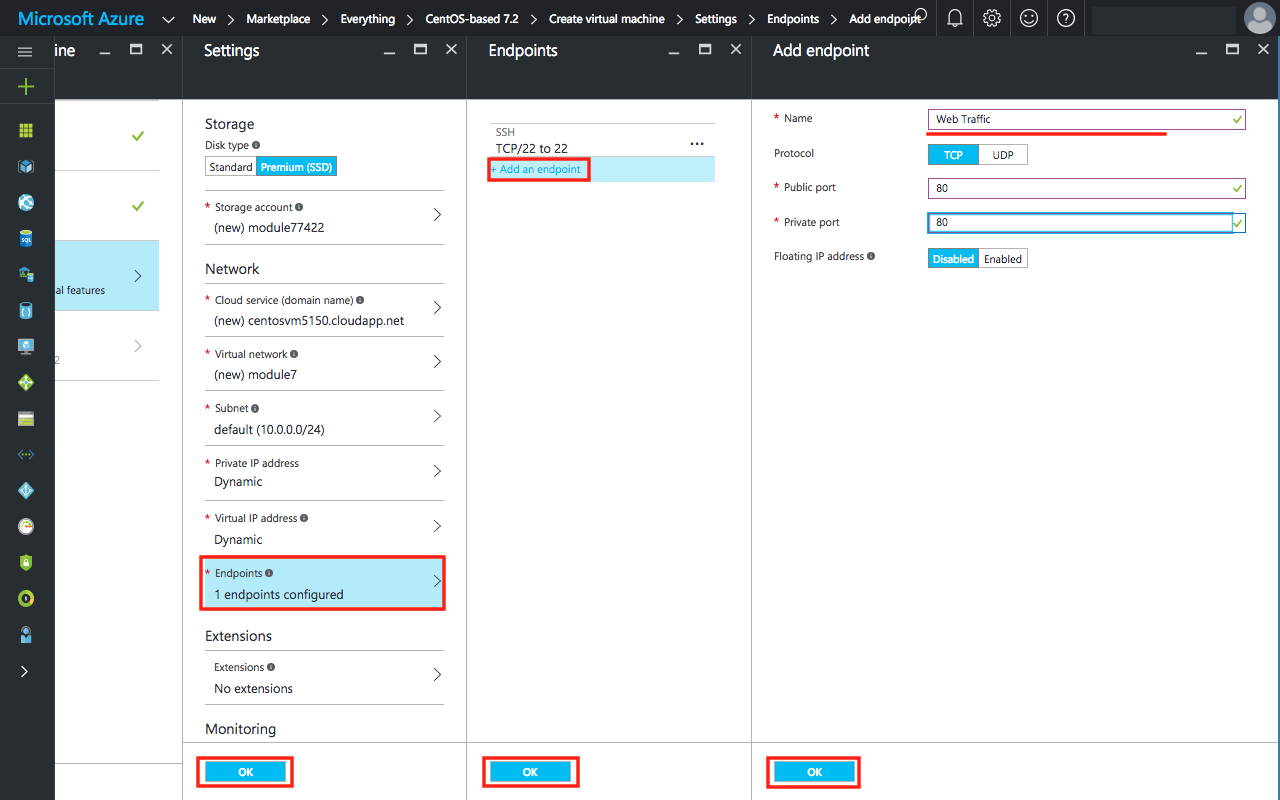
1. In a web browser, navigate to <http://portal.azure.com>. Sign into the portal using your subscription.
2. In the Azure portal, click “New” to launch a new virtual machine
3. In the next menu enter “centos” in the search box
4. Choose the “CentOS-based 7.x” image. This can be any of the version 7 CentOS platforms (7.0, 7.1, 7.2, etc). This lab was written using CentOS 7.1. 
5. Choose the CentOS-based 7.x image  
     
   
6. Choose the Classic deployment model
7. Click Create

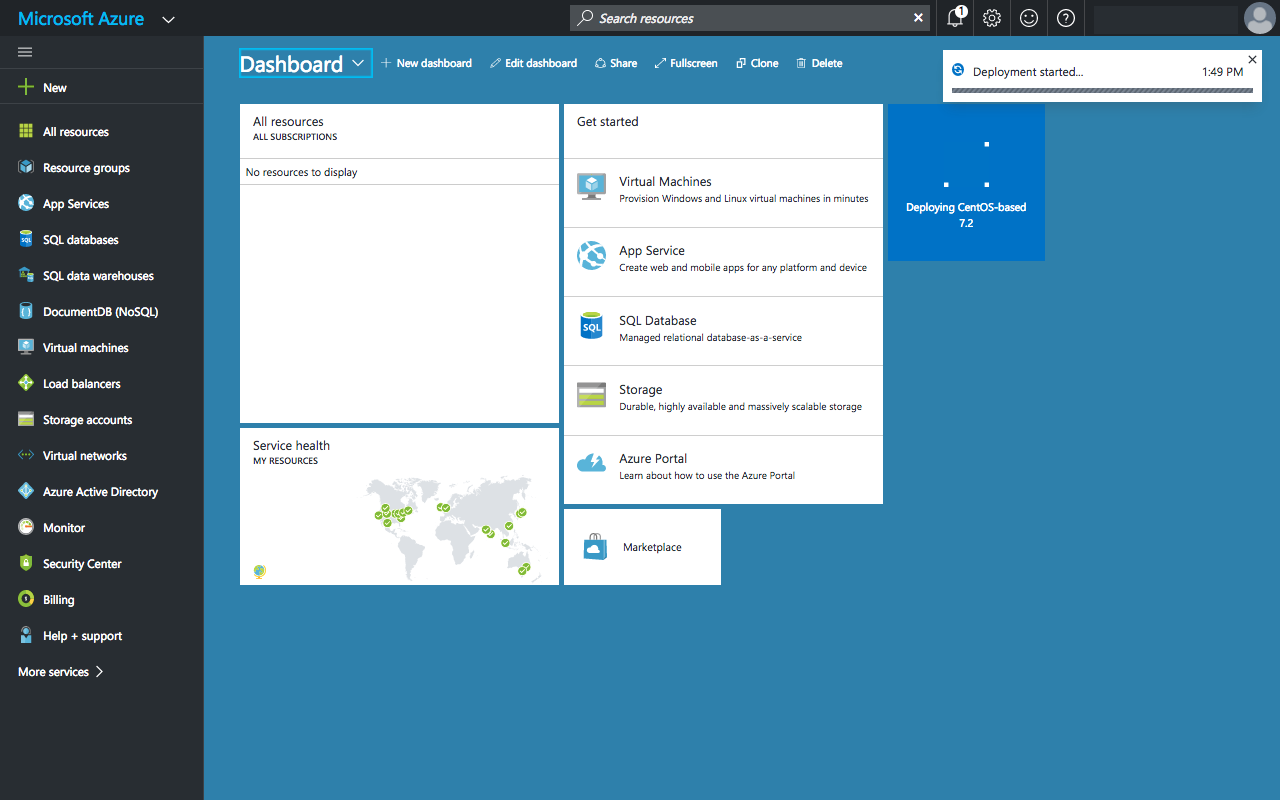


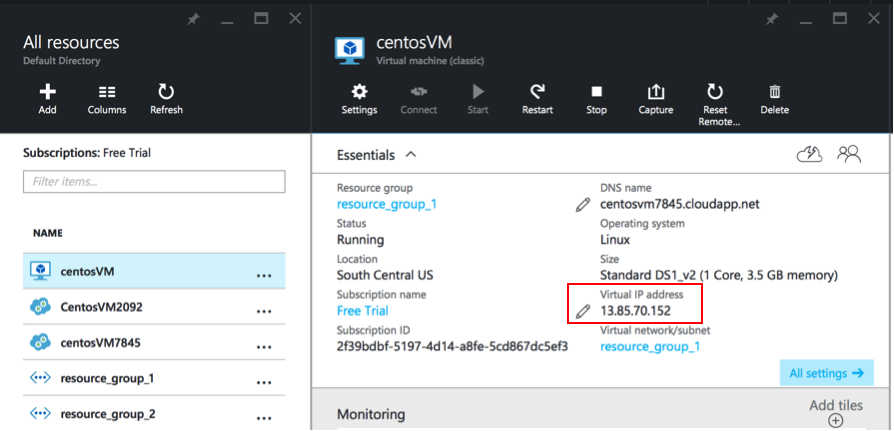
1. Complete the form
   1. Use any VM Name
   2. Use any Username
   3. Create a Password
   4. Create a new Resource Group (with any name)
   5. Choose the default Location
   6. Click ‘Next’

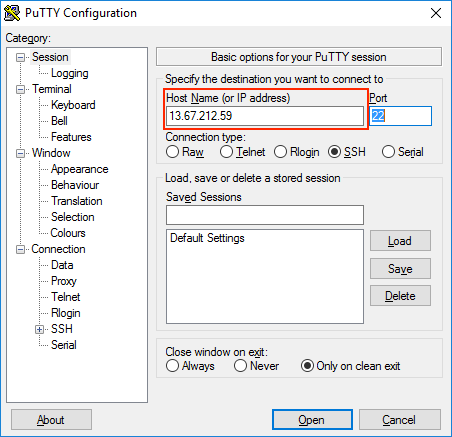


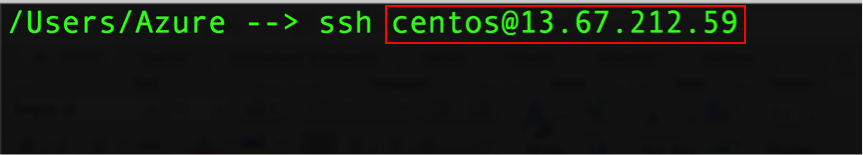
1. Choose Machine Size
   1. Choose the smallest machine size offered
   2. Click ‘Select’
2. Configure the Endpoint
3. Click ‘Add an endpoint’
4. Complete the form
   1. Use TCP as the Protocol
   2. Give the Endpoint a name
   3. Use port 80 as the Public and Private port
   4. Click OK for each screen to launch the virtual machine



1. Your virtual machine will launch
2. Find the IP Address of your new VM
   1. Click ‘All resources’ from the main Dashboard
   2. Find your new VM
   3. Find the IP Address

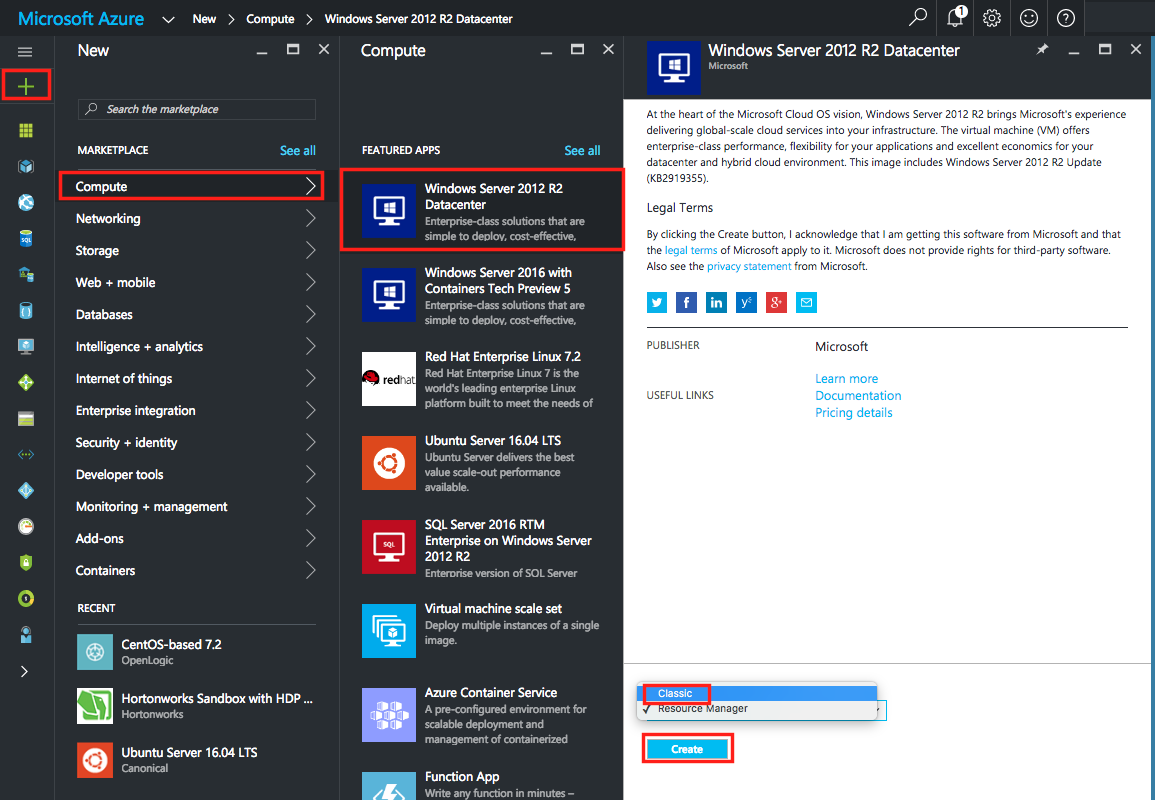
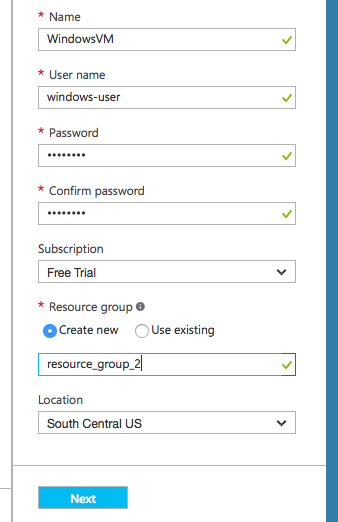
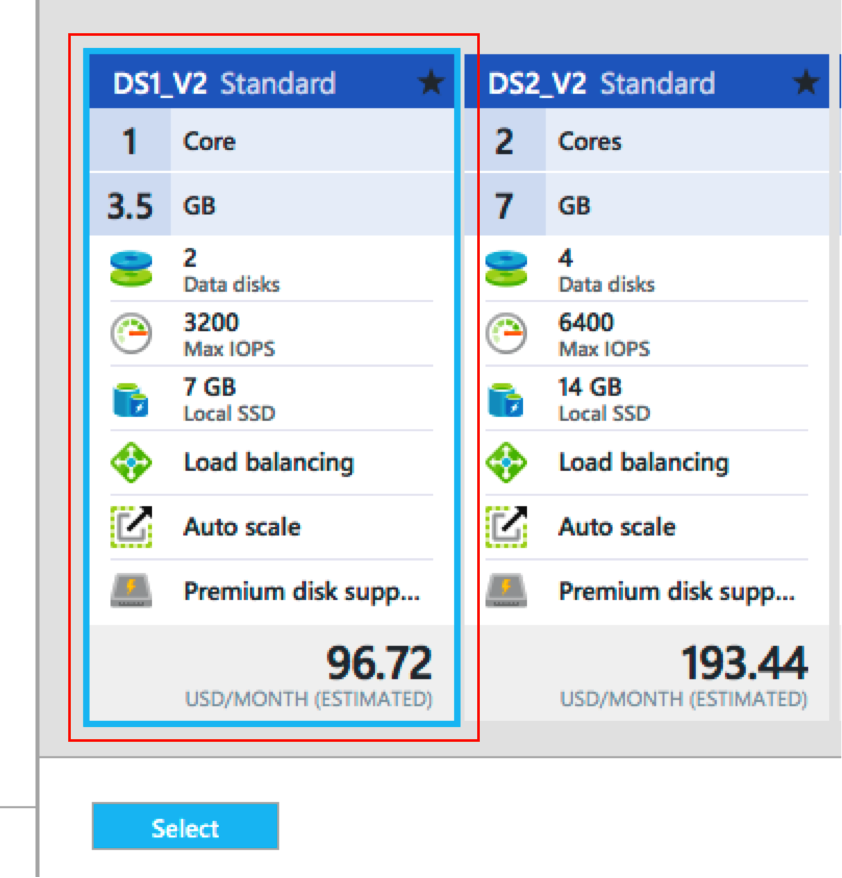
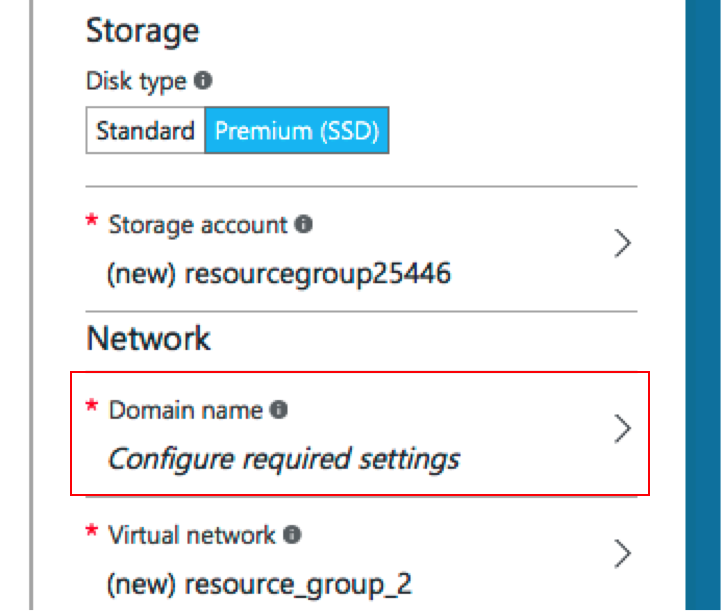
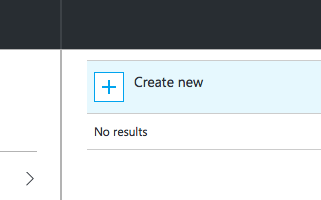
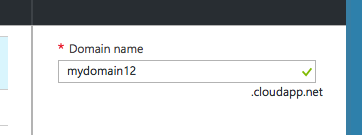
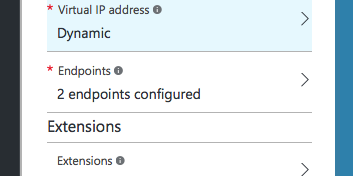
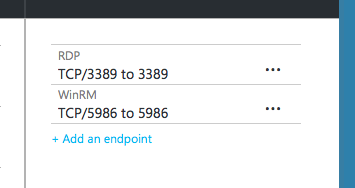


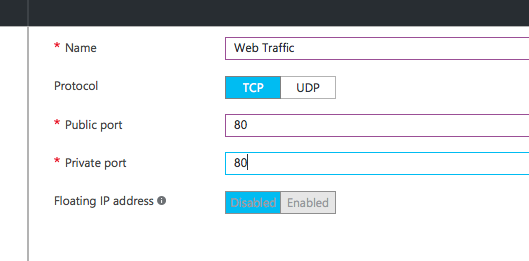
1. Connect to the VM with a PuTTY for Windows or Terminal for Mac
2. To use PuTTY, open a PuTTY console, with the following configuration (using your IP address):  
     
   
3. To use Terminal for Mac, run the command:  
   ssh username@ipaddress  
   For example: ssh [centos@13.67.212.59](mailto:centos@13.67.212.59)

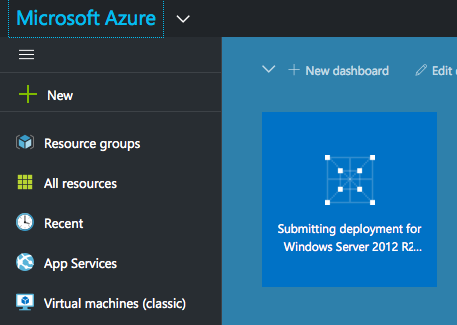


## Exercise 2: Launch a Windows Server 2012 R2 Datacenter VM

Now that you’ve created a CentOS VM, let’s launch a Windows VM

1. From the dashboard, click ‘New’ to start.
2. Click ‘Compute’
3. Choose the Windows Server 2012 R2 Datacenter image
4. Choose the Classic deployment model
5. Click Create  
     
   
6. Complete the form  
     
   
7. Choose the Machine Size (use the least expensive machine size)  
   
8. Configure the Domain Name  
   
9. Create a new Domain Name  
   
10. Create a new Domain Name. This can be arbitrary so if the domain name you have chosen is not available, create a different domain name of your choice. Click ‘OK’  
    
11. Create a new endpoint  
    
12. Click ‘Add an endpoint’ to configure the endpoints  
    
13. Configure the Endpoint
    1. Complete the form
    2. Use TCP as the Protocol
    3. Give the Endpoint a name
    4. Use port 80 as the Public and Private port



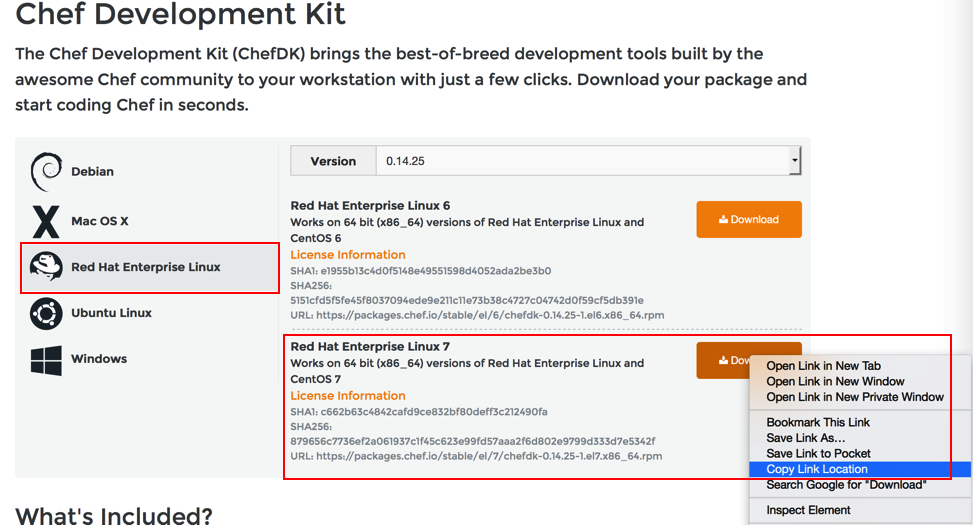
1. Click ‘OK’ several times to clear the various screens and launch your virtual machine.
2. Your instance will launch  
     
   
3. Find the new VM
   1. From the Dashboard click ‘All Resources’
   2. Use the search function
   3. Find the new VM
   4. Click on the VM to find its IP Address
4. Connect to the new Windows VM using Remote Desktop Connection (from a Windows workstation) or install Microsoft Remote Desktop (if you are using a Mac).
5. After clicking on the new VM from the ‘All Resources’ blade, click ‘Connect’ to download the RDP connection information (and open this connectoid in your RDP client), or use the IP Address and the username and password that you created to connect manually from the RDP client.

## Exercise 3: Install the Chef Development Kit (ChefDK) on each VM

Now that you’ve created two VM’s, let’s install the ChefDK on each VM

Part 1: installing the ChefDK on CentOS

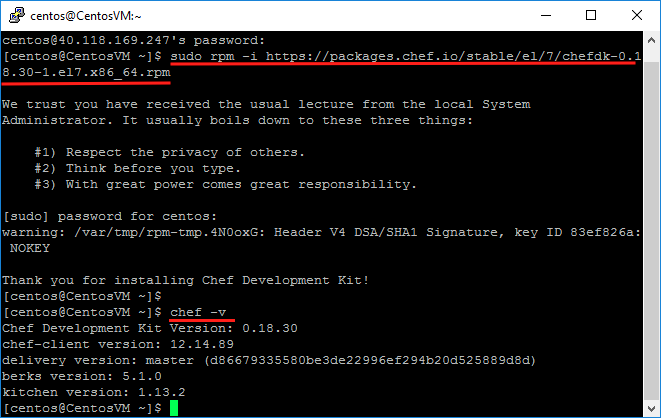
1. Find the ChefDK installation rpm (Red Hat Package Manager)
   1. Using a web browser, navigate to <http://downloads.chef.io/chef-dk>
   2. Choose ‘Red Hat Enterprise Linux’
   3. Find the Download button for ‘Red Hat Enterprise Linux 7’
   4. Right Click (to bring up the context menu) on the Download button and choose the option to copy or save the link location



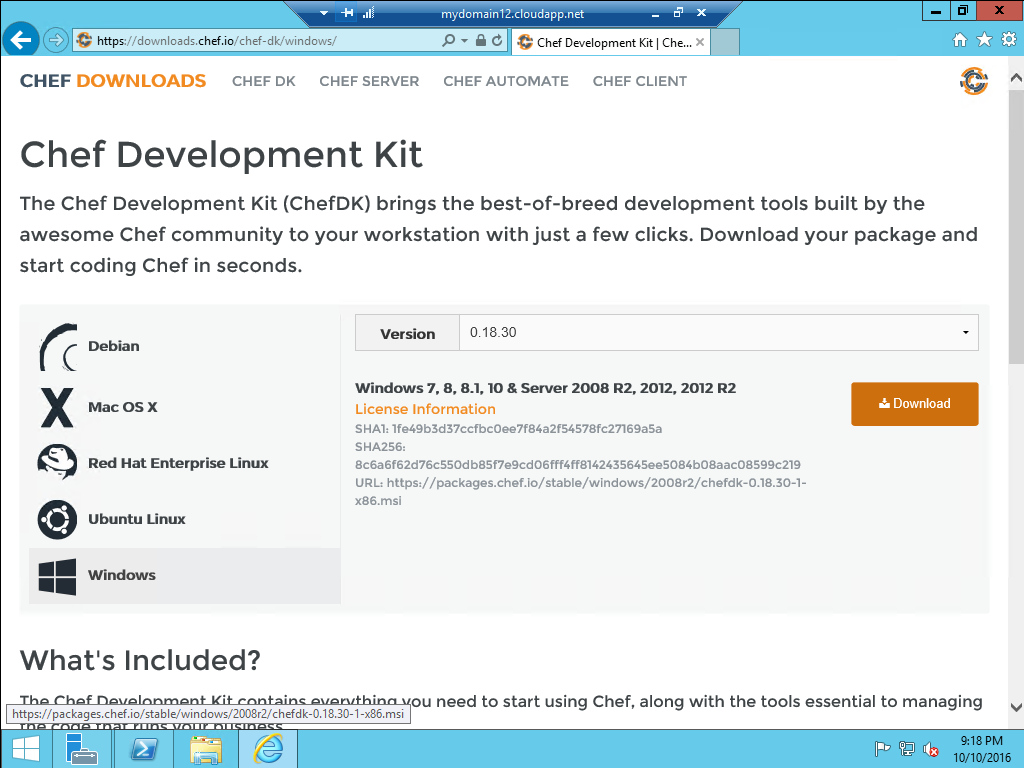
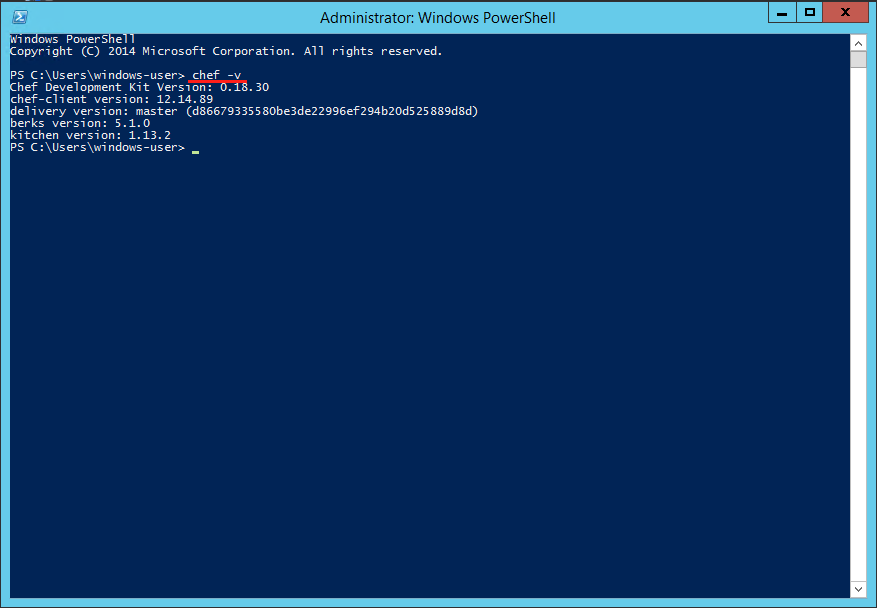
1. From the SSH session into the CentOS VM, run the command:

$sudo rpm -i <paste\_link\_from\_chefdk\_page>, for example:  
$sudo rpm -i <https://packages.chef.io/stable/el/7/chefdk-0.14.25-1.el7.x86_64.rpm>

1. Watch the ChefDK installation complete
2. Verify the ChefDK installation
   1. From the terminal window, execute the command **$chef -v**
   2. Reporting of the version numbers indicates a successful installation



Part 2: installing the ChefDK on the Windows VM

1. Note: this operation is done from the new Windows VM that you just launched
2. Find the ChefDK installation Windows msi
   1. Using a web browser from the VM, navigate to <http://downloads.chef.io/chef-dk>
3. Choose ‘Windows’
4. Find the Download button for ‘Windows’
   1. Note: if you do not see the download button, install another browser, such as Firefox
5. Download and install the Windows msi  
   
6. Verify the ChefDK installation
   1. From a PowerShell window, execute the ‘chef -v’ command   
      PS> **chef -v**
   2. Reporting of the version numbers indicates a successful installation  
      

## Exercise 4: Create a cookbook and recipe for CentOS, converge the node and verify your functioning web server

Now that you’ve installed the ChefDK, let’s write a recipe:

1. Create a ‘cookbooks’ directory
   1. Log into the Linux virtual machine using SSH from either Terminal (on a Mac) or PuTTY (from Windows)
   2. From the home directory, create a ‘cookbooks’ directory  
      **$mkdir cookbooks**
   3. This is where cookbooks will be stored
2. Create a new cookbook
   1. From the home directory, use the **chef generate** command to create the apache cookbook. This will create the specific cookbook directory and its needed subdirectories and files:  
      **$chef generate cookbook cookbooks/apache**  
      Note: this assumes you are running the command from the home directory, hence the path of **cookbooks**/apache
3. Create a new recipe, to be stored in the new cookbook, that contains the code to install, configure and start the web server
   1. From the SSH terminal, create a new file named   
      ‘install-apache.rb’ in the ~/cookbooks/apache/recipe directory
   2. Use any editor. Nano is a good editor for those not familiar with vi or emacs.  
      **$nano /home/<user>/cookbooks/apache/recipes/install-apache.rb**
4. Add the Chef code to install the Apache web server
   1. In the new file, add the resources to install, configure and start Apache
   2. Start with the ‘package’ resource to install Apache
   3. The package name is ‘httpd’  
        
      package ‘httpd’ do  
       action :install  
      end
5. Instruct Chef to write a web page for Apache to serve up
   1. Use the ‘file’ resource to create a web page
   2. The file name is ‘/var/www/html/index.html’. This is the path on the VM where we want to write the HTML file. This location was chosen because it is the default path and filename where Apache will look for the HTML file to serve when a request comes in on port 80.

file ‘/var/www/html/index.html’ do  
 content ‘<html><body><h1>Hello World</h1></body></html>’  
 action :create  
end

1. Start the Apache web service
   1. Use the ‘service’ resource to start Apache
   2. ‘httpd’ is the service name
   3. Brackets denote that two separate actions are being implemented
   4. The two actions ‘enable’ the service to start upon reboot as well ‘start’ the service now

service ‘httpd’ do  
 action [:enable, :start]   
end

1. Completed install-apache.rb File

package ‘httpd’ do

action :install

end

file ‘/var/www/html/index.html’ do

content ‘<html><body><h1>Hello World</h1></body></html>’

action :create

end

service ‘httpd’ do

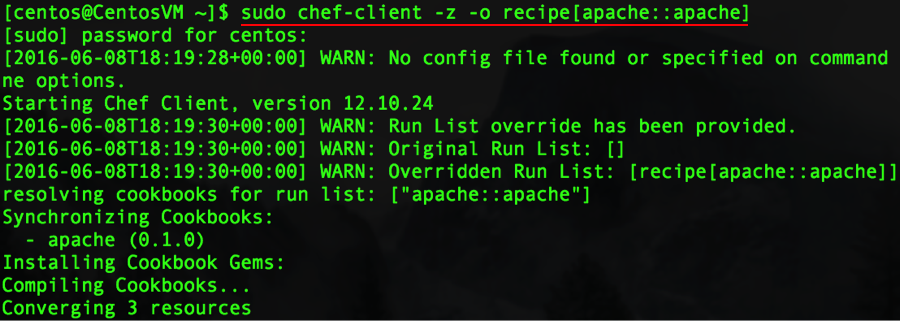
action [:enable, :start]

end

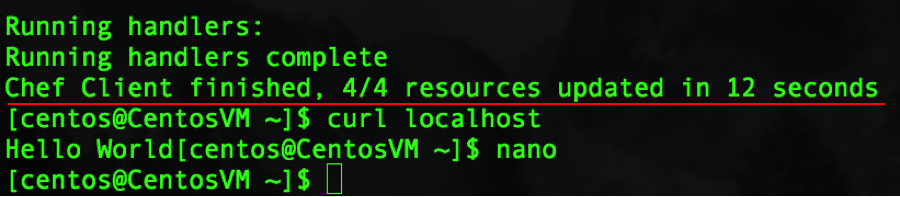
1. Converge the Node
   1. Instruct the node to execute the Chef recipe
   2. From the SSH command line, in the home directory, invoke chef-client
   3. First, move to the home directory  
      **$cd ~**
   4. Next, execute the command as follows. Note the –z option instructs chef-client to run in   
      local mode and not use a Chef Server, and the –o option points to the recipe we want to run and the cookbook in which that recipe exists.

$sudo chef-client -z -o recipe[apache::install-apache]

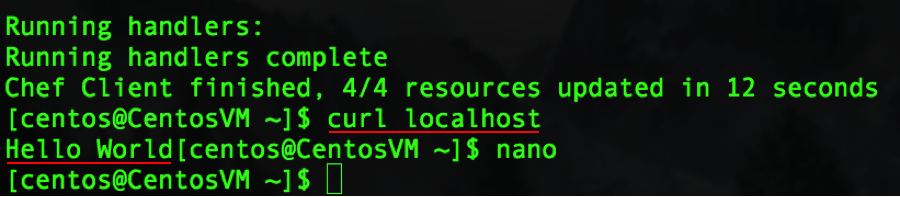
* 1. Watch the node install, configure and start the Apache service



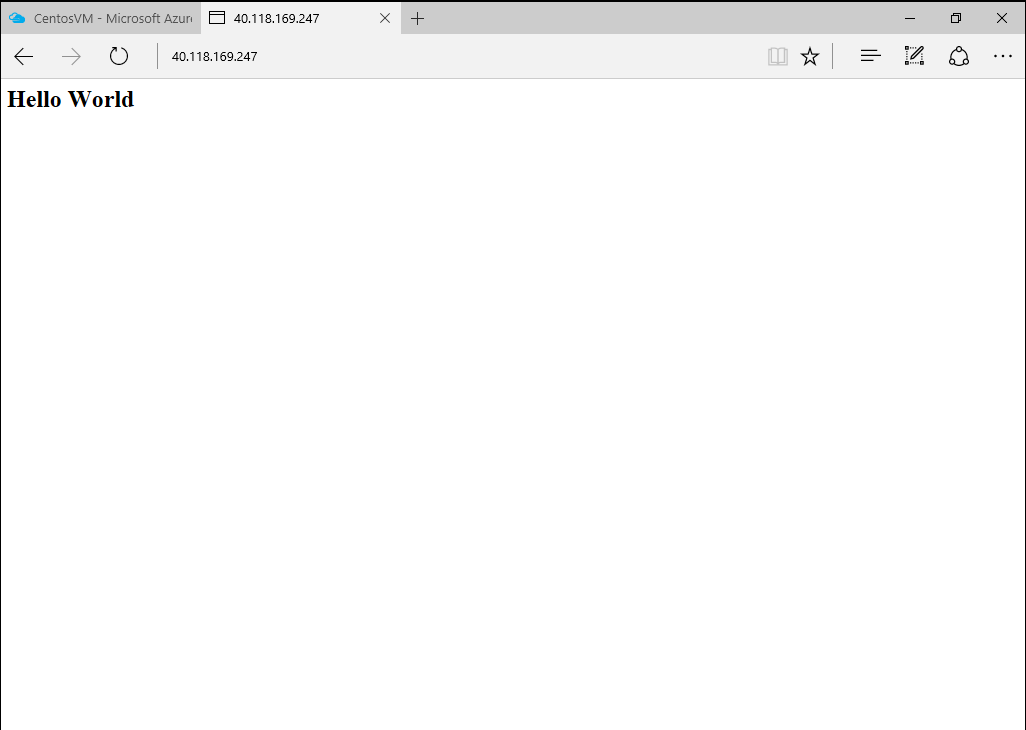
* 1. Note the following:
     1. ‘Chef Client finished’ message
     2. Total number of resources updated
     3. Time taken by the run



1. Verify Web Server Functionality via the command line to eliminate potential VM networking issues
   1. **$curl localhost** can verify (without being limited by network issues):
      1. the web server has been installed
      2. the httpd service is running
      3. the web page is being served correctly



1. Verify Web Server Functionality visually
   1. Find the IP Address of the virtual machine
   2. Enter the IP Address into a web browser and see the ‘Hello World’ page displayed



## Exercise 5: Create a cookbook and recipe for Windows Server 2012 R2 Datacenter, converge the node and verify your functioning web server

Now that you’ve got a functioning CentOS-based webserver, let’s do the same for Windows

1. Create a cookbooks directory
   1. Log into the Windows instance using RDP
   2. From the home directory, create a cookbooks directory  
      **PS>mkdir cookbooks**
   3. This is where cookbooks will be stored
2. Create a new cookbook
   1. From the home directory, use the **chef generate** command to create the iis cookbook. This will create the specific cookbook directory its needed subdirectories and files:  
      **PS> chef generate cookbook cookbooks/iis**  
      Note: this assumes you are running the command from the home directory, hence the path of **cookbooks**/iis
3. Create a new recipe, to be stored in the new cookbook, that contains the code to install, configure and start the web server
   1. From the Windows RDP Session, create a new file named   
      ‘install-iis.rb’ in the ~/cookbooks/iis/recipe directory
   2. Use any editor. You can install an editor such as Notepad++, Sublime Text or Atom, or for just this exercise, you can use Notepad
4. Add the Chef code to install, configure and start the IIS web server
   1. Start with the ‘powershell\_script’ resource to install IIS
      1. The package name is ‘Install IIS’ but with PowerShell Script this doesn’t impact the installation at all. It is the ‘code’ attribute that does the actual install.
      2. The ‘guard\_interpreter ‘ determines which interpreter will implement the idempotence (in case the web server is already installed)
      3. The ‘not\_if’ line is the logic for the idempotence, stating that if the web server is NOT installed, then do this installation, otherwise if the web server IS installed, skip the installation.

powershell\_script 'Install IIS' do

code 'Add-WindowsFeature Web-Server'

guard\_interpreter :powershell\_script

not\_if "(Get-WindowsFeature -Name Web-Server).Installed"

action :run

end

1. Instruct Chef to write a web page for IIS to serve up
   1. Use the ‘file’ resource to create a web page
   2. The file name is ‘c:\inetpub\wwwroot\Default.htm’. This is the path on the VM where we want to write the HTML file. This location was chosen because it is the default path and filename where IIS will look for the HTML file to serve when a request comes in on port 80.  
        
      file ‘c:\inetpub\wwwroot\Default.htm’ do  
       content ‘<html><body><h1>Hello World</h1></body>’  
       action :create  
      end
2. Start the IIS web service
   1. Use the ‘service’ resource to start IIS
   2. ‘w3svc’ is the service name for IIS
   3. Brackets denote that two separate actions are being implemented
   4. The two actions ‘enable’ the service to start upon reboot as well ‘start’ the service now  
        
      service ‘w3svdc’ do  
       action [:enable, :start]   
      end
3. Completed install-iis.rb File (verify that your recipe is named correctly from the Powershell command line)

powershell\_script 'Install IIS' do

code 'Add-WindowsFeature Web-Server'

guard\_interpreter :powershell\_script

not\_if "(Get-WindowsFeature -Name Web-Server).Installed"

action :run

end

file 'c:\inetpub\wwwroot\Default.htm' do

content '<html><body><h1>Hello World</h1></body></html>'

action :create

end

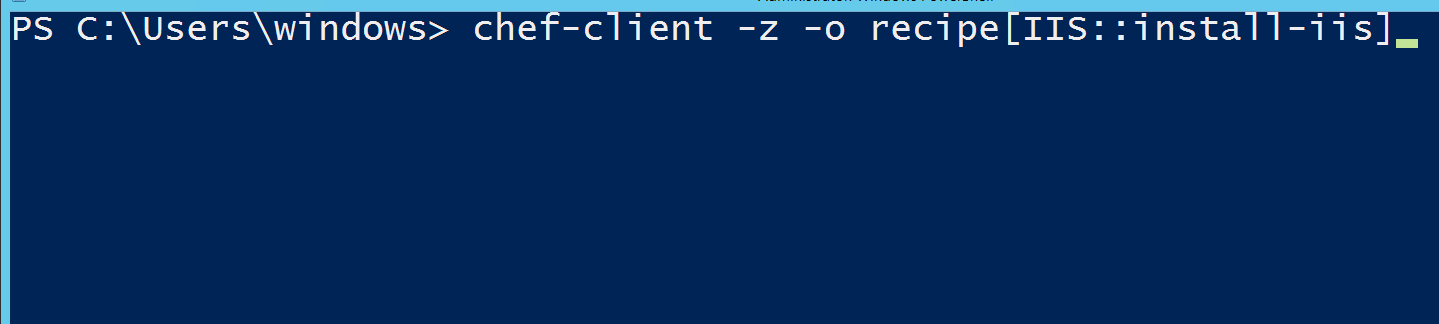
service 'w3svc' do

action [:enable, :start]

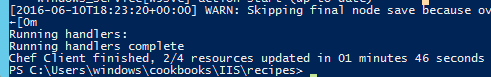
end

1. Converge the Node
   1. Instruct the node to execute the Chef recipe
   2. From the Powershell command line, in the home directory, invoke chef-client
   3. First, move to the home directory  
      **$cd ~**
   4. Next, execute the command as follows. Note the –z option instructs chef-client to run in local mode and not use a Chef Server, and the –o option points to the recipe we want to run and the cookbook in which that recipe exists.

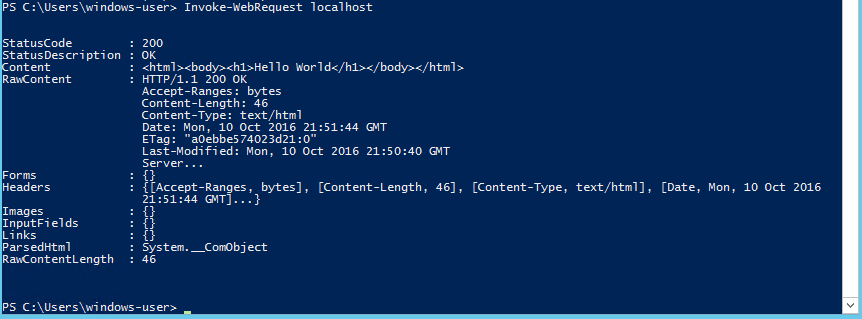
chef-client -z -o recipe[iis::install-iis]



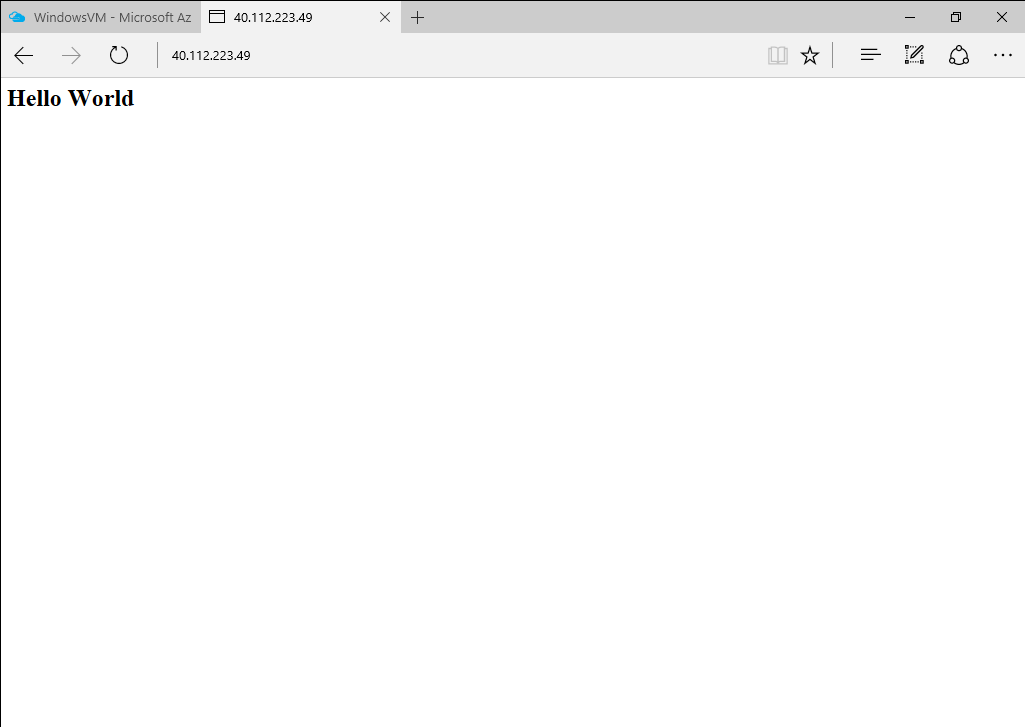
* 1. Watch the node install, configure and start the IIS service
  2. Note the following:
     1. ‘Chef Client finished’ message
     2. Total number of resources updated
     3. Time taken by the run



1. Verify Web Server Functionality via the command line to eliminate potential VM networking issues
   1. **PS>Invoke-WebRequest localhost** can verify (without being limited by network issues):
      1. the web server has been installed
      2. the w3svc (IIS) service is running
      3. the web page is being served correctly



1. Verify Web Server Functionality visually
   1. Find the IP Address of the virtual machine
   2. Enter the IP Address into a web browser and see the ‘Hello World’ page displayed



\*\*\*Don’t forget to delete your VM’s in order to avoid wasting your Azure credit\*\*\*

## Summary

In this hands-on lab, you learned how to:

* Launch a virtual CentOS 7.1 and Windows Server 2012 R2 Datacenter machine
* Install the Chef Development Kit on each VM
* Create a web server cookbook on each VM
* Write a recipe to install, configure and start a web server on each VM
* Verify that the web server is running from the command line and from a web browser