Setup Kubernetes natively on Windows 10

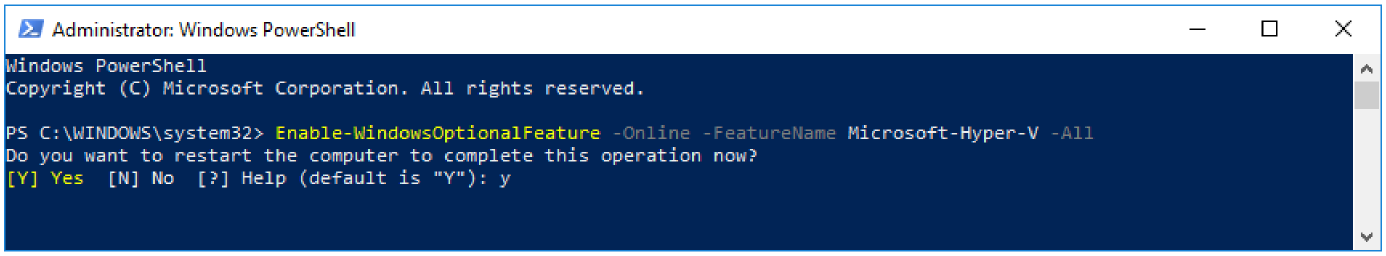
In this lab you are going to setup a native Docker+Kubernetes environment on Windows 10 (Pro, Enterprise, or Education) which can be used for the rest of the SIG labs. For the installation we are going to use `Chocolatey` a Software Package Manager for Windows (similar to apt-get on Linux).

Prerequisites

Before getting started with this lab have Hyper-V enabled on your Windows machine. Open a PowerShell console (run as administrator) and enter the following command:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

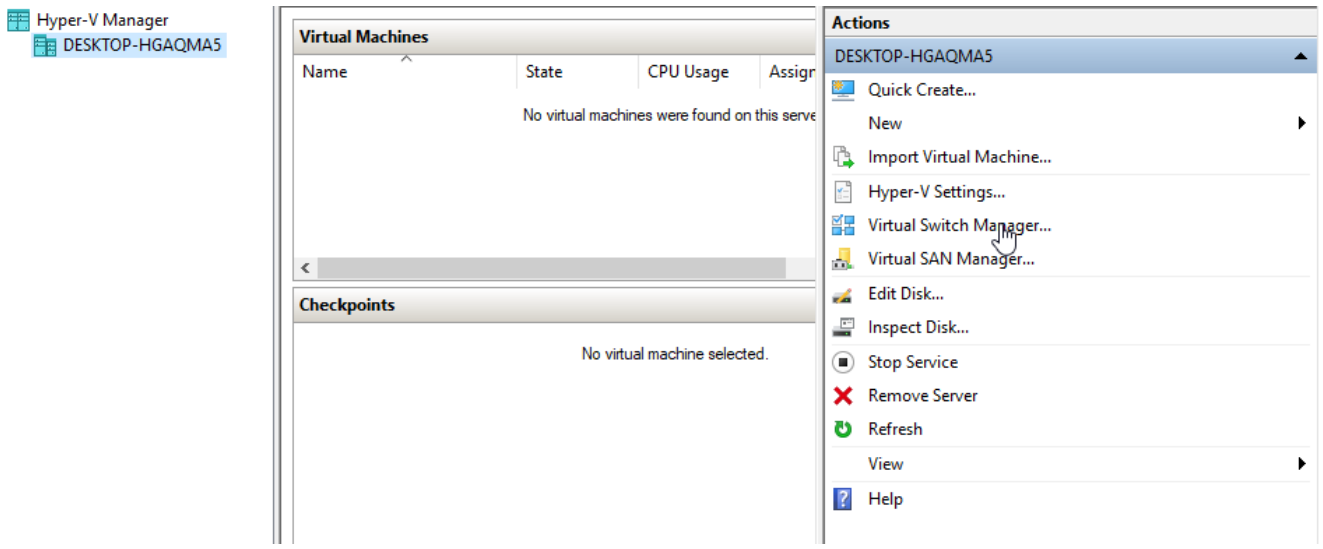
If the command couldn't be found, make sure you're running PowerShell as Administrator. When the installation has completed, **reboot your system**.



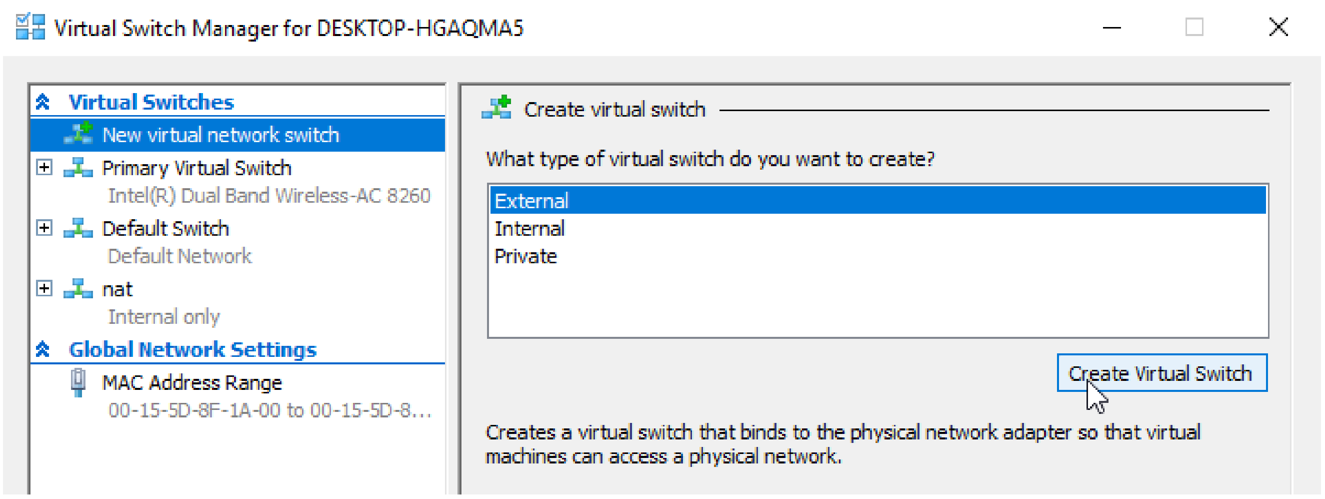
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
Do you want to restart the computer to complete this operation now?
[Y] Yes  [N] No  [?] Help (default is "Y"): y
```

After enabling Hyper-V on your machine open `Hyper-V Manager` (Windows+R).

1. Click in manager on **Virtual Switch Manager**



2. Create a new create a new **External virtual switch**.


Virtual Switch Manager for DESKTOP-HGAQMA5

Virtual Switches
- New virtual network switch
- Primary Virtual Switch
  Intel(R) Dual Band Wireless-AC 8260
- Default Switch
  Default Network
- nat
  Internal only

Global Network Settings
- MAC Address Range
  00-15-5D-8F-1A-00 to 00-15-5D-8...

Create virtual switch

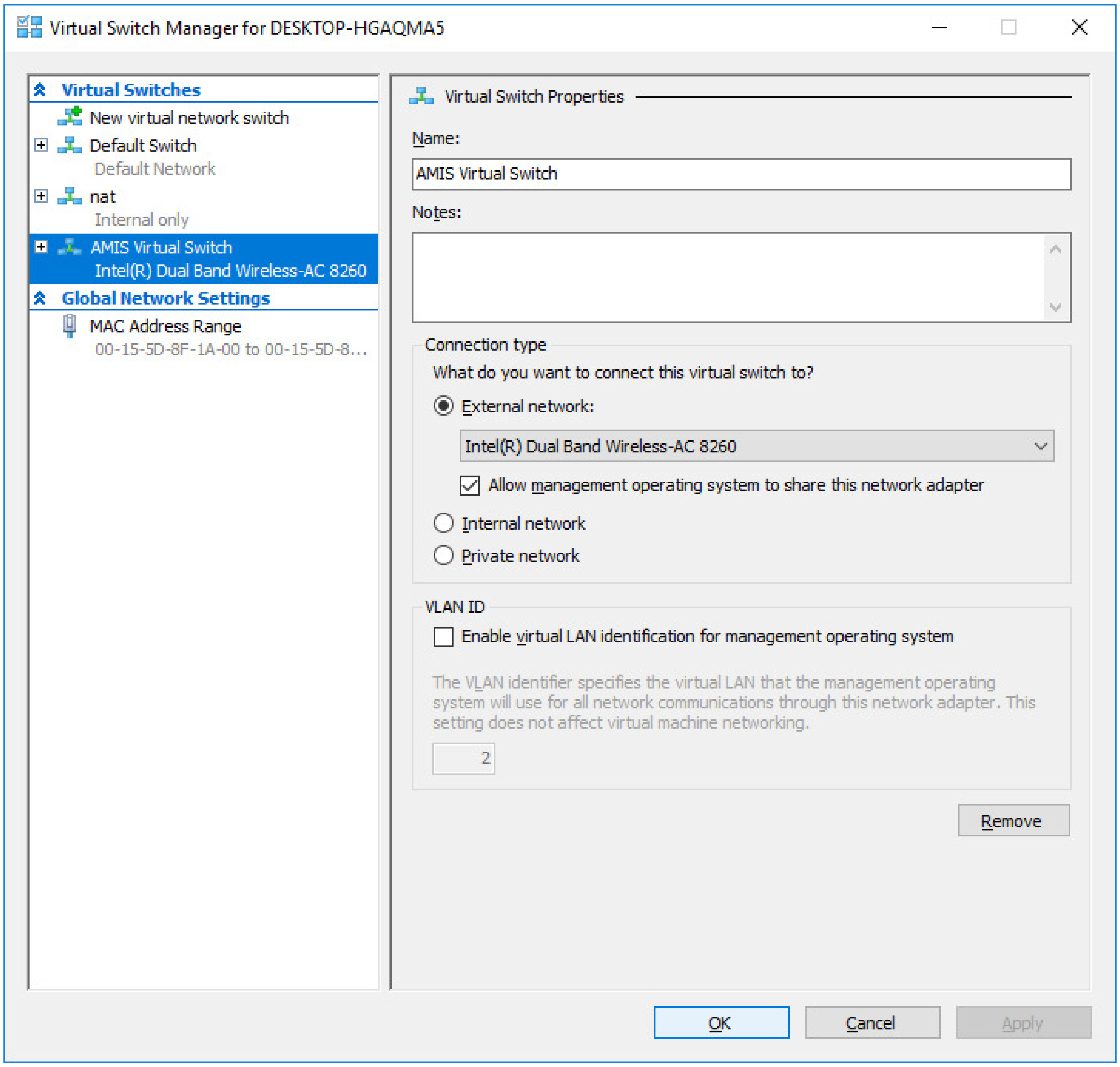
What type of virtual switch do you want to create?

External
Internal
Private

Create Virtual Switch

Creates a virtual switch that binds to the physical network adapter so that virtual machines can access a physical network.

3. Use the name: `AMIS Virtual Switch`.


Virtual Switch Manager for DESKTOP-HGAQMA5

Virtual Switches
- New virtual network switch
- Default Switch
  Default Network
- nat
  Internal only
- AMIS Virtual Switch
  Intel(R) Dual Band Wireless-AC 8260

Global Network Settings
- MAC Address Range
  00-15-5D-8F-1A-00 to 00-15-5D-8...

Virtual Switch Properties

Name: AMIS Virtual Switch

Notes:

Connection type

What do you want to connect this virtual switch to?

● External network:

Intel(R) Dual Band Wireless-AC 8260

☑ Allow management operating system to share this network adapter

○ Internal network

○ Private network

VLAN ID

☐ Enable virtual LAN identification for management operating system

The VLAN identifier specifies the virtual LAN that the management operating system will use for all network communications through this network adapter. This setting does not affect virtual machine networking.

2

4. Accept the settings by clicking on **OK**.

This virtual switch is going to be used when starting `minikube` for the first time.

Step 1/7 - Install Chocolatey

The sane way to manage software on Windows ;). Chocolatey is a single, unified interface designed to easily work with all aspects of managing Windows software (installers, zip archives, runtime binaries, internal and 3rd party software) using a packaging framework that understands both versioning and dependency requirements.

You can download the installation manually from https://chocolatey.org/install#installing-chocolatey or user PowerShell by executing the following command:

```
Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

This will install `chocolatey` to your machine. You should expect similar output as below.

```
Getting latest version of the Chocolatey package for download.
...
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
  and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
...
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
 first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
```

Step 2/7 - Install Boxstarter

Boxstarter leverages Chocolatey packages to automate the installation of software and create repeatable, scripted Windows environments. Chocolatey makes installing software very easy with no user intervention. Boxstarter enhances Chocolatey's functionality and provides an environment that is optimized for installing a complete environment on a fresh OS install, as well as some other specific scenarios.

We use it here to be able to start/open a PowerShell console with the correct privileges and environment settings out-of-the-box. Run the following command in your current PowerShell console:

```
choco install boxstarter -y
```

This will install `boxstarter` to your machine. You should expect similar output as below.

```
Chocolatey v0.10.11
```

```
Upgrading the following packages:
boxstarter
By upgrading you accept licenses for the packages.

You have Boxstarter v2.11.0 installed. Version 2.12.0 is available based on your source(s).
Progress: Downloading BoxStarter.Common 2.12.0... 100%
Progress: Downloading boxstarter 2.12.0... 100%
Progress: Downloading BoxStarter.WinConfig 2.12.0... 100%
Progress: Downloading boxstarter.bootstrapper 2.12.0... 100%
Progress: Downloading boxstarter.chocolatey 2.12.0... 100%
Progress: Downloading Boxstarter.HyperV 2.12.0... 100%
...
BoxStarter.HyperV v2.12.0 [Approved]
boxstarter.hyperv package files upgrade completed.
```

Step 3/7 - Start Boxstarter Shell

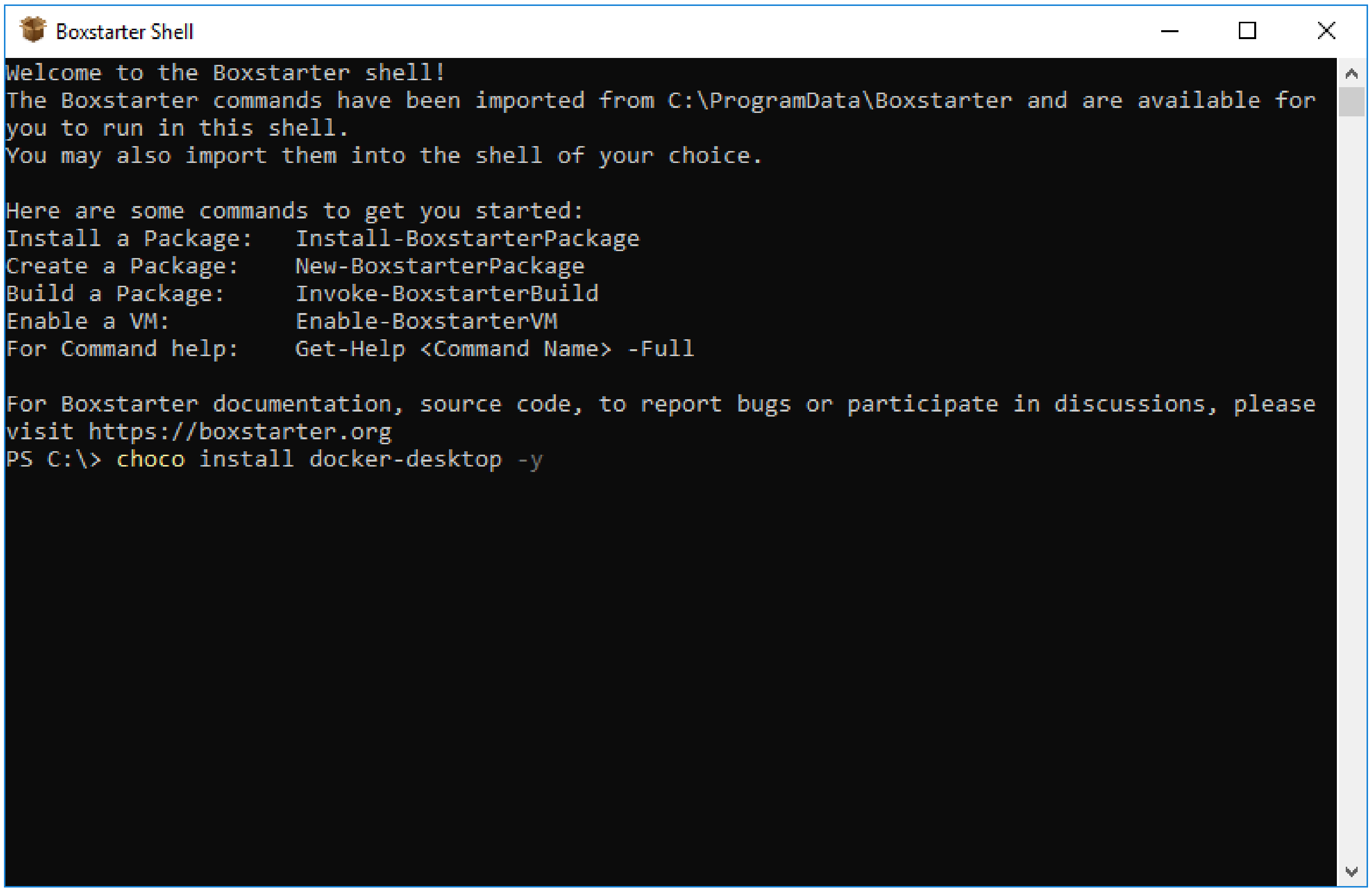
After installation we can start a Boxstart Shell by clicking on the desktop icon or execute the following command (Windows+R):

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy bypass -NoExit -
Command "&'C:\ProgramData\Boxstarter\BoxstarterShell.ps1'"
```

Step 4/7 - Install Docker Desktop 2.0

Next step is to install Docker Desktop 2.0+ (latest version) if you not already have the latest version installed on your machine. In the `Boxstarter Shell` run the following command:

```
choco install docker-desktop -y
```



```
Welcome to the Boxstarter shell!
The Boxstarter commands have been imported from C:\ProgramData\Boxstarter and are available for
you to run in this shell.
You may also import them into the shell of your choice.

Here are some commands to get you started:
Install a Package:   Install-BoxstarterPackage
Create a Package:    New-BoxstarterPackage
Build a Package:     Invoke-BoxstarterBuild
Enable a VM:         Enable-BoxstarterVM
For Command help:    Get-Help <Command Name> -Full

For Boxstarter documentation, source code, to report bugs or participate in discussions, please
visit https://boxstarter.org
PS C:\> choco install docker-desktop -y
```

This will install `docker-desktop` to your Windows 10 machine. You should expect similar output as below.

```
docker-desktop v2.0.0.0 [Approved]
docker-desktop package files install completed. Performing other installation steps.
The package docker-desktop wants to run 'chocolateyinstall.ps1'.
Do you want to run the script?([Y]es/[N]o/[P]rint): y

Downloading docker-for-windows
  from 'https://download.docker.com/win/stable/30215/Docker%20for%20Windows%20Installer.exe'
Progress: 100% - Completed download of
C:\Users\robert_m\AppData\Local\Temp\chocolatey\docker-desktop\2.0.0.2\Docker for Windows
Installer.exe (537.71 MB).
Installing docker-for-windows...

docker-for-windows has been installed.
  docker-desktop may be able to be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
 see the changes (or in powershell/cmd.exe just type `refreshenv`).
 The install of docker-desktop was successful.
  Software installed to 'C:\Program Files\Docker\Docker'
```

Test Docker installation

Check if the installation was correct by checking its version by running `docker -v` in the *boxstarter* shell.

```
PS C:\> docker -v

Docker version 18.09.1, build 4c52b90
```

Step 5/7 - Install Kubectl

Next we are going to install the Kubernetes command-line tool, `kubectl`, to deploy and manage applications on Kubernetes. Using kubectl, we can inspect cluster resources; create, delete, and update components; look at our new cluster; bring up example apps and scale-up and down. In the `Boxstarter Shell` run the following command:

```
choco install kubernetes-cli -y
```

This will install `kubectl` to you Windows 10 machine. You should expect similar output as below.

```
kubernetes-cli v1.13.2 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.

Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-
windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-
windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
 ShimGen has successfully created a shim for kubectl.exe
 The install of kubernetes-cli was successful.
  Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'
```

Test Kubectl installation

Check your installation by retrieving its version. Type `kubectl version` in the *boxstarter* shell.

```
PS C:\> kubectl version

Client Version: version.Info{Major:"1", Minor:"10", GitVersion:"v1.10.11",
GitCommit:"637c7e288581ee40ab4ca210618a89a555b6e7e9", GitTreeState:"clean", BuildDate:"2018-
11-26T14:38:32Z", GoVersion:"go1.9.3", Compiler:"gc", Platform:"windows/amd64"}
```

~~ Ignore the connection error, as we do net have a running cluster ~~

Create config folder in user home

To save environment settings and configurations we need to create a .kube folder in our user home. Run the command `mkdir ~\.kube` to create the folder.

```
PS C:\> mkdir ~\.kube

Directory: C:\Users\robert_m

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        1/25/2019  11:58 AM                .kube

------------------------------
```

Step 6/7 - Install Minikube

The final component we need to install is Minikube. Minikube is a tool that makes it easy to run Kubernetes locally. Minikube runs a single-node Kubernetes cluster inside a VM on your laptop to deploy our solution on. In the `Boxstarter Shell` run the following command:

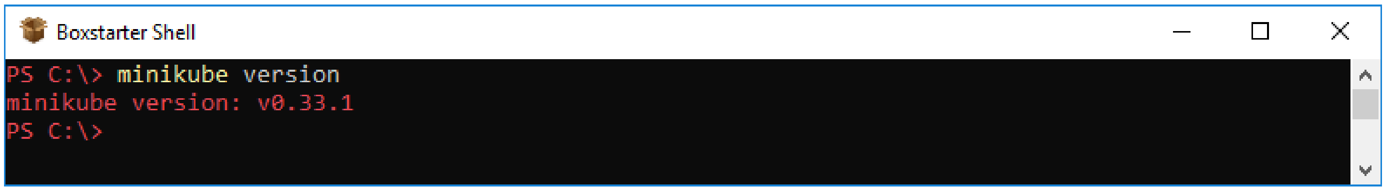
```
choco install minikube -y
```

This will install `minikube` to your Windows 10 machine. You should expect similar output as below.

```
Progress: Downloading Minikube 0.33.1... 100%

Minikube v0.33.1 [Approved]
minikube package files upgrade completed. Performing other installation steps.
 ShimGen has successfully created a shim for minikube.exe
 The install of minikube was successful.
  Software install location not explicitly set, could be in package or
  default install location if installer.
```

Test Minikube installation

To test if the installation was successful type in and execute the command `minikube version` in the boxstarter shell. This should return the same version as installed, as shown in the image below.



```
PS C:\> minikube version
minikube version: v0.33.1
PS C:\>
```

Step 7/7 - Create/Start Minikube cluster

The last steps we need to do to prepare our environment is to create, config and start a Minikube instance of the Hyper-V VM. To create/start a local Kubernetes cluster executed the following command in the `Boxstarter Shell`. Notice the command uses the configured Hyper-V virtual switch **AMIS Virtual Switch**.

```
minikube start --vm-driver hyperv --hyperv-virtual-switch "AMIS Virtual Switch"
```

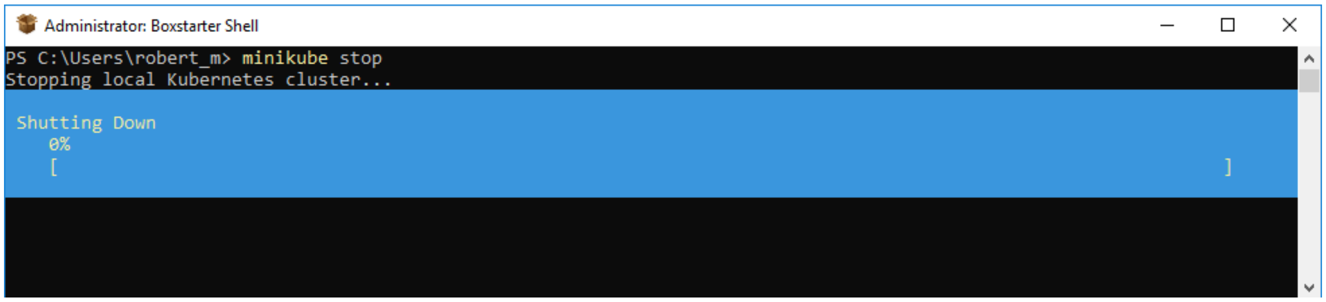
This will download and create the `minikube` on your Windows 10 machine. You should expect similar output as below. This process will take up a few minutes.

```
Starting local Kubernetes v1.13.2 cluster...
Starting VM...
Downloading Minikube ISO
 181.48 MB / 181.48 MB [============================================] 100.00% 0s
Getting VM IP address...
Moving files into cluster...
Downloading kubelet v1.13.2
Downloading kubeadm v1.13.2
Finished Downloading kubeadm v1.13.2
Finished Downloading kubelet v1.13.2
Setting up certs...
Connecting to cluster...
Setting up kubeconfig...
Stopping extra container runtimes...
Starting cluster components...
Verifying kubelet health ...
Verifying apiserver health ...Kubectl is now configured to use the cluster.
Loading cached images from config file.

Everything looks great. Please enjoy minikube!
```

Change Minikube VM settings

Because of a bug with Hyper-V and dynamically addressing memory we need to change the VM settings. Before we can change the settings we need to stop **Minikube** and *disable dynamic memory*. Perform the following steps:

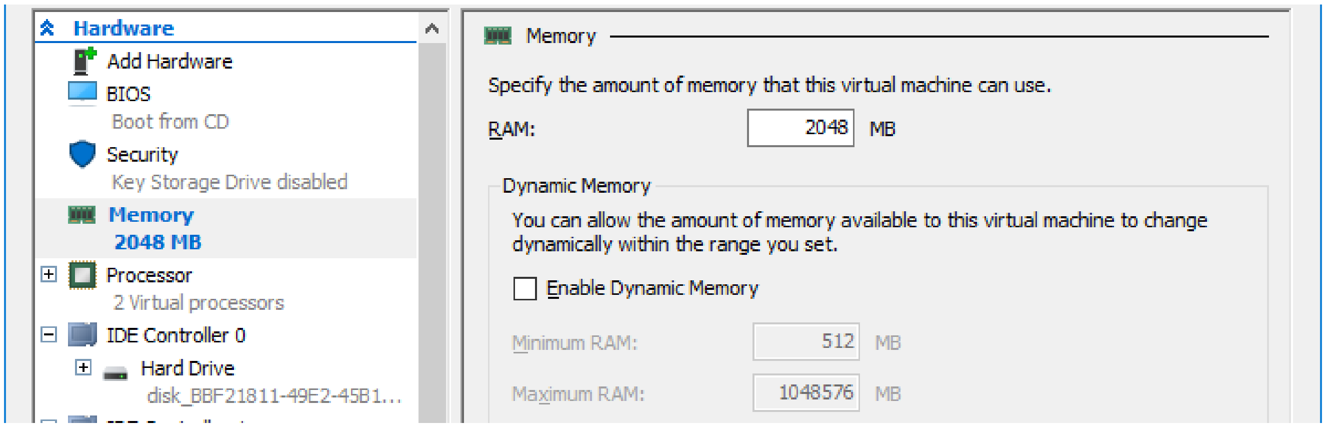
1. In the `boxstarter shell` run the command `minikube stop`



```
PS C:\Users\robert_m> minikube stop
Stopping local Kubernetes cluster...

 Shutting Down
    0%
    [                                                                          ]
```

~ If this takes more then a minute, go into Hyper-V manager, connect to the VM, login as root, and shutdown the minikube VM ~

2. Open the `Hyper-V Manager` (Windows+R) and right-click on the stopped `minikube` VM.
3. Select **Settings** option and navigate to **Memory** settings
4. **Disable** the *Dynamic Memory* setting (uncheck checkbox) and accept the changes.



Specify the amount of memory that this virtual machine can use.

RAM: 2048 MB

Dynamic Memory

You can allow the amount of memory available to this virtual machine to change dynamically within the range you set.

☐ Enable Dynamic Memory

Minimum RAM: 512 MB

Maximum RAM: 1048576 MB

Conclusion

As of now your Windows 10 native environment is ready for usage in the next labs.